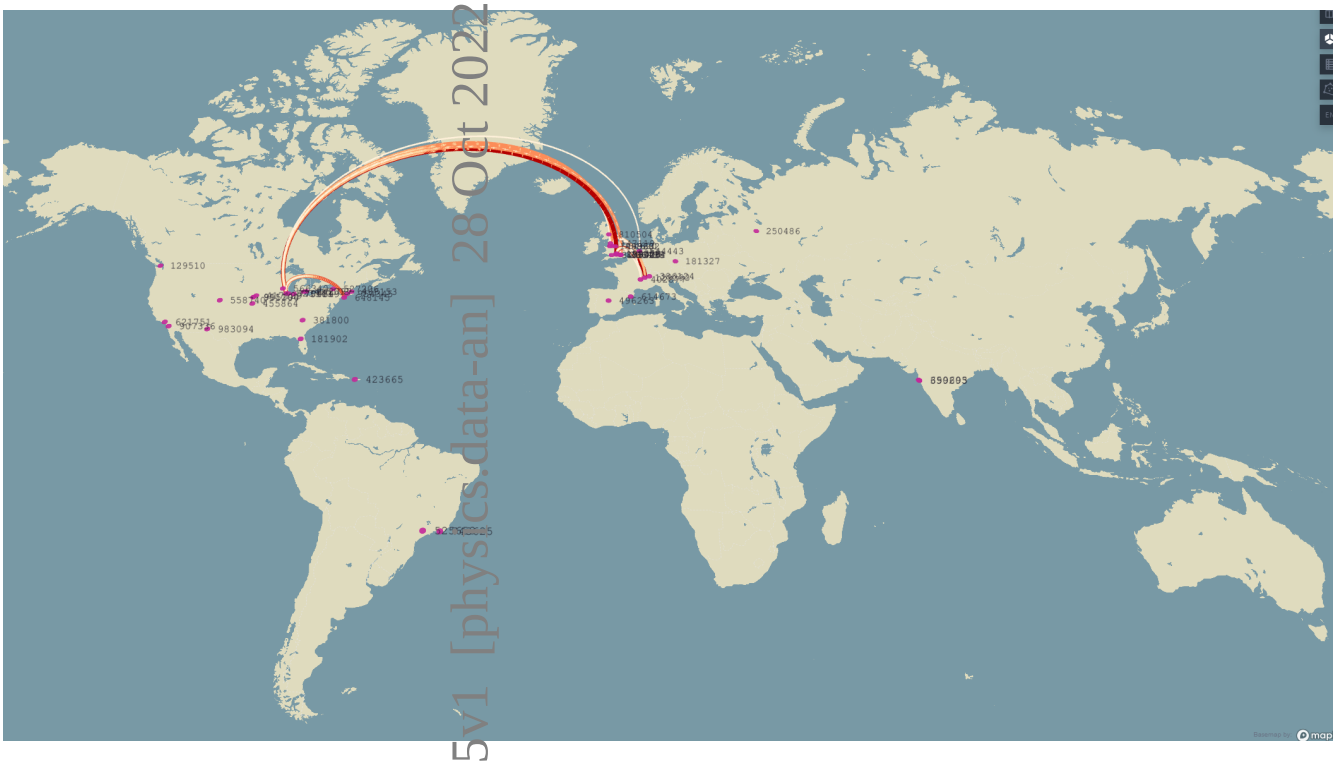
DUNE Offline Computing

Conceptual Design Report



October 31, 2022

The DUNE Collaboration

DEEP UNDERGROUND NEUTRINO EXPERIMENT

FERMILAB-DESIGN-22-01

This document was prepared by the DUNE collaboration using the resources of the Fermi National Accelerator Laboratory (Fermilab), a U.S. Department of Energy, Office of Science, HEP User Facility. Fermilab is managed by Fermi Research Alliance, LLC (FRA), acting under Contract No. DE-AC02-07CH11359. This work was supported by CNPq, FAPERJ, FAPEG and FAPESP, Brazil; CFI, IPP and NSERC, Canada; CERN; MŠMT, Czech Republic; ERDF, H2020-EU and MSCA, European Union; CNRS/IN2P3 and CEA, France; INFN, Italy; FCT, Portugal; NRF, South Korea; CAM, Fundación "La Caixa", Junta de Andalucía-FEDER, MICINN, and Xunta de Galicia, Spain; SERI and SNSF, Switzerland; TÜBİTAK, Turkey; The Royal Society and UKRI/STFC, United Kingdom; DOE and NSF, United States of America.

The ProtoDUNE-SP and ProtoDUNE-DP detectors were constructed and operated on the CERN Neutrino Platform. We gratefully acknowledge the support of the CERN management, and the CERN EP, BE, TE, EN and IT Departments for NP04/ProtoDUNE-SP.

This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231.

Authors

A. Abed Abud,[35] B. Abi,[160] R. Acciarri,[67] M. A. Acero,[11] M. R. Adames,[197] G. Adamov,[73] M. Adamowski,[67] D. Adams,[20] M. Adinolfi,[19] C. Adriano,[30] A. Aduszkiewicz,[82] J. Aguilar,[129] Z. Ahmad,[208] J. Ahmed,[211] B. Aimard,[53] F. Akbar,[178] K. Allison,[43] S. Alonso Monsalve,[35] M. Alrashed,[121] C. Alt,[60] A. Alton,[12] R. Alvarez,[39] P. Amedo,[87, 86] J. Anderson,[7] D. A. Andrade,[88] C. Andreopoulos,[181, 131] M. Andreotti,[95, 68] M. P. Andrews,[67] F. Andrianala,[5] S. Andringa,[130] N. Anfimov,[119] W. L. Anicézio Campanelli,[63] A. Ankowski,[187] M. Antoniassi,[197] M. Antonova,[86] A. Antoshkin,[119] S. Antusch,[13] A. Aranda-Fernandez,[42] L. Arellano,[137] L. O. Arnold,[45] M. A. Arroyave,[59] J. Asaadi,[200] L. Asquith,[195] A. Aurisano,[40] V. Aushev,[127] D. Autiero,[111] M. Ayala-Torres,[41] F. Azfar,[160] A. Back,[92] H. Back,[161] J. J. Back,[211] I. Bagaturia,[73] L. Bagby,[67] N. Balashov,[119] S. Balasubramanian,[67] P. Baldi,[24] B. Baller,[67] B. Bambah,[83] F. Barao,[130, 113] G. Barenboim,[86] G. J. Barker,[211] W. Barkhouse,[152] C. Barnes,[141] G. Barr,[160] J. Barranco Monarca,[78] A. Barros,[197] N. Barros,[130, 62] J. L. Barrow,[138] A. Basharina-Freshville,[206] A. Bashyal,[7] V. Basque,[67] C. Batchelor,[58] J.B.R. Battat,[212] F. Battisti,[160] F. Bay,[4] M. C. Q. Bazetto,[30] J. L. L. Bazo Alba,[173] J. F. Beacom,[158] E. Bechetoille,[111] B. Behera,[44] E. Belchior,[30] L. Bellantoni,[67] G. Bellettini,[103, 171] V. Bellini,[94, 31] O. Beltramello,[35] N. Benekos,[35] C. Benitez Montiel,[9] D. Benjamin,[20] F. Bento Neves,[130] J. Berger,[44] S. Berkman,[67] P. Bernardini,[97, 182] R. M. Berner,[14] A. Bersani,[96] S. Bertolucci,[93, 17] M. Betancourt,[67] A. Betancur Rodríguez,[59] A. Bevan,[176] Y. Bezawada,[23] A. T. Bezerra,[63] T. J. Bezerra,[195] J. Bhambure,[192] A. Bhardwaj,[133] V. Bhatnagar,[163] M. Bhattacharjee,[90] D. Bhattarai,[147] S. Bhuller,[19] B. Bhuyan,[90] S. Biagi,[105] J. Bian,[24] M. Biassoni,[98] K. Biery,[67] B. Bilki,[15, 109] M. Bishai,[20] V. Bisignani,[100] A. Bitadze,[137] A. Blake,[128] F. D. Blaszczyk,[67] G. C. Blazey,[153] D. Blend,[109] E. Blucher,[37] J. Boissevain,[132] A. Bobyshev,[67] S. Bolognesi,[34] T. Bolton,[121] L. Bomben,[98, 108] M. Bonesini,[98, 143] C. Bonilla-Diaz,[32] F. Bonini,[20] A. Booth,[176] F. Boran,[15] S. Bordoni,[35] A. Borkum,[195] N. Bostan,[109] P. Bour,[50] D. Boyden,[153] J. Bracinik,[16] D. Braga,[67] D. Brailsford,[128] A. Branca,[98] A. Brandt,[200] J. Bremer,[35] C. Brew,[181] S. J. Brice,[67] C. Brizzolari,[98, 143] C. Bromberg,[142] J. Brooke,[19] A. Bross,[67] G. Brunetti,[98, 143] M. Brunetti,[211] L. Brynmoor,[159] N. Buchanan,[44] H. Budd,[178] J. Buergi,[14] G. Caceres V.,[23] I. Cagnoli,[93, 17] T. Cai,[218] D. Caiulo,[111] R. Calabrese,[95, 68] P. Calafiura,[129] J. Calcutt,[159] M. Calin,[21] L. Calivers,[14] S. Calvez,[44] E. Calvo,[39] A. Caminata,[96] D. Caratelli,[27] D. Carber,[44] J. C. Carceller,[206] G. Carini,[20] B. Carlus,[111] M. F. Carneiro,[20] P. Carniti,[98] I. Caro Terrazas,[44] H. Carranza,[200] N. Carrara,[23] L. Carroll,[121] T. Carroll,[215] J. F. Castaño Forero,[6] A. Castillo,[185] E. Catano-Mur,[214] C. Cattadori,[98] F. Cavalier,[164] G. Cavallaro,[98] F. Cavanna,[67] S. Centro,[162] G. Cerati,[67] A. Cervelli,[93] A. Cervera Villanueva,[86] K. Chakraborty,[170] M. Chalifour,[35] A. Chappell,[211] E. Chardonnet,[165] N. Charitonidis,[35] A. Chatterjee,[172] S. Chattopadhyay,[208] H. Chen,[20] M. Chen,[24] Y. Chen,[14, 187] Z. Chen,[192] Z. Chen-Wishart,[179] Y. Cheon,[205] D. Cherdack,[82] C. Chi,[45] S. Childress,[67] R. Chirco,[88] A. Chiriacescu,[21] N. Chitirasreemadam,[103, 171] K. Cho,[124] S. Choate,[153] D. Chokheli,[73] P. S. Chong,[168] B. Chowdhury,[7] A. Christensen,[44] D. Christian,[67] G. Christodoulou,[35] A. Chukanov,[119] M. Chung,[205] E. Church,[161] V. Cicero,[93, 17] P. Clarke,[58] G. Cline,[129] T. E. Coan,[191] A. G. Cocco,[100] J. A. B. Coelho,[165] J. Collot,[77] E. Conley,[56] J. M. Conrad,[138] M. Convery,[187] S. Copello,[96] P. Cova,[99, 166] L. Cremaldi,[147] L. Cremonesi,[176] J. I. Crespo-Anadón,[39] M. Crisler,[67] E. Cristaldo,[99, 9] J. Crnkovic,[67] R. Cross,[128] A. Cudd,[43] C. Cuesta,[39] Y. Cui,[26] D. Cussans,[19] O. Dalager,[24] R. Dallavalle,[165] H. da Motta,[33] Z. A. Dar,[214] L. Da Silva Peres,[66] C. David,[218, 67] Q. David,[111] G. S. Davies,[147] S. Davini,[96] J. Dawson,[165] K. De,[200] S. De,[2] P. De Almeida,[30] P. Debbins,[109] I. De Bonis,[53] M. P. Decowski,[151, 3] A. de Gouvêa,[154] P. C. De Holanda,[30] I. L. De Icaza Astiz,[195] A. Deisting,[136] P. De Jong,[151, 3] A. De la Torre,[39] A. Delbart,[34] V. De Leo,[184, 104] D. Delepine,[78] M. Delgado,[98, 143] A. Dell'Acqua,[35] N. Delmonte,[99, 166] P. De Lurgio,[7] P. J. DeMar,[67] J. R. T. de Mello Neto,[66] D. M. DeMuth,[207] S. Dennis,[29] C. Densham,[181] G. W. Deptuch,[20] A. De Roeck,[35] V. De Romeri,[86] G. De Souza,[30] J. P. Detje,[29] R. Devi,[116] R. Dharmapalan,[81] M. Dias,[204] J. S. Díaz,[92] F. Díaz,[173] F. Di Capua,[100, 148] A. Di Domenico,[184, 104] S. Di Domizio,[96, 72] L. Di Giulio,[35] P. Ding,[67] L. Di Noto,[96, 72] C. Distefano,[105] R. Diurba,[14] M. Diwan,[20] Z. Djurcic,[7] D. Doering,[187] S. Dolan,[35] F. Dolek,[15] M. J. Dolinski,[55] L. Domine,[187] S. Donati,[103, 171] Y. Donon,[35] S. Doran,[110] D. Douglas,[142] A. Dragone,[187] F. Drielsma,[187] L. Duarte,[204] D. Duchesneau,[53] K. Duffy,[160, 67] K. Dugas,[24] P. Dunne,[89] B. Dutta,[198] H. Duyang,[188] O. Dvornikov,[81] D. A. Dwyer,[129] A. S. Dyshkant,[153] M. Eads,[153] A. Earle,[195] D. Edmunds,[142] J. Eisch,[67] L. Emberger,[137, 139] P. Englezos,[180] A. Ereditato,[216] T. Erjavec,[23] C. O. Escobar,[67] J. J. Evans,[137] E. Ewart,[92] A. C. Ezeribe,[186] K. Fahey,[67] L. Fajt,[35] A. Falcone,[98, 143] M. Fani',[132] C. Farnese,[101] Y. Farzan,[112] D. Fedoseev,[119] J. Felix,[78] Y. Feng,[110] E. Fernandez-Martinez,[135] F. Ferraro,[96, 72] L. Fields,[155] P. Filip,[49] A. Filkins,[196] F. Filthaut,[151, 177] R. Fine,[132] G. Fiorillo,[100, 148] M. Fiorini,[95, 68] V. Fischer,[110] R. S. Fitzpatrick,[141] W. Flanagan,[52] B. Fleming,[37, 216] R. Flight,[178] S. Fogarty,[44] W. Foreman,[88] J. Fowler,[56] J. Franc,[50] D. Franco,[216] J. Freeman,[67] J. Freestone,[137] J. Fried,[20] A. Friedland,[187] S. Fuess,[67] I. K. Furic,[69] K. Furman,[176] A. P. Furmanski,[146] A. Gabrielli,[93, 17] A. Gago,[173] H. Gallagher,[203]

A. Gallas,[164] A. Gallego-Ros,[39] N. Gallice,[99,144] V. Galymov,[111] E. Gamberini,[35] T. Gamble,[186] F. Ganacim,[197] R. Gandhi,[79] S. Ganguly,[67] F. Gao,[172] S. Gao,[20] D. Garcia-Gamez,[74] M. Á. García-Peris,[86] S. Gardiner,[67] D. Gastler,[18] A. Gauch,[14] J. Gauvreau,[157] P. Gauzzi,[184,104] G. Ge,[45] N. Geffroy,[53] B. Gelli,[30] A. Gendotti,[60] S. Gent,[190] Z. Ghorbani-Moghaddam,[96] P. Giammaria,[30] T. Giammaria,[95,68] N. Giangiacomi,[202] D. Gibin,[162,101] I. Gil-Botella,[39] S. Gilligan,[159] A. Gioiosa,[103] C. Girerd,[111] A. K. Giri,[91] D. Gnani,[129] O. Gogota,[127] M. Gold,[149] S. Gollapinni,[132] K. Gollwitzer,[67] R. A. Gomes,[64] L. V. Gomez Bermeo,[185] L. S. Gomez Fajardo,[185] F. Gonnella,[16] D. Gonzalez-Diaz,[87] M. Gonzalez-Lopez,[135] M. C. Goodman,[7] O. Goodwin,[137] S. Goswami,[170] C. Gotti,[98] E. Goudzovski,[16] C. Grace,[129] R. Gran,[145] E. Granados,[78] P. Granger,[34] C. Grant,[18] D. Gratieri,[71] P. Green,[137] S. Greenberg,[22,129] L. Greenler,[215] J. Greer,[19] J. Grenard,[35] W. C. Griffith,[195] F. T. Groetschla,[35] M. Groh,[44] K. Grzelak,[210] W. Gu,[20] E. Guardincerri,[132] V. Guarino,[7] M. Guarise,[95,68] R. Guenette,[137] E. Guerard,[164] M. Guerzoni,[93] D. Guffanti,[98] A. Guglielmi,[101] B. Guo,[188] A. Gupta,[187] V. Gupta,[151,3] K. K. Guthikonda,[125] P. Guzowski,[137] M. M. Guzzo,[30] S. Gwon,[38] C. Ha,[38] K. Haaf,[67] A. Habig,[145] H. Hadavand,[200] R. Haenni,[14] L. Hagaman,[216] A. Hahn,[67] J. Haiston,[189] P. Hamacher-Baumann,[160] T. Hamernik,[67] P. Hamilton,[89] J. Han,[172] D. A. Harris,[218,67] J. Hartnell,[195] T. Hartnett,[181] J. Harton,[44] T. Hasegawa,[123] C. Hasnip,[160] R. Hatcher,[67] K. W. Hatfield,[24] A. Hatzikoutelis,[183] C. Hayes,[92] K. Hayrapetyan,[176] J. Hays,[176] E. Hazen,[18] M. He,[82] A. Heavey,[67] K. M. Heeger,[216] J. Heise,[194] S. Henry,[178] M. A. Hernandez Morquecho,[88] K. Herner,[67] V. Hewes,[40] C. Hilgenberg,[146] T. Hill,[84] S. J. Hillier,[16] A. Himmel,[67] E. Hinkle,[37] L.R. Hirsch,[197] J. Hoff,[67] A. Holin,[181] E. Hoppe,[161] G. A. Horton-Smith,[121] M. Hostert,[146] A. Hourlier,[138] B. Howard,[67] R. Howell,[178] J. Hoyos Barrios,[140] I. Hristova,[181] M. S. Hronek,[67] J. Huang,[23] R. Huang,[129] Z. Hulcher,[187] G. Iles,[89] N. Ilic,[202] A. M. Iliescu,[93] R. Illingworth,[67] G. Ingratta,[93,17] A. Ioannisian,[217] B. Irwin,[146] L. Isenhower,[1] M. Ismerio Oliveira,[66] R. Itay,[187] C.M. Jackson,[161] V. Jain,[2] E. James,[67] W. Jang,[200] B. Jargowsky,[24] F. Jediny,[50] D. Jena,[67] Y. S. Jeong,[38] C. Jesús-Valls,[85] X. Ji,[20] J. Jiang,[192] L. Jiang,[209] A. Jipa,[21] J. H. Jo,[216] F. R. Joaquim,[130,113] W. Johnson,[189] B. Jones,[200] R. Jones,[186] N. Jovancevic,[156] M. Judah,[172] C. K. Jung,[192] T. Junk,[67] Y. Jwa,[45] M. Kabirnezhad,[89] A. Kaboth,[179,181] I. Kadenko,[127] I. Kakorin,[119] A. Kalitkina,[119] D. Kalra,[45] O. Kamer Koseyan,[109] F. Kamiya,[65] D. M. Kaplan,[88] G. Karagiorgi,[45] G. Karaman,[109] A. Karcher,[129] Y. Karyotakis,[53] S. Kasai,[126] S. P. Kasetti,[133] L. Kashur,[44] I. Katsioulas,[16] N. Kazaryan,[217] E. Kearns,[18] P. Keener,[168] K.J. Kelly,[35] E. Kemp,[30] O. Kemularia,[73] W. Ketchum,[67] S. H. Kettell,[20] M. Khabibullin,[107] A. Khotjantsev,[107] A. Khvedelidze,[73] D. Kim,[198] B. King,[67] B. Kirby,[45] M. Kirby,[67] J. Klein,[168] J. Kleykamp,[147] A. Klustova,[89] T. Kobilarcik,[67] K. Koehler,[215] L. W. Koerner,[82] D. H. Koh,[187] S. Kohn,[22,129] P. P. Koller,[14] L. Kolupaeva,[119] D. Korablev,[119] M. Kordosky,[214] T. Kosc,[77] U. Kose,[35] V. A. Kostelecký,[92] K. Kothekar,[19] I. Kotler,[55] V. Kozhukalov,[119] R. Kralik,[195] L. Kreczko,[19] F. Krennrich,[110] I. Kreslo,[14] W. Kropp,[24] T. Kroupova,[168] Y. Kudenko,[107] V. A. Kudryavtsev,[186] S. Kuhlmann,[7] S. Kulagin,[107] J. Kumar,[81] P. Kumar,[186] P. Kunze,[53] R. Kuravi,[129] N. Kurita,[187] C. Kuruppu,[188] V. Kus,[50] T. Kutter,[133] J. Kvasnicka,[49] D. Kwak,[205] A. Lambert,[129] B. J. Land,[168] C. E. Lane,[55] K. Lang,[201] T. Langford,[216] M. Langstaff,[137] F. Lanni,[35] O. Lantwin,[53] J. Larkin,[20] P. Lasorak,[89] D. Last,[168] A. Laundrie,[215] G. Laurenti,[93] A. Lawrence,[129] P. Laycock,[20] I. Lazanu,[21] M. Lazzaroni,[99,144] T. Le,[203] S. Leardini,[87] J. Learned,[81] P. LeBrun,[111] T. LeCompte,[187] C. Lee,[67] Z. I. Lee,[159] V. Legin,[127] G. Lehmann Miotto,[35] R. Lehnert,[92] M. A. Leigui de Oliveira,[65] M. Leitner,[129] L. M. Lepin,[137] S. W. Li,[187] Y. Li,[20] H. Liao,[121] C. S. Lin,[129] S. Lin,[133] R. A. Lineros,[32] J. Ling,[193] A. Lister,[215] B. R. Littlejohn,[88] J. Liu,[24] Y. Liu,[37] S. Lockwitz,[67] T. Loew,[129] M. Lokajicek,[49] I. Lomidze,[73] K. Long,[89] T. Lord,[211] J. M. LoSecco,[155] W. C. Louis,[132] X.-G. Lu,[211] K.B. Luk,[22,129] B. Lunday,[168] X. Luo,[27] E. Luppi,[95,68] T. Lux,[85] V. P. Luzio,[65] J. Maalmi,[164] D. MacFarlane,[187] A. A. Machado,[30] P. Machado,[67] C. T. Macias,[92] J. R. Macier,[67] A. Maddalena,[76] A. Madera,[35] P. Madigan,[22,129] S. Magill,[7] K. Mahn,[142] A. Maio,[130,62] A. Major,[56] K. Majumdar,[131] J. A. Maloney,[51] I. V. Mandrichenko,[67] G. Mandrioli,[93] R. C. Mandujano,[24] J. Maneira,[130,62] L. Manenti,[206] S. Manly,[178] A. Mann,[203] K. Manolopoulos,[181] M. Manrique Plata,[92] V. N. Manyam,[20] M. Marchan,[67] A. Marchionni,[67] W. Marciano,[20] D. Marfatia,[81] C. Mariani,[209] J. Maricic,[81] F. Marinho,[114] A. D. Marino,[43] T. Markiewicz,[187] D. Marsden,[137] M. Marshak,[146] C. M. Marshall,[178] J. Marshall,[211] J. Marteau,[111] J. Martín-Albo,[86] N. Martinez,[121] D.A. Martinez Caicedo,[189] F. Martínez López,[176] P. Martínez Miravé,[86] S. Martynenko,[20] V. Mascagna,[98,108] K. Mason,[203] A. Mastbaum,[180] F. Matichard,[129] S. Matsuno,[81] J. Matthews,[133] C. Mauger,[168] N. Mauri,[93,17] K. Mavrokoridis,[131] I. Mawby,[211] R. Mazza,[98] A. Mazzacane,[67] T. McAskill,[212] E. McCluskey,[67] N. McConkey,[137] K. S. McFarland,[178] C. McGrew,[192] A. McNab,[137] A. Mefodiev,[107] P. Mehta,[117] P. Melas,[10] O. Mena,[86] H. Mendez,[174] P. Mendez,[35] D. P. Méndez,[20] A. Menegolli,[102,167] G. Meng,[101] M. D. Messier,[92] W. Metcalf,[133] M. Mewes,[92] H. Meyer,[213] T. Miao,[67] G. Michna,[190] V. Mikola,[206] R. Milincic,[81] G. Miller,[137] W. Miller,[146] J. Mills,[203] O. Mineev,[107] A. Minotti,[98,143] O. G. Miranda,[41] S. Miryala,[20] C. S. Mishra,[67] S. R. Mishra,[188] A. Mislivec,[146] M. Mitchell,[133] D. Mladenov,[35]

I. Mocioiu,[169] K. Moffat,[57] A. Mogan,[44] N. Moggi,[93,17] R. Mohanta,[83] T. A. Mohayai,[67] N. Mokhov,[67] J. Molina,[9] L. Molina Bueno,[86] E. Montagna,[93,17] A. Montanari,[93] C. Montanari,[102,67,167] D. Montanari,[67] D. Montanino,[97,182] L. M. Montaño Zetina,[41] S. H. Moon,[205] M. Mooney,[44] A. F. Moor,[29] D. Moreno,[6] D. Moretti,[98] C. Morris,[82] C. Mossey,[67] M. Mote,[133] E. Motuk,[206] C. A. Moura,[65] J. Mousseau,[141] G. Mouster,[128] W. Mu,[67] L. Mualem,[28] J. Mueller,[44] M. Muether,[213] F. Muheim,[58] A. Muir,[54] M. Mulhearn,[23] D. Munford,[82] H. Muramatsu,[146] M. Murphy,[209] S. Murphy,[60] J. Musser,[92] J. Nachtman,[109] Y. Nagai,[61] S. Nagu,[134] M. Nalbandyan,[217] R. Nandakumar,[181] D. Naples,[172] S. Narita,[115] A. Nath,[90] A. Navrer-Agasson,[137] N. Nayak,[20] M. Nebot-Guinot,[58] K. Negishi,[115] J. K. Nelson,[214] M. Nelson,[109] J. Nesbit,[215] M. Nessi,[67,35] D. Newbold,[181] M. Newcomer,[168] H. Newton,[54] R. Nichol,[206] F. Nicolas-Arnaldos,[74] A. Nikolica,[168] J. Nikolov,[156] E. Niner,[67] K. Nishimura,[81] A. Norman,[67] A. Norrick,[67] P. Novella,[86] J. A. Nowak,[128] M. Oberling,[7] J. P. Ochoa-Ricoux,[24] A. Olivier,[178] A. Olshevskiy,[119] Y. Onel,[109] Y. Onishchuk,[127] L. Otiniano Ormachea,[46,106] J. Ott,[24] L. Pagani,[23] G. Palacio,[59] O. Palamara,[67] S. Palestini,[35] J. M. Paley,[67] M. Pallavicini,[96,72] C. Palomares,[39] S. Pan,[170] W. Panduro Vazquez,[179] E. Pantic,[23] V. Paolone,[172] V. Papadimitriou,[67] R. Papaleo,[105] A. Papanestis,[181] S. Paramesvaran,[19] S. Parke,[67] E. Parozzi,[98,143] S. Parsa,[14] Z. Parsa,[20] S. Parveen,[117] M. Parvu,[21] D. Pasciuto,[103] S. Pascoli,[57,17] R. Pasetes,[67] L. Pasqualini,[93,17] J. Pasternak,[89] J. Pater,[137] C. Patrick,[58,206] L. Patrizii,[93] R. B. Patterson,[28] S. J. Patton,[129] T. Patzak,[165] A. Paudel,[67] L. Paulucci,[65] Z. Pavlovic,[67] G. Pawloski,[146] D. Payne,[131] V. Pec,[49] S. J. M. Peeters,[195] A. Pena Perez,[187] E. Pennacchio,[111] A. Penzo,[109] O. L. G. Peres,[30] C. Pernas,[214] J. Perry,[58] D. Pershey,[56] G. Pessina,[98] G. Petrillo,[187] C. Petta,[94,31] R. Petti,[188] V. Pia,[93,17] L. Pickering,[179] F. Pietropaolo,[35,101] V. L. Pimentel,[47,30] G. Pinaroli,[20] K. Plows,[160] R. Plunkett,[67] F. Pompa,[86] X. Pons,[35] N. Poonthottathil,[110] F. Poppi,[93,17] S. Pordes,[67] J. Porter,[195] S. D. Porzio,[14] M. Potekhin,[20] R. Potenza,[94,31] B. V. K. S. Potukuchi,[116] J. Pozimski,[89] M. Pozzato,[93,17] S. Prakash,[30] T. Prakash,[129] C. Pratt,[23] M. Prest,[98] F. Psihas,[67] D. Pugnere,[111] X. Qian,[20] J. L. Raaf,[67] V. Radeka,[20] J. Rademacker,[19] R. Radev,[35] B. Radics,[218] A. Rafique,[7] E. Raguzin,[20] M. Rai,[211] M. Rajaoalisoa,[40] I. Rakhno,[67] A. Rakotonandrasana,[5] L. Rakotondravohitra,[5] R. Rameika,[67] M. A. Ramirez Delgado,[168] B. Ramson,[67] A. Rappoldi,[102,167] G. Raselli,[102,167] P. Ratoff,[128] S. Raut,[192] H. Razafinime,[40] R. F. Razakamiandra,[5] E. M. Rea,[146] J. S. Real,[77] B. Rebel,[215,67] R. Rechenmacher,[67] M. Reggiani-Guzzo,[137] J. Reichenbacher,[189] S. D. Reitzner,[67] H. Rejeb Sfar,[35] E. Renner,[132] A. Renshaw,[82] S. Rescia,[20] F. Resnati,[35] M. Ribas,[197] S. Riboldi,[99] C. Riccio,[192] G. Riccobene,[105] L. C. J. Rice,[172] J. S. Ricol,[77] A. Rigamonti,[35] Y. Rigaut,[60] E. V. Rincón,[59] A. Ritchie-Yates,[179] D. Rivera,[132] R. Rivera,[67] A. Robert,[77] J. L. Rocabado Rocha,[86] L. Rochester,[187] M. Roda,[131] P. Rodrigues,[160] M. J. Rodriguez Alonso,[35] J. Rodriguez Rondon,[189] E. Romeo,[100] S. Rosauro-Alcaraz,[164] P. Rosier,[164] M. Rossella,[102,167] M. Rossi,[35] M. Ross-Lonergan,[132] J. Rout,[117] P. Roy,[213] A. Rubbia,[60] C. Rubbia,[75] B. Russell,[129] D. Ruterbories,[178] A. Rybnikov,[119] A. Saa-Hernandez,[87] R. Saakyan,[206] S. Sacerdoti,[165] N. Sahu,[91] P. Sala,[99,35] N. Samios,[20] O. Samoylov,[119] M. C. Sanchez,[70] V. Sandberg,[132] D. A. Sanders,[147] D. Sankey,[181] D. Santoro,[99] N. Saoulidou,[10] P. Sapienza,[105] C. Sarasty,[40] I. Sarcevic,[8] G. Savage,[67] V. Savinov,[172] G. Scanavini,[216] A. Scaramelli,[102] A. Scarff,[186] A. Scarpelli,[20] T. Schefke,[133] H. Schellman,[159,67] S. Schifano,[95,68] P. Schlabach,[67] D. Schmitz,[37] A. W. Schneider,[138] K. Scholberg,[56] A. Schukraft,[67] E. Segreto,[30] A. Selyunin,[119] C. R. Senise,[204] J. Sensenig,[168] D. Sgalaberna,[60] M. H. Shaevitz,[45] S. Shafaq,[117] F. Shaker,[218] M. Shamma,[26] P. Shanahan,[67] R. Sharankova,[203] H. R. Sharma,[116] R. Sharma,[20] R. Kumar,[175] K. Shaw,[195] T. Shaw,[67] K. Shchablo,[111] C. Shepherd-Themistocleous,[181] A. Sheshukov,[119] W. Shi,[192] S. Shin,[118] I. Shoemaker,[209] D. Shooltz,[142] R. Shrock,[192] J. Silber,[129] L. Simard,[164] F. Simon,[67,139] J. Sinclair,[187] G. Sinev,[189] Jaydip Singh,[134] J. Singh,[134] L. Singh,[48] P. Singh,[176] V. Singh,[48] S. Singh Chauhan,[163] R. Sipos,[35] G. Sirri,[93] A. Sitraka,[189] K. Siyeon,[38] K. Skarpaas,[187] E. Smith,[92] P. Smith,[92] J. Smolik,[50] M. Smy,[24] E.L. Snider,[67] P. Snopok,[88] D. Snowden-Ifft,[157] M. Soares Nunes,[196] H. Sobel,[24] M. Soderberg,[196] S. Sokolov,[119] C. J. Solano Salinas,[106] S. Söldner-Rembold,[137] S.R. Soleti,[129] N. Solomey,[213] V. Solovov,[130] W. E. Sondheim,[132] M. Sorel,[86] A. Sotnikov,[119] J. Soto-Oton,[39] A. Sousa,[40] K. Soustruznik,[36] F. Spagliardi,[160] M. Spanu,[98,143] J. Spitz,[141] N. J. C. Spooner,[186] K. Spurgeon,[196] D. Stalder,[9] M. Stancari,[67] L. Stanco,[101,162] J. Steenis,[23] R. Stein,[19] H. M. Steiner,[129] A. F. Steklain Lisbôa,[197] A. Stepanova,[119] J. Stewart,[20] B. Stillwell,[37] J. Stock,[189] F. Stocker,[35] T. Stokes,[133] M. Strait,[146] T. Strauss,[67] L. Strigari,[198] A. Stuart,[42] J. G. Suarez,[59] J. Subash,[16] P. Sundararajan,[24] A. Surdo,[97] V. Susic,[13] L. Suter,[67] C. M. Sutera,[94,31] Y. Suvorov,[100,148] R. Svoboda,[23] B. Szczerbinska,[199] A. M. Szelc,[58] N. Talukdar,[188] J. Tamara,[6] H. A. Tanaka,[187] S. Tang,[20] B. Tapia Oregui,[201] A. Tapper,[89] S. Tariq,[67] E. Tarpara,[20] N. Tata,[80] E. Tatar,[84] R. Tayloe,[92] A. M. Teklu,[192] P. Tennessen,[129,4] M. Tenti,[93] K. Terao,[187] F. Terranova,[98,143] G. Testera,[96] T. Thakore,[40] A. Thea,[181] A. Thompson,[198] C. Thorn,[20] S. C. Timm,[67] V. Tishchenko,[20] N. Todorović,[156] L. Tomassetti,[95,68] A. Tonazzo,[165] D. Torbunov,[20] M. Torti,[98,143] M. Tortola,[86] F. Tortorici,[94,31] N. Tosi,[93]

D. Totani,[27] M. Toups,[67] C. Touramanis,[131] R. Travaglini,[93] J. Trevor,[28] S. Trilov,[19] W. H. Trzaska,[120] Y. Tsai,[24] Y.-T. Tsai,[187] Z. Tsamalaidze,[73] K. V. Tsang,[187] N. Tsverava,[73] S. Tufanli,[35] C. Tull,[129] J. Turner,[57] J. Tyler,[121] E. Tyley,[186] M. Tzanov,[133] L. Uboldi,[35] M. A. Uchida,[29] J. Urheim,[92] T. Usher,[187] H. Utaegbulam,[196] S. Uzunyan,[153] M. R. Vagins,[122,24] P. Vahle,[214] S. Valder,[195] G. D. A. Valdiviesso,[63] E. Valencia,[78] R. Valentim,[204] Z. Vallari,[28] E. Vallazza,[98] J. W. F. Valle,[86] S. Vallecorsa,[35] R. Van Berg,[168] R. G. Van de Water,[132] D. Vanegas Forero,[140] D. Vannerom,[138] F. Varanini,[101] D. Vargas Oliva,[202] G. Varner,[81] S. Vasina,[119] N. Vaughan,[159] K. Vaziri,[67] J. Vega,[46] S. Ventura,[101] A. Verdugo,[39] S. Vergani,[29] M. A. Vermeulen,[151] M. Verzocchi,[67] M. Vicenzi,[96,72] H. Vieira de Souza,[165] C. Vignoli,[76] C. Vilela,[35] B. Viren,[20] T. Vrba,[50] Q. Vuong,[178] T. Wachala,[150] A. Wagner,[146] A. V. Waldron,[176] M. Wallbank,[40] T. Walton,[67] H. Wang,[25] J. Wang,[189] L. Wang,[129] M.H.L.S. Wang,[67] X. Wang,[67] Y. Wang,[25] Y. Wang,[192] K. Warburton,[110] D. Warner,[44] M.O. Wascko,[89] D. Waters,[206] A. Watson,[16] K. Wawrowska,[181,195] P. Weatherly,[55] A. Weber,[136,67] M. Weber,[14] H. Wei,[133] A. Weinstein,[110] D. Wenman,[215] M. Wetstein,[110] J. Whilhelmi,[216] A. White,[200] A. White,[216] B. White,[67] S. White,[67] L. H. Whitehead,[29] D. Whittington,[196] M. J. Wilking,[192] A. Wilkinson,[206] C. Wilkinson,[129] Z. Williams,[200] F. Wilson,[181] R. J. Wilson,[44] W. Wisniewski,[187] J. Wolcott,[203] J. Wolfs,[178] T. Wongjirad,[203] A. Wood,[82] K. Wood,[129] E. Worcester,[20] M. Worcester,[20] M. Wospakrik,[67] K. Wresilo,[29] C. Wret,[178] S. Wu,[146] W. Wu,[67] W. Wu,[24] Y. Xiao,[24] I. Xiotidis,[89] B. Yaeggy,[40] E. Yandel,[27] G. Yang,[192] K. Yang,[160] T. Yang,[67] A. Yankelevich,[24] N. Yershov,[107] K. Yonehara,[67] Y. S. Yoon,[38] T. Young,[152] B. Yu,[20] H. Yu,[20] H. Yu,[193] J. Yu,[200] Y. Yu,[88] W. Yuan,[58] R. Zaki,[218] J. Zalesak,[49] L. Zambelli,[53] B. Zamorano,[74] A. Zani,[99] L. Zazueta,[214] G. P. Zeller,[67] J. Zennamo,[67] K. Zeug,[215] C. Zhang,[20] S. Zhang,[92] Y. Zhang,[172] M. Zhao,[20] E. Zhivun,[20] E. D. Zimmerman,[43] S. Zucchelli,[93,17] J. Zuklin,[49] V. Zutshi,[153] and R. Zwaska[67]

(The DUNE Collaboration)

[1] *Abilene Christian University, Abilene, TX 79601, USA*
[2] *University of Albany, SUNY, Albany, NY 12222, USA*
[3] *University of Amsterdam, NL-1098 XG Amsterdam, The Netherlands*
[4] *Antalya Bilim University, 07190 Döşemealtı/Antalya, Turkey*
[5] *University of Antananarivo, Antananarivo 101, Madagascar*
[6] *Universidad Antonio Nariño, Bogotá, Colombia*
[7] *Argonne National Laboratory, Argonne, IL 60439, USA*
[8] *University of Arizona, Tucson, AZ 85721, USA*
[9] *Universidad Nacional de Asunción, San Lorenzo, Paraguay*
[10] *University of Athens, Zografou GR 157 84, Greece*
[11] *Universidad del Atlántico, Barranquilla, Atlántico, Colombia*
[12] *Augustana University, Sioux Falls, SD 57197, USA*
[13] *University of Basel, CH-4056 Basel, Switzerland*
[14] *University of Bern, CH-3012 Bern, Switzerland*
[15] *Beykent University, Istanbul, Turkey*
[16] *University of Birmingham, Birmingham B15 2TT, United Kingdom*
[17] *Università del Bologna, 40127 Bologna, Italy*
[18] *Boston University, Boston, MA 02215, USA*
[19] *University of Bristol, Bristol BS8 1TL, United Kingdom*
[20] *Brookhaven National Laboratory, Upton, NY 11973, USA*
[21] *University of Bucharest, Bucharest, Romania*
[22] *University of California Berkeley, Berkeley, CA 94720, USA*
[23] *University of California Davis, Davis, CA 95616, USA*
[24] *University of California Irvine, Irvine, CA 92697, USA*
[25] *University of California Los Angeles, Los Angeles, CA 90095, USA*
[26] *University of California Riverside, Riverside CA 92521, USA*
[27] *University of California Santa Barbara, Santa Barbara, California 93106 USA*
[28] *California Institute of Technology, Pasadena, CA 91125, USA*
[29] *University of Cambridge, Cambridge CB3 0HE, United Kingdom*
[30] *Universidade Estadual de Campinas, Campinas - SP, 13083-970, Brazil*
[31] *Università di Catania, 2 - 95131 Catania, Italy*
[32] *Universidad Católica del Norte, Antofagasta, Chile*
[33] *Centro Brasileiro de Pesquisas Físicas, Rio de Janeiro, RJ 22290-180, Brazil*
[34] *IRFU, CEA, Université Paris-Saclay, F-91191 Gif-sur-Yvette, France*
[35] *CERN, The European Organization for Nuclear Research, 1211 Meyrin, Switzerland*
[36] *Institute of Particle and Nuclear Physics of the Faculty of Mathematics and Physics of the Charles University, 180 00 Prague 8, Czech Republic*
[37] *University of Chicago, Chicago, IL 60637, USA*

[38]*Chung-Ang University, Seoul 06974, South Korea*
[39]*CIEMAT, Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas, E-28040 Madrid, Spain*
[40]*University of Cincinnati, Cincinnati, OH 45221, USA*
[41]*Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (Cinvestav), Mexico City, Mexico*
[42]*Universidad de Colima, Colima, Mexico*
[43]*University of Colorado Boulder, Boulder, CO 80309, USA*
[44]*Colorado State University, Fort Collins, CO 80523, USA*
[45]*Columbia University, New York, NY 10027, USA*
[46]*Comisión Nacional de Investigación y Desarrollo Aeroespacial, Lima, Peru*
[47]*Centro de Tecnologia da Informacao Renato Archer, Amarais - Campinas, SP - CEP 13069-901*
[48]*Central University of South Bihar, Gaya, 824236, India*
[49]*Institute of Physics, Czech Academy of Sciences, 182 00 Prague 8, Czech Republic*
[50]*Czech Technical University, 115 19 Prague 1, Czech Republic*
[51]*Dakota State University, Madison, SD 57042, USA*
[52]*University of Dallas, Irving, TX 75062-4736, USA*
[53]*Laboratoire d'Annecy de Physique des Particules, Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, LAPP-IN2P3, 74000 Annecy, France*
[54]*Daresbury Laboratory, Cheshire WA4 4AD, United Kingdom*
[55]*Drexel University, Philadelphia, PA 19104, USA*
[56]*Duke University, Durham, NC 27708, USA*
[57]*Durham University, Durham DH1 3LE, United Kingdom*
[58]*University of Edinburgh, Edinburgh EH8 9YL, United Kingdom*
[59]*Universidad EIA, Envigado, Antioquia, Colombia*
[60]*ETH Zurich, Zurich, Switzerland*
[61]*Eötvös Loránd University, 1053 Budapest, Hungary*
[62]*Faculdade de Ciências da Universidade de Lisboa - FCUL, 1749-016 Lisboa, Portugal*
[63]*Universidade Federal de Alfenas, Poços de Caldas - MG, 37715-400, Brazil*
[64]*Universidade Federal de Goias, Goiania, GO 74690-900, Brazil*
[65]*Universidade Federal do ABC, Santo André - SP, 09210-580, Brazil*
[66]*Universidade Federal do Rio de Janeiro, Rio de Janeiro - RJ, 21941-901, Brazil*
[67]*Fermi National Accelerator Laboratory, Batavia, IL 60510, USA*
[68]*University of Ferrara, Ferrara, Italy*
[69]*University of Florida, Gainesville, FL 32611-8440, USA*
[70]*Florida State University, Tallahassee, FL, USA*
[71]*Fluminense Federal University, 9 Icaraí Niterói - RJ, 24220-900, Brazil*
[72]*Università degli Studi di Genova, Genova, Italy*
[73]*Georgian Technical University, Tbilisi, Georgia*
[74]*University of Granada & CAFPE, 18002 Granada, Spain*
[75]*Gran Sasso Science Institute, L'Aquila, Italy*
[76]*Laboratori Nazionali del Gran Sasso, L'Aquila AQ, Italy*
[77]*University Grenoble Alpes, CNRS, Grenoble INP, LPSC-IN2P3, 38000 Grenoble, France*
[78]*Universidad de Guanajuato, Guanajuato, C.P. 37000, Mexico*
[79]*Harish-Chandra Research Institute, Jhunsi, Allahabad 211 019, India*
[80]*Harvard University, Cambridge, MA 02138, USA*
[81]*University of Hawaii, Honolulu, HI 96822, USA*
[82]*University of Houston, Houston, TX 77204, USA*
[83]*University of Hyderabad, Gachibowli, Hyderabad - 500 046, India*
[84]*Idaho State University, Pocatello, ID 83209, USA*
[85]*Institut de Física d'Altes Energies (IFAE)—Barcelona Institute of Science and Technology (BIST), Barcelona, Spain*
[86]*Instituto de Física Corpuscular, CSIC and Universitat de València, 46980 Paterna, Valencia, Spain*
[87]*Instituto Galego de Fisica de Altas Enerxias, A Coruña, Spain*
[88]*Illinois Institute of Technology, Chicago, IL 60616, USA*
[89]*Imperial College of Science Technology and Medicine, London SW7 2BZ, United Kingdom*
[90]*Indian Institute of Technology Guwahati, Guwahati, 781 039, India*
[91]*Indian Institute of Technology Hyderabad, Hyderabad, 502285, India*
[92]*Indiana University, Bloomington, IN 47405, USA*
[93]*Istituto Nazionale di Fisica Nucleare Sezione di Bologna, 40127 Bologna BO, Italy*
[94]*Istituto Nazionale di Fisica Nucleare Sezione di Catania, I-95123 Catania, Italy*
[95]*Istituto Nazionale di Fisica Nucleare Sezione di Ferrara, I-44122 Ferrara, Italy*
[96]*Istituto Nazionale di Fisica Nucleare Sezione di Genova, 16146 Genova GE, Italy*
[97]*Istituto Nazionale di Fisica Nucleare Sezione di Lecce, 73100 - Lecce, Italy*
[98]*Istituto Nazionale di Fisica Nucleare Sezione di Milano Bicocca, 3 - I-20126 Milano, Italy*
[99]*Istituto Nazionale di Fisica Nucleare Sezione di Milano, 20133 Milano, Italy*
[100]*Istituto Nazionale di Fisica Nucleare Sezione di Napoli, I-80126 Napoli, Italy*

[101]*Istituto Nazionale di Fisica Nucleare Sezione di Padova, 35131 Padova, Italy*
[102]*Istituto Nazionale di Fisica Nucleare Sezione di Pavia, I-27100 Pavia, Italy*
[103]*Istituto Nazionale di Fisica Nucleare Laboratori Nazionali di Pisa, Pisa PI, Italy*
[104]*Istituto Nazionale di Fisica Nucleare Sezione di Roma, 00185 Roma RM, Italy*
[105]*Istituto Nazionale di Fisica Nucleare Laboratori Nazionali del Sud, 95123 Catania, Italy*
[106]*Universidad Nacional de Ingeniería, Lima 25, Perú*
[107]*Institute for Nuclear Research of the Russian Academy of Sciences, Moscow 117312, Russia*
[108]*University of Insubria, Via Ravasi, 2, 21100 Varese VA, Italy*
[109]*University of Iowa, Iowa City, IA 52242, USA*
[110]*Iowa State University, Ames, Iowa 50011, USA*
[111]*Institut de Physique des 2 Infinis de Lyon, 69622 Villeurbanne, France*
[112]*Institute for Research in Fundamental Sciences, Tehran, Iran*
[113]*Instituto Superior Técnico - IST, Universidade de Lisboa, Portugal*
[114]*Instituto Tecnológico de Aeronáutica, Sao Jose dos Campos, Brazil*
[115]*Iwate University, Morioka, Iwate 020-8551, Japan*
[116]*University of Jammu, Jammu-180006, India*
[117]*Jawaharlal Nehru University, New Delhi 110067, India*
[118]*Jeonbuk National University, Jeonrabuk-do 54896, South Korea*
[119]*Joint Institute for Nuclear Research, Dzhelepov Laboratory of Nuclear Problems 6 Joliot-Curie, Dubna, Moscow Region, 141980 RU*
[120]*University of Jyvaskyla, FI-40014, Finland*
[121]*Kansas State University, Manhattan, KS 66506, USA*
[122]*Kavli Institute for the Physics and Mathematics of the Universe, Kashiwa, Chiba 277-8583, Japan*
[123]*High Energy Accelerator Research Organization (KEK), Ibaraki, 305-0801, Japan*
[124]*Korea Institute of Science and Technology Information, Daejeon, 34141, South Korea*
[125]*K L University, Vaddeswaram, Andhra Pradesh 522502, India*
[126]*National Institute of Technology, Kure College, Hiroshima, 737-8506, Japan*
[127]*Taras Shevchenko National University of Kyiv, 01601 Kyiv, Ukraine*
[128]*Lancaster University, Lancaster LA1 4YB, United Kingdom*
[129]*Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*
[130]*Laboratório de Instrumentação e Física Experimental de Partículas, 1649-003 Lisboa and 3004-516 Coimbra, Portugal*
[131]*University of Liverpool, L69 7ZE, Liverpool, United Kingdom*
[132]*Los Alamos National Laboratory, Los Alamos, NM 87545, USA*
[133]*Louisiana State University, Baton Rouge, LA 70803, USA*
[134]*University of Lucknow, Uttar Pradesh 226007, India*
[135]*Madrid Autonoma University and IFT UAM/CSIC, 28049 Madrid, Spain*
[136]*Johannes Gutenberg-Universität Mainz, 55122 Mainz, Germany*
[137]*University of Manchester, Manchester M13 9PL, United Kingdom*
[138]*Massachusetts Institute of Technology, Cambridge, MA 02139, USA*
[139]*Max-Planck-Institut, Munich, 80805, Germany*
[140]*University of Medellín, Medellín, 050026 Colombia*
[141]*University of Michigan, Ann Arbor, MI 48109, USA*
[142]*Michigan State University, East Lansing, MI 48824, USA*
[143]*Università del Milano-Bicocca, 20126 Milano, Italy*
[144]*Università degli Studi di Milano, I-20133 Milano, Italy*
[145]*University of Minnesota Duluth, Duluth, MN 55812, USA*
[146]*University of Minnesota Twin Cities, Minneapolis, MN 55455, USA*
[147]*University of Mississippi, University, MS 38677 USA*
[148]*Università degli Studi di Napoli Federico II , 80138 Napoli NA, Italy*
[149]*University of New Mexico, Albuquerque, NM 87131, USA*
[150]*H. Niewodniczański Institute of Nuclear Physics, Polish Academy of Sciences, Cracow, Poland*
[151]*Nikhef National Institute of Subatomic Physics, 1098 XG Amsterdam, Netherlands*
[152]*University of North Dakota, Grand Forks, ND 58202-8357, USA*
[153]*Northern Illinois University, DeKalb, IL 60115, USA*
[154]*Northwestern University, Evanston, Il 60208, USA*
[155]*University of Notre Dame, Notre Dame, IN 46556, USA*
[156]*University of Novi Sad, 21102 Novi Sad, Serbia*
[157]*Occidental College, Los Angeles, CA 90041*
[158]*Ohio State University, Columbus, OH 43210, USA*
[159]*Oregon State University, Corvallis, OR 97331, USA*
[160]*University of Oxford, Oxford, OX1 3RH, United Kingdom*
[161]*Pacific Northwest National Laboratory, Richland, WA 99352, USA*
[162]*Universtà degli Studi di Padova, I-35131 Padova, Italy*
[163]*Panjab University, Chandigarh, 160014 U.T., India*

[164] *Université Paris-Saclay, CNRS/IN2P3, IJCLab, 91405 Orsay, France*
[165] *Université Paris Cité, CNRS, Astroparticule et Cosmologie, Paris, France*
[166] *University of Parma, 43121 Parma PR, Italy*
[167] *Università degli Studi di Pavia, 27100 Pavia PV, Italy*
[168] *University of Pennsylvania, Philadelphia, PA 19104, USA*
[169] *Pennsylvania State University, University Park, PA 16802, USA*
[170] *Physical Research Laboratory, Ahmedabad 380 009, India*
[171] *Università di Pisa, I-56127 Pisa, Italy*
[172] *University of Pittsburgh, Pittsburgh, PA 15260, USA*
[173] *Pontificia Universidad Católica del Perú, Lima, Perú*
[174] *University of Puerto Rico, Mayaguez 00681, Puerto Rico, USA*
[175] *Punjab Agricultural University, Ludhiana 141004, India*
[176] *Queen Mary University of London, London E1 4NS, United Kingdom*
[177] *Radboud University, NL-6525 AJ Nijmegen, Netherlands*
[178] *University of Rochester, Rochester, NY 14627, USA*
[179] *Royal Holloway College London, TW20 0EX, United Kingdom*
[180] *Rutgers University, Piscataway, NJ, 08854, USA*
[181] *STFC Rutherford Appleton Laboratory, Didcot OX11 0QX, United Kingdom*
[182] *Università del Salento, 73100 Lecce, Italy*
[183] *San Jose State University, San José, CA 95192-0106, USA*
[184] *Sapienza University of Rome, 00185 Roma RM, Italy*
[185] *Universidad Sergio Arboleda, 11022 Bogotá, Colombia*
[186] *University of Sheffield, Sheffield S3 7RH, United Kingdom*
[187] *SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA*
[188] *University of South Carolina, Columbia, SC 29208, USA*
[189] *South Dakota School of Mines and Technology, Rapid City, SD 57701, USA*
[190] *South Dakota State University, Brookings, SD 57007, USA*
[191] *Southern Methodist University, Dallas, TX 75275, USA*
[192] *Stony Brook University, SUNY, Stony Brook, NY 11794, USA*
[193] *Sun Yat-Sen University, Guangzhou, 510275*
[194] *Sanford Underground Research Facility, Lead, SD, 57754, USA*
[195] *University of Sussex, Brighton, BN1 9RH, United Kingdom*
[196] *Syracuse University, Syracuse, NY 13244, USA*
[197] *Universidade Tecnológica Federal do Paraná, Curitiba, Brazil*
[198] *Texas A&M University, College Station, Texas 77840*
[199] *Texas A&M University - Corpus Christi, Corpus Christi, TX 78412, USA*
[200] *University of Texas at Arlington, Arlington, TX 76019, USA*
[201] *University of Texas at Austin, Austin, TX 78712, USA*
[202] *University of Toronto, Toronto, Ontario M5S 1A1, Canada*
[203] *Tufts University, Medford, MA 02155, USA*
[204] *Universidade Federal de São Paulo, 09913-030, São Paulo, Brazil*
[205] *Ulsan National Institute of Science and Technology, Ulsan 689-798, South Korea*
[206] *University College London, London, WC1E 6BT, United Kingdom*
[207] *Valley City State University, Valley City, ND 58072, USA*
[208] *Variable Energy Cyclotron Centre, 700 064 West Bengal, India*
[209] *Virginia Tech, Blacksburg, VA 24060, USA*
[210] *University of Warsaw, 02-093 Warsaw, Poland*
[211] *University of Warwick, Coventry CV4 7AL, United Kingdom*
[212] *Wellesley College, Wellesley, MA 02481, USA*
[213] *Wichita State University, Wichita, KS 67260, USA*
[214] *William and Mary, Williamsburg, VA 23187, USA*
[215] *University of Wisconsin Madison, Madison, WI 53706, USA*
[216] *Yale University, New Haven, CT 06520, USA*
[217] *Yerevan Institute for Theoretical Physics and Modeling, Yerevan 0036, Armenia*
[218] *York University, Toronto M3J 1P3, Canada*

Contents

List of Figures

List of Tables

Part I

Overview

Chapter 1

Introduction

This document describes Offline Software and Computing for the Deep Underground Neutrino Experiment (DUNE) experiment, in particular, the conceptual design of the offline computing needed to accomplish its physics goals. Our emphasis in this document is the development of the computing infrastructure needed to acquire, catalog, reconstruct, simulate and analyze the data from the DUNE experiment and its prototypes. In this effort, we concentrate on developing the tools and systems that facilitate the development and deployment of advanced algorithms. Rather than prescribing particular algorithms, our goal is to provide resources that are flexible and accessible enough to support creative software solutions as HEP computing evolves and to provide computing that achieves the physics goals of the DUNE experiment.

This chapter provides an introduction to the DUNE experiment. The last section (Section 1.5) summarizes the physics drivers of offline computing challenges that inform the rest of this document.

1.1 Introduction

DUNE will begin running in the late 2020's. The goals of the experiment include 1) studying neutrino oscillations using a beam of neutrinos sent from Fermi National Accelerator Laboratory (Fermilab) in Illinois to the Sanford Underground Research Facility (SURF) in Lead, South Dakota, 2) studying astrophysical neutrino sources and rare processes and 3) understanding the physics of neutrino interactions in matter. DUNE will consist of a modular far detector (FD) located about 1.5 km underground at SURF in South Dakota, USA, 1300 km from Fermilab, and a near detector (ND) located on site at Fermilab in Illinois. The DUNE detectors will be exposed to the world's most intense neutrino beam originating at Fermilab. A high-precision near detector, 574 m from the neutrino source on the Fermilab site, will be used to characterize the intensity and energy spectrum of this wide-band beam. The overriding physics goals of the DUNE experiment are the search for leptonic charge conjugation and parity (CP) violation, the search for nucleon decay as a signature of a Grand Unified Theory underlying the Standard Model, the observation of supernova neutrino bursts (SNBs) from supernovae in our galaxy, and studies of solar neutrinos.

This document concentrates on the neutrino oscillation and supernova capabilities of the experiment

as their needs for high data volumes and high-precision measurements at low interaction energies drive the computing needs for the experiment.

When produced, the neutrino beam from Fermilab will consist almost entirely of muon-type neutrinos. Neutrinos are known to come in (at least) three flavors that can be distinguished by their interactions – electron neutrinos produce electrons when they interact via charged currents; muon neutrinos, muons; and tau neutrinos, tau particles. But these flavors do not correspond to fixed mass states. All three flavors of neutrinos are mixtures of mass states, much as light polarized in the x direction can be considered a superposition of x' and y' polarizations along alternate axes rotated by 45 degrees. When neutrinos propagate through space, it is the mass state that sets their wavelength and if the neutrino goes far enough, the multiple mass states corresponding to the initial flavor state will get out of phase. When the resulting mixed state is later probed about its flavor, the detected flavor may be different than it was when the neutrino was produced. This phenomenon is known as neutrino oscillation, and has been observed in multiple experiments since it was first confirmed in 1998[1].

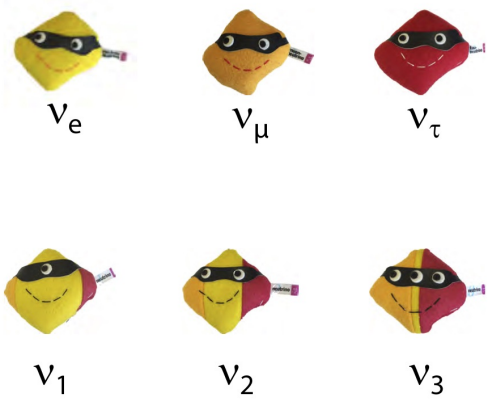


Figure 1.1: Illustration of the neutrino flavor and mass states. The mass states are a superposition of the flavor states. Courtesy the particlezoo.net.

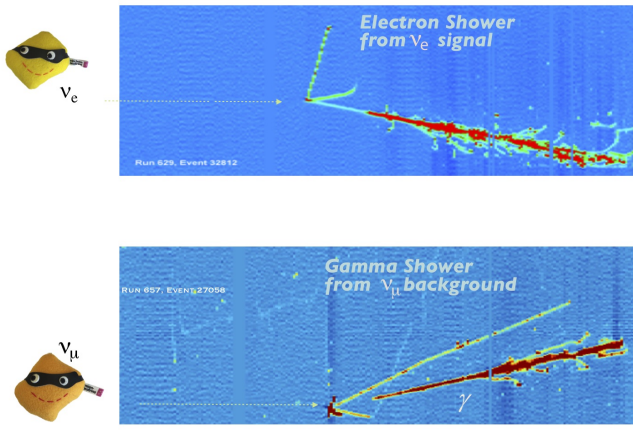


Figure 1.2: Electron neutrino appearance signal (top) and background (bottom) as seen in the ArgoNeut experiment[2]. In the true appearance signal, an electron is seen emerging from the primary vertex, then showering. In the background interaction, a muon neutrino enters and produces a final-state muon and photons that propagate some distance before showering.

DUNE, in particular, wishes to understand the conversion of the muon neutrinos created in Illinois into electron neutrinos at the FD in South Dakota and compare that conversion rate between neutrino and antineutrino beams. The location of the FD and energy of the neutrino beam were chosen to maximize the oscillation effect. A difference in the conversion rate for neutrinos and antineutrinos could be evidence for matter-antimatter asymmetry in the neutrino sector, a phenomenon called charge-parity symmetry violation (CPV).

To make these measurements, the experiment must be able to distinguish electron-neutrino interactions from the dominant muon-neutrino interactions one would expect in the absence of oscillations. Doing this requires a very large detector, as neutrino interactions are intrinsically rare, but also an extremely fine-grained one, as well. Noble liquid time projection chambers (TPCs), which read out large transparent volumes of liquid by drifting the electrons produced when charged particles from neutrino interactions ionize the liquid to charge-sensitive detectors through strong electric fields (E fields), have the needed capabilities of extremely large scale and fine-grained resolution. The TPCs are augmented by photon detection systems that provide precise timing of interactions and reconstructed particles. The proposed DUNE far detector will instrument four $14\,\text{m} \times 12\,\text{m} \times 58\,\text{m}$ volumes of liquid argon (LAr) with readout granularity of $\sim$0.5 cm. The FD modules will be located 4850 ft below the earth's surface to reduce the rate of cosmic rays traversing the detector by orders of magnitude and thus allow sensitivity to very low-energy solar and astrophysical neutrinos, as well as the higher-energy neutrinos produced in the beam at Fermilab. See the FD and ND design reports[3, 4, 5] for full descriptions of the liquid argon time-projection chamber (LArTPC) technology.

Additionally, physics programs focused on nucleon decay, the detection of a SNB, and other Beyond Standard Model (BSM) signatures take advantage of the large size of the detector and flexible readout window of the data acquisition (DAQ) and LArTPC.

The neutrino beam from Fermilab will be pulsed approximately once per second, 24 hours per day during running periods, with 15 million pulses per year. Because neutrinos interact extremely rarely, we expect to detect of order 7,000 neutrino interactions/year in each of the four planned 10 kt far detector modules located at the FD site in South Dakota.[1].

A full TDR for the first far detector module, which uses horizontal drift technology (FD1-HD), is available in References [6, 7, 8, 9]. A CDR for a second detector module that implements vertical drift technology (FD2-VD) is available in Reference [10]. This module is planned to come online soon after the FD1-HD module. Computing needs for both of these modules are included in the present document. The first two far detector modules should go live late in this decade with commissioning of the DAQ systems expected to start in 2027.

1.2 ProtoDUNE Tests at CERN

Building an experiment of this size requires an extensive period of prototyping. The Argoneut[11], MicroBooNE[12] and ICARUS[13] collaborations have demonstrated the capabilities of LArTPCs for neutrino detection on scales between 1 and 500 metric tons (tonnes, t) of fiducial mass. In preparation for the DUNE experiment, a campaign for testing full-sized FD components in 700 tonne capacity

[1] This is based on the beam repetition rate of 0.83 Hz and an estimated uptime for the accelerator complex of 56% [6].

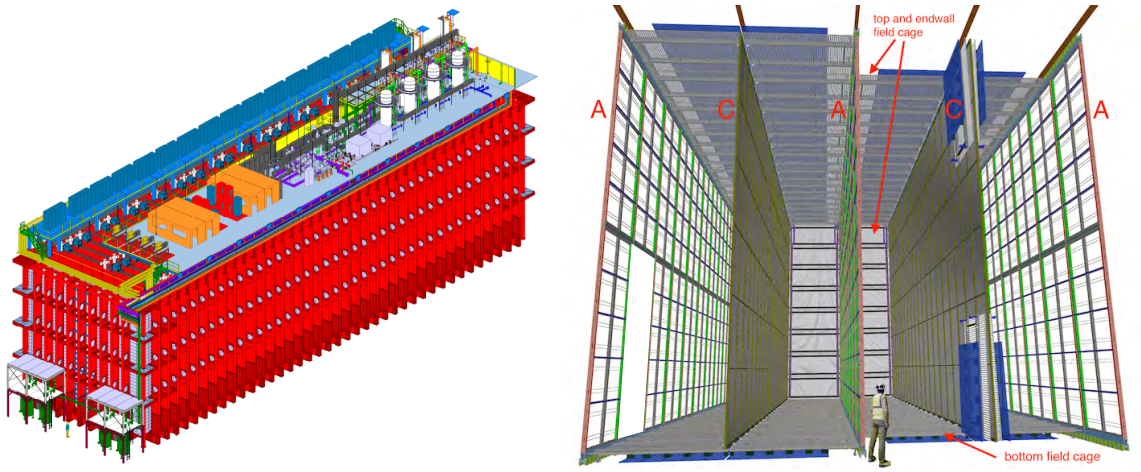


Figure 1.3: (Left) A far detector cryostat that houses a 10 kt FD module with horizontal drift technology. The figure of a person at the bottom left of the image indicates the scale. (Right) A 10 kt DUNE FD SP module, showing the alternating 58 m long (into the page), 12 m high anode (A) and cathode (C) planes, as well as the field cage that surrounds the drift regions between the anode and cathode planes. The modular anode and cathode planes are constructed of units called anode plane assemblies (APAs) and cathode plane assemblies (CPAs); the blank area on the left side was added to show the profile of a single APA.

cryostats in the Experiment Hall North One (EHN1) hadronic test beams at the European Laboratory for Particle Physics (CERN) was launched in 2018. Both single-phase (SP) (horizontal drift) and dual-phase (DP) (liquid and gas, vertical drift) prototypes, called ProtoDUNE-SP and ProtoDUNE-DP respectively, were constructed and operated. The complete data-taking chain from detector construction to full offline reconstruction and analysis of data was tested, and the results provided considerable insight into the computing challenges for the full DUNE experiment.

1.2.1 ProtoDUNE-SP

The ProtoDUNE-SP experiment[14], located at the CERN Neutrino Platform within EHN1, began taking data in late 2018. ProtoDUNE-SP collected ionization electrons and scintillation light directly from the LAr. The readout system consists of anode plane assemblies (APAs) and a Photon Detection System (PDS). Each APA consists of an aluminum frame with three layers of active wires that form a grid on each side of the APA, with each layer strung at angles chosen to reduce ambiguities in event reconstruction. Based on the drift field and the layer separation, the relative voltage between the layers is chosen to ensure transparency of the first two layers (U and V) to the drifting electrons. These layers produce bipolar induction signals as the electrons pass through them. The final layer (Y) collects the drifting electrons, resulting in a unipolar signal. The drift time of the electrons provides a measurement of the x-coordinate of ionization while the z-coordinate is determined from the vertical Y wires. The U and V layers are at $+/-60$ degrees and the combination of U, V, and Y ionization pattern collected on the grid of anode wires provides the reconstruction in the remaining coordinate perpendicular to the drift direction. The integrated area of the wire signal is proportional to the collected ionization charge. Figure 1.4 illustrates the principle of operation.

FD1-HD, the first of the four FD modules and described in Section 1.3, will use horizontal-drift SP technology with APA readout, similar to that used in ProtoDUNE-SP.

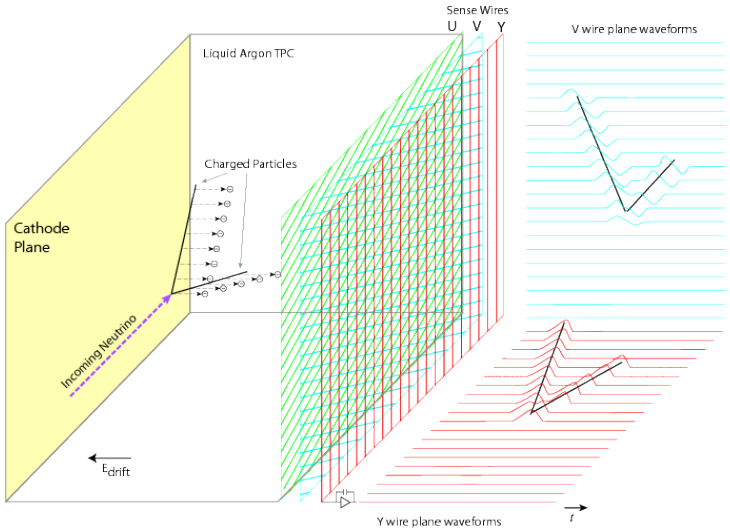


Figure 1.4: Diagram from [15] illustrating the signal formation in a LArTPC with three wire planes [16]. For simplicity, the signal in the first (U) induction plane is omitted in the illustration.

The ProtoDUNE-SP detector [14, 17], immersed in a 770 t volume of liquid argon, includes a cathode plane in the center and sets of three anode plane assemblies mounted on the sides of the liquid volume opposing the cathode, creating equal and opposite horizontal electric (E) fields on the two sides of the cathode plane. The maximum drift distance is 3 m with a nominal voltage of 150 kV across that distance and a field strength of 500 V/cm. Each APA has 2560 channels and each channel reads out a 12 bit analog-to-digital converter (ADC) every 0.5 μsec. For ProtoDUNE-SP the readout time appropriate for a 3 m drift was set to 3 msec, resulting in 6000 12 bit samples per channel for every readout window. The total data size for six anode plane assemblies is thus 140 MB with additional header data and data from photon detectors and external tagging systems (details available in [14]), bringing the nominal trigger record size to about 180 MB. Lossless compression of the TPC readout data was implemented in the DAQ, resulting in a final compressed trigger record size of about 75 MB.

All or part of the detector will be read out for time intervals of a few ms herein called a "trigger record" or "event" triggered by beam and/or activity in the FD. DUNE prefers to use the term "trigger record" rather than "event" for the unit of processing, as it avoids confusion between an interaction "event" and a readout "event."

The ProtoDUNE-SP test beam ran at rates of up to 25 Hz over a period of six weeks at beam momenta between 0.5 and 7 GeV/c. Time-of-flight and Cherenkov counters in the beamline provided beam flavor tagging. Around 8M total "physics" records were written, with around 3M having beam tag information. In total 850 TB of raw test beam data were written, along with 1 PB of commissioning and cosmic data. These data were successfully cataloged and written to storage at both CERN and Fermilab at rates of up to 2 GB/sec.

Thanks to significant prior effort in the LArTPC computing and algorithms community, reconstruction software was quickly developed and production workflows integrated. Keep-up processing of the first reconstruction pass began soon after data taking started, and two weeks after the end of data taking

was complete. These results were extremely useful in demonstrating the capabilities of the detector; they are summarized in Volume II of the FD1-HD TDR [6]. A second pass, with improved treatment of instrumental effects ranging from sticky codes, 2D deconvolution, and correction for space charge effects was completed in late 2019. Another pass with major improvements to the electrostatic modeling and reconstruction algorithms was completed in 2021.

Below are three figures that show examples of the output of production processing algorithms. Figure 1.5 illustrates the signal processing stage of reconstruction, where raw ADC signals have noise and sticky codes removed and are then deconvolved to yield Gaussian hit candidates. This impacts the reconstruction algorithms and framework processing. Figures 1.6 and 1.7 illustrate full pattern recognition and event reconstruction, respectively.

Figure 1.5 illustrates the result of the first stage of reconstruction which has two main parts. The first applies digital noise filtering techniques to reduce the effects of thermal and coherent noise (and to counteract ADC imperfections in a small ($\sim 1\%$) of channels). The second stage performs further signal processing in two steps. First, a model of the detector response is deconvolved from the noise filtered waveforms. Importantly, this deconvolution is across both the longitudinal drift time dimension and across the 'channel direction'. This is required as electrons drifting near wires will induce measurable current not just on the wire of closest approach but across multiple neighboring wires and over a drift time of about 100 μs. This deconvolution process inherently amplifies low frequency noise. To combat this, a second step locates signal regions of interest and over that region the waveform baseline is recalculated. These signal waveform regions are then input to all subsequent pattern recognition and event reconstruction as illustrated in Figures 1.6 and 1.7.

Compressed raw input trigger records were of order 75 MB in size and took 500-600 seconds to reconstruct, of which around 180 s was signal processing and the remainder high-level reconstruction dominated by 40-60 cosmic rays per readout. Memory footprints for data processing ranged between 2.5 and 4 GB. Output record sizes were reduced to 35 MB by dropping the raw waveforms after hit finding. Data reconstruction campaigns took of order 4-6 weeks (similar to the original data taking) and utilized up to 15,000 cores on Open Science Grid (OSG) and Worldwide LHC Computing Grid (WLCG) resources. Job submission was done through the Production Operations Management System (POMS)[18] job management system developed at Fermilab. POMS supports submissions to Fermilab-dedicated resources and selected OSG and WLCG sites. Figure 1.8 shows the distribution of wall hours used for the first pass of reconstruction in 2019. These metrics have influenced the ideas regarding memory utilization, event data model, and workload/workflow management in this document.

For reconstruction, data were streamed via `xrootd`[19] from dCache storage at Fermilab (and in some cases CERN where a second copy was stored) to the remote sites. Despite individual processing jobs taking 15-30 hours to complete, network interruptions rarely caused job failures. These numbers and performance metrics are used later to predict future resources needs for the full DUNE experiment.

1.2.2 ProtoDUNE Dual-Phase and its Evolution to the Vertical Drift Design

In parallel with the single-phase horizontal-drift detector tests, a vertical drift DP readout prototype (ProtoDUNE-DP) in a similar cryostat was also tested. ProtoDUNE-DP was not exposed to beam but ran on cosmics in 2019 and 2020.

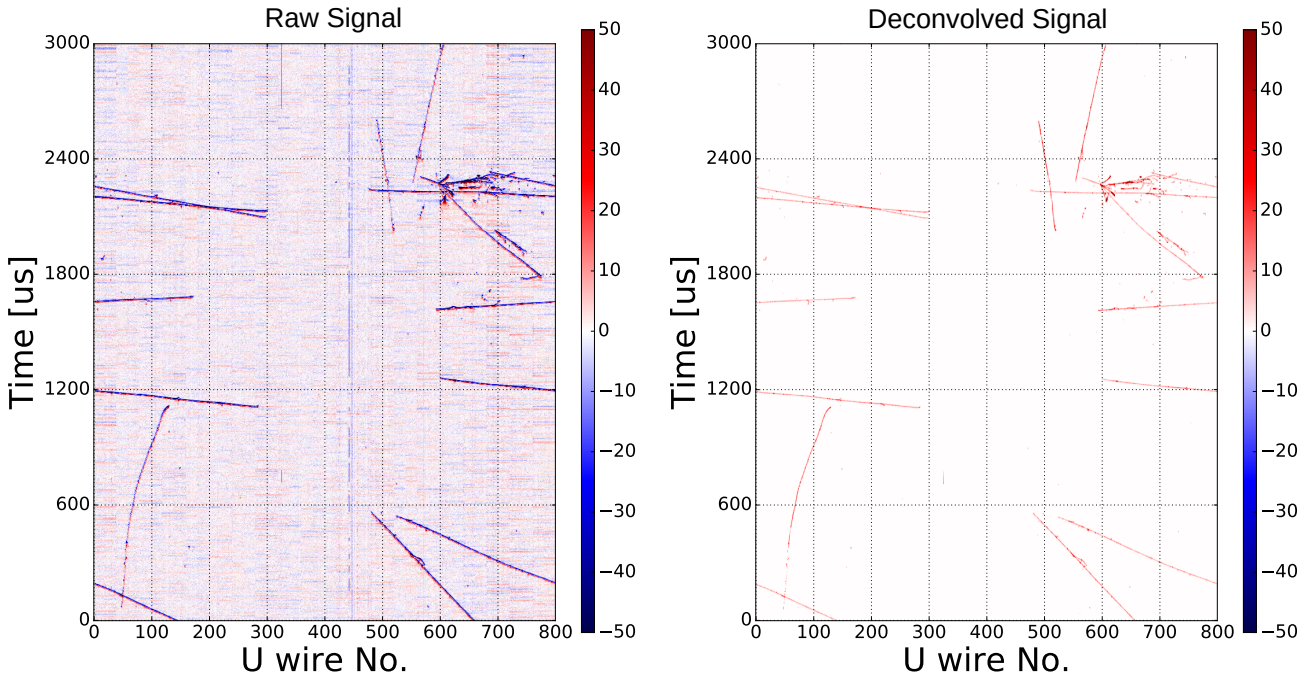


Figure 1.5: Comparison of raw (left) and deconvolved induction U-plane signals (right) before and after the signal processing procedure from a ProtoDUNE-SP trigger record. The bipolar shape with red (blue) color representing positive (negative) signals is converted to the unipolar shape after the 2D deconvolution.

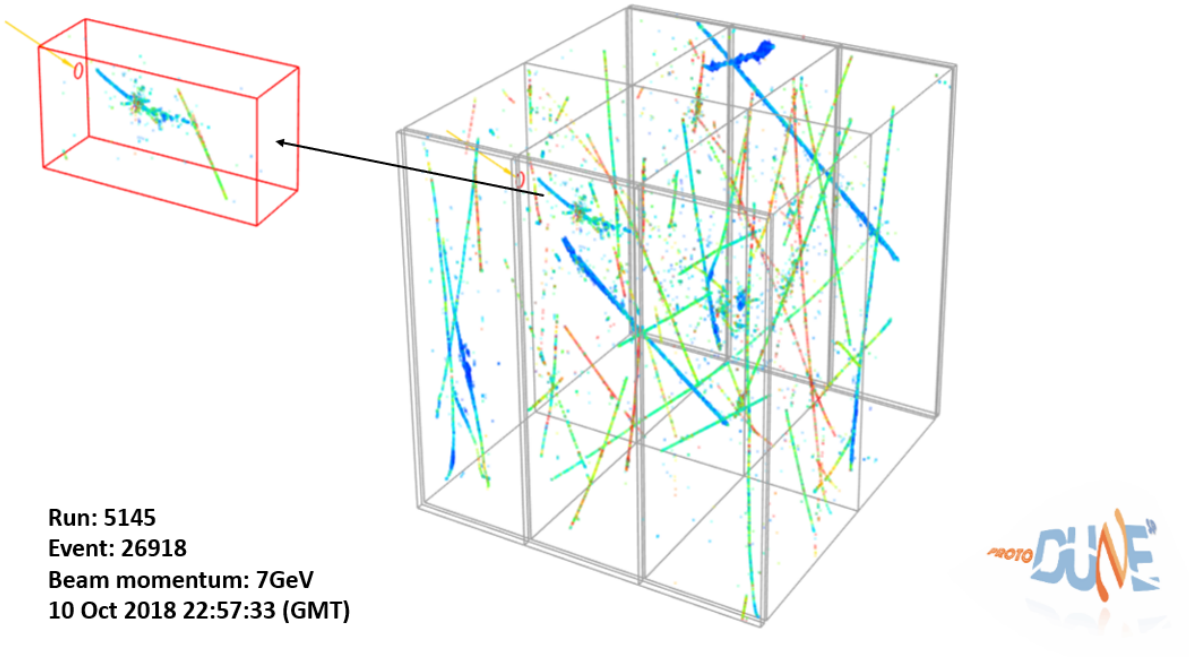


Figure 1.6: The ProtoDUNE-SP detector (gray box) showing the direction of the particle beam (yellow line on the very far left) and the outlines of the six anode plane assemblies. Cosmic rays can be seen throughout the white box, while the red box highlights the beam region of interest with an interaction of the 7 GeV beam. The 3D points are obtained using the Space Point Solver reconstruction algorithm.

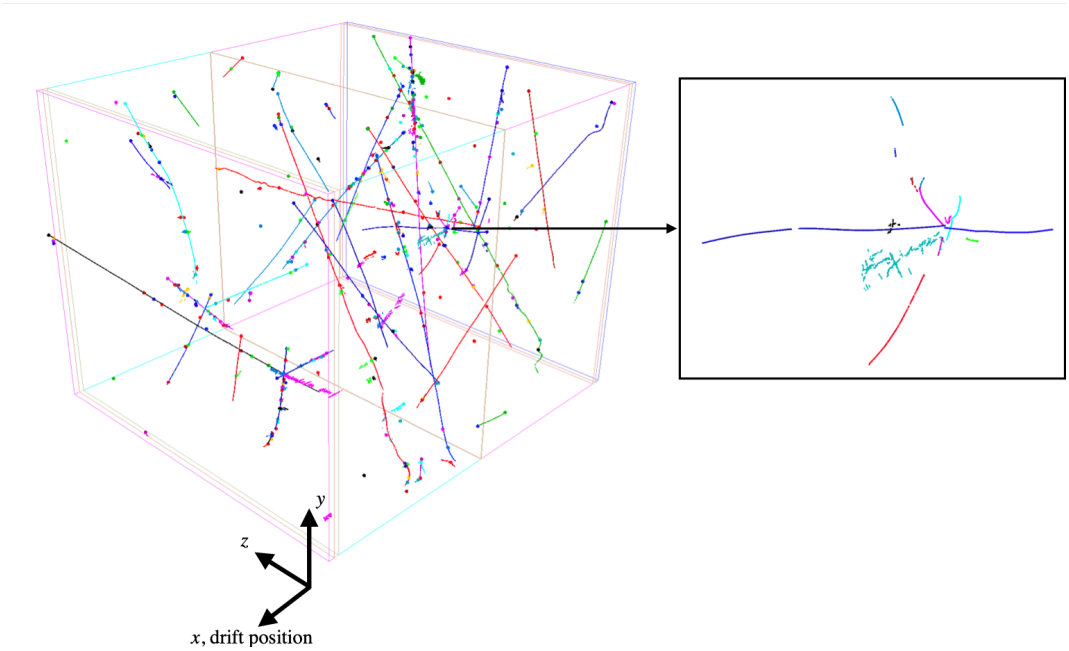


Figure 1.7: Pandora [20] reconstruction of cosmic rays and beam interaction in a ProtoDUNE-SP trigger record. The left side of the figure shows the full detector volume with all interactions, including cosmic rays, and the right side shows the identified beam interaction.

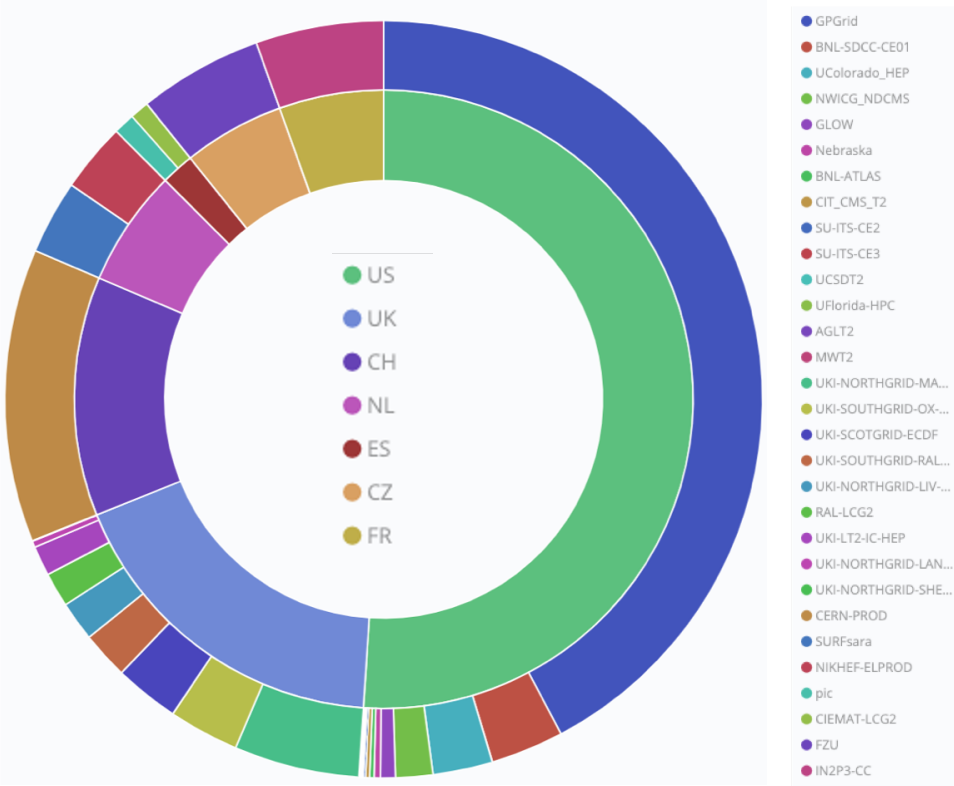


Figure 1.8: Reconstruction and simulation processing distribution across sites for DUNE production in calendar 2019. The inner circle shows national contributions while the outer circle shows individual site contributions.

The basic operating principle of ProtoDUNE-DP is shown in Figure 1.9. As in ProtoDUNE-SP, charged particles that traverse the active volume of the LArTPC ionize the medium while also producing scintillation light that is detected by a Photon Detection System. The ionization electrons drift vertically upward toward an extraction grid just below the liquid-vapor interface. After reaching the grid, an E field stronger than the drift field extracts the electrons from the liquid up into the gas phase. Once in the gas, electrons encounter micro-pattern gas detectors, called large electron multipliers (LEMs), with high-field regions. The LEMs amplify the electrons in avalanches that occur in these high-field regions. The amplified charge is then collected and recorded on a 2D anode consisting of two sets of 3.125 mm pitch gold-plated copper strips that provide the x and y coordinates (and thus two views) of an interaction.

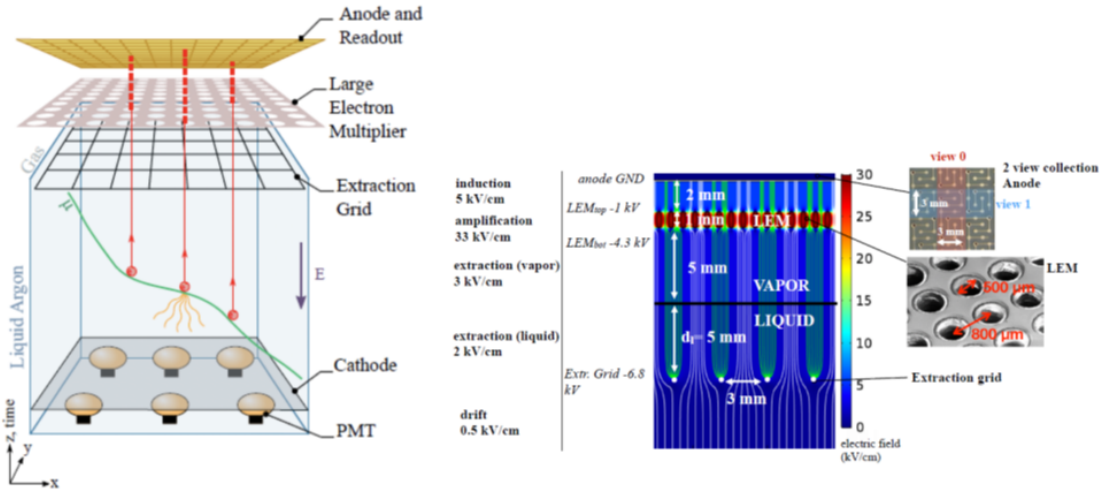


Figure 1.9: Principle of DP readout (left), and thicknesses and HV values for electron extraction from liquid to gaseous argon, their multiplication by LEMs, and their collection on the x and y readout anode plane (right) The HV values are indicated for a drift field of 0.5 kV/cm in LAr.

The readout area surface is 6 m × 6 m, subdivided into four 3×3 m^2 charge-readout planes (CRPs). Each CRP is an independent detector element that performs electron extraction, amplification, and collection. The 7680 readout channels are read by 12 bit ADCs every 0.4 μsec. The ProtoDUNE-DP detector consists of a 700 t volume of LAr, with a vertical drift length of 6 m, corresponding to a full drift window of 4 ms (10,000 samples).

The ProtoDUNE-DP detector began taking cosmic ray data in August 2019. Thanks to preceding data challenges, these data have been successfully integrated into the full data cataloging and reconstruction chain and were reconstructed as they became available. A total of 1.45M trigger records were collected; the size of the raw data files (run sequence files) was 3.2 GB, each file containing 30 trigger records. Cosmic ray data are displayed in Figure 1.10: on the left a horizontal muon track is shown with the corresponding waveform on a channel, giving an idea of the low noise conditions. A trigger record including an electromagnetic shower and two muon decays and a trigger record with an example of multiple hadronic interactions in a shower are shown on the right.

All data ($\sim$330 TB) taken during different campaigns have been copied to Fermilab. A subsample is composed of data sets taken during detector transient conditions, motivated by various specific testing

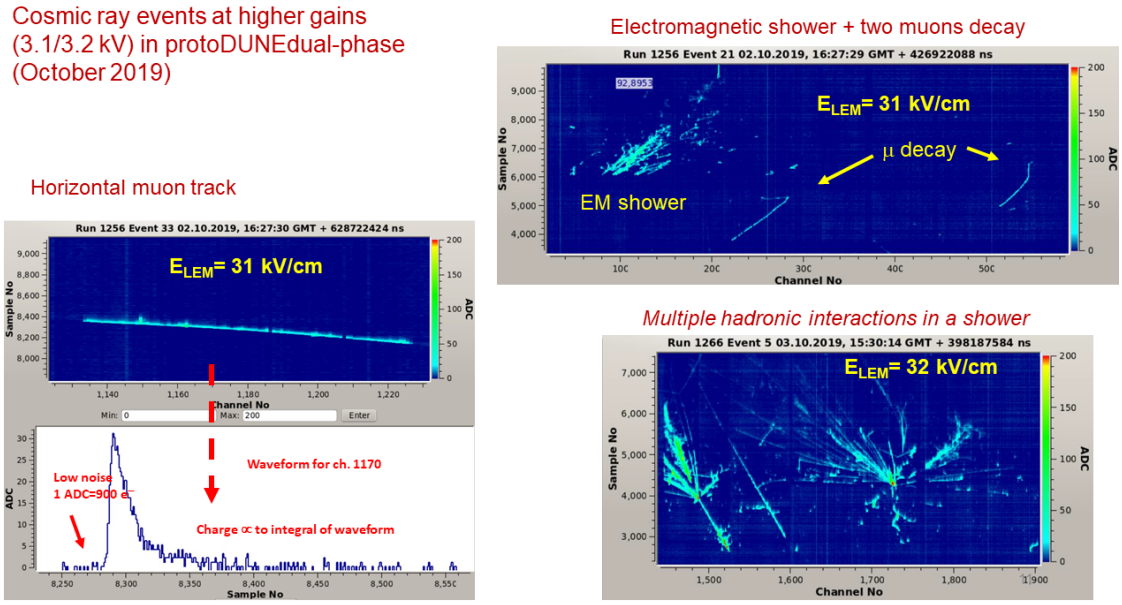


Figure 1.10: Cosmic ray data from the DP prototype, ProtoDUNE-DP.

needs; all cosmic ray data taken in well defined and stable detector conditions in 2019 and 2020 ($\approx$377K events) have been processed with Liquid Argon Software (LArSoft) by performing the reconstruction of hits and 2D tracks. A second pass, including Pandora reconstruction algorithms, started in spring 2021. The memory footprint is between 1.9 and 2.5 GB. Data management and job submission was successfully done through the same systems as ProtoDUNE-SP.

Experiences with the dual-phase CRP-readout and vertical-drift, and with the horizontal-drift single-phase APA-readout, motivated development of the single-phase CRP-readout, vertical-drift detector planned for the FD2-VD concept. The FD2-VD incorporates many of the design aspects developed for the DP, such as the CRPs; the main difference with respect to the DP design is the removal of the extraction stage to the gas phase and the subsequent charge amplification stage. This eliminates the grid biased at high voltage (to transfer the electrons from the liquid to the gas) and the LEMs used to amplify the signal in the gas. The FD2-VD CRPs perform charge readout using perforated PCB anodes with finely segmented strip electrodes that are immersed in the LAr. The CRPs operate in a conceptually similar manner to the HD anode plane assemblies in that there are two sets of induction strips and one set of collection strips. Each set of strips is constructed at an angle offset to the other two angles to provide disambiguation in both spatial dimensions.

Figure 1.11 illustrates the PCB-based charge readout concept for the VD detector. The electron drift direction is vertical. Two separate drift volumes of 6.5 m are defined by a cathode plane at roughly mid-height in the detector volume. Ionization electrons above the cathode will drift upwards; ionization electrons in the liquid below the cathode will drift downwards. The FD2-VD prototype components have undergone testing starting in 2021 with a small cold box; some examples of cosmics tracks are shown in figure 1.12. A full prototype, known as Module 0, is expected to be completed and installed inside the NP02 cryostat in 2022 and 2023. Long-term operation and full characterization with a charged particle beam and cosmics will follow.

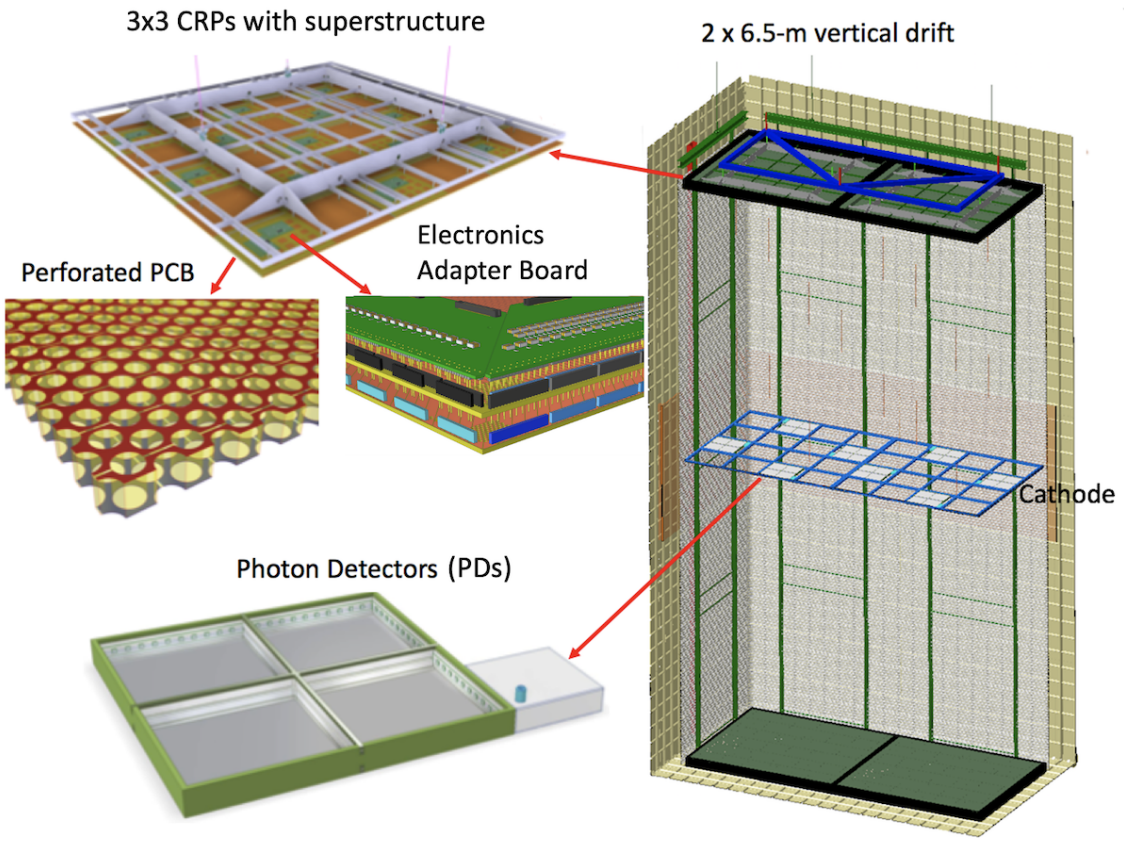


Figure 1.11: Vertical drift solution with PCB-based charge readout

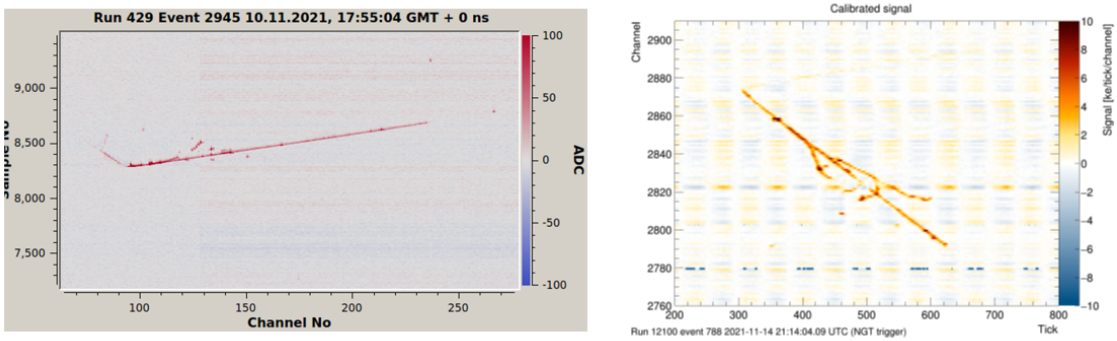


Figure 1.12: Examples of cosmic tracks in the Vertical Drift cold box as visualized with a raw-data event display (left) and after calibration has been applied (right).

1.2.3 Conclusions from Prototype Tests

ProtoDUNE runs are continuing and will continue through beam tests in 2022-23 at CERN. Data cataloging, movement and storage techniques were tested before the start of the 2018-2020 ProtoDUNE-SP and ProtoDUNE-DP runs and were able to handle the full rate of the experiments. Reconstruction algorithms were also in place on time and produced early results that led to increased understanding of the detector and improved calibrations for a second iteration. These tests also identified some deficiencies in our infrastructure, including incomplete schemes for the transmission of configuration and conditions information between hardware operations and offline computing. The first round of test beam runs have been extremely valuable; they helped identify which variables are important to transmit and they motivated the design of improved systems for gathering and storing that information.

An additional beam and cosmic ray run of ProtoDUNE-HD is planned for 2022-23, with beam and cosmic tests of the FD2-VD design to follow in 2023-24, allowing further development and testing of our computing infrastructure before the full detector comes online in the late 2020s. Additionally, there are ongoing smaller-scale readout and noise tests of both APA and CRP in cold boxes at CERN and small cryostats at Fermilab.

1.3 Far Detector

The full DUNE FD will begin with one FD1-HD module to be installed at SURF starting near the end of this decade. A second FD2-VD module will be installed and commissioned in parallel. High-intensity neutrino and antineutrino beams should arrive after a year or so of commissioning of the detector and the Long-Baseline Neutrino Facility (LBNF) beamline. The first module will be a scaled-up version of ProtoDUNE-SP with 150 anode plane assemblies stacked two anode plane assemblies deep the length of the cryostat, down the center and along the long walls. The argon volume will be $15 \times 14 \times 62\,\mathrm{m}^3$ with a total (fiducial) mass of 17 kt (10 kt). Section 6.3 summarizes the expected event rates and data volumes for the first two modules. Two additional detector modules, possibly with updated or novel technologies, will be added later. For now, we assume that data volumes and rates coming from other technologies, will not exceed the values for the FD1-HD or FD2-VD.

The detector will be sensitive to energy deposits on the order of $\sim 1\,\mathrm{MeV}$. The decay of ^{39}Ar,^{42}Ar, and other radioactive nuclei will produce a prohibitively high trigger rate if the threshold were to be set much below $\sim 5\,\mathrm{MeV}$. On the other hand, beam neutrino interactions will typically deposit well above 100 MeV. An energy-based trigger threshold will provide near perfect detection efficiency for beam neutrino interactions. More sophisticated triggering algorithms should also allow standalone detection of astrophysical sources, including higher-energy solar neutrinos and SNB candidates.

The data rates will be dominated by 4,500 cosmic rays expected per module/day. These events are vital for monitoring and aligning the detector. The next highest rate source of events will be calibration campaigns with radioactive and neutron sources and lasers. In all cases, the goal is to gather data from the full volume of the detector with as fine a granularity as possible.

Beam interactions themselves are expected to be quite rare, occurring in only 1/2000 of beam gates ($\simeq 2$/hr) Extraction of oscillation parameters will require both powerful background rejection, discussed in Section 1.1, and precise calibration of the energy scale of the experiment, hence the much larger

calibration samples.

Beam, cosmic ray and solar neutrino interactions are reasonably localized in time and space, involving a small fraction of a module over a few milliseconds. The DUNE DAQ group has plans to design and implement sparse readout of Far Detectors that will allow significant reduction in data size without loss of physics information if suitable triggers are used. Details of the expected rates and data volumes are described in Section 6.7.

1.3.1 Supernova candidates

Supernova candidates pose a unique problem for data acquisition and reconstruction. SNB physics in DUNE is discussed in some detail in the far detector TDR[6], reference [21] and a dedicated paper [22] and only summarized here.

DUNE is uniquely sensitive to ν_e through the reaction $\nu_e + \mathrm{Ar} \rightarrow e^- + \mathrm{K}^*$ which leaves an electron trajectory that can be used to estimate the pointing direction of the supernova explosion. Figure 1.13 shows the expected neutrino interaction rates from supernovae as a function of their distance.

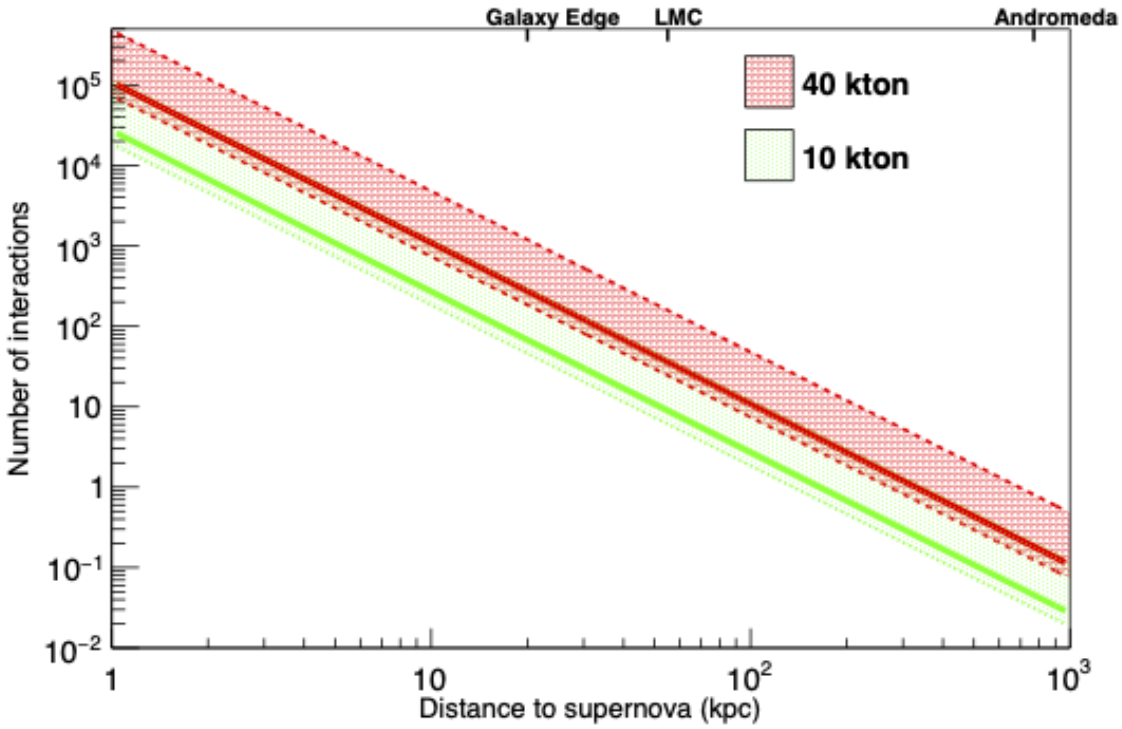


Figure 1.13: Supernova rates in the DUNE detector as a function of distance from the source, from reference [22].

A typical core-collapse supernova 10 kpc away is expected to yield around 3,000 charged current (CC) electron neutrino interactions across four detector modules. SNB candidates will be quite different from beam interactions, having small interactions with energies in the 5-30 MeV range spread across the full volume of the detector modules over many seconds, in contrast to the localized, coincident, 500-

10,000 MeV signature of beam neutrino interactions. These differences impose interesting requirements on the DAQ and computing models for the experiment.

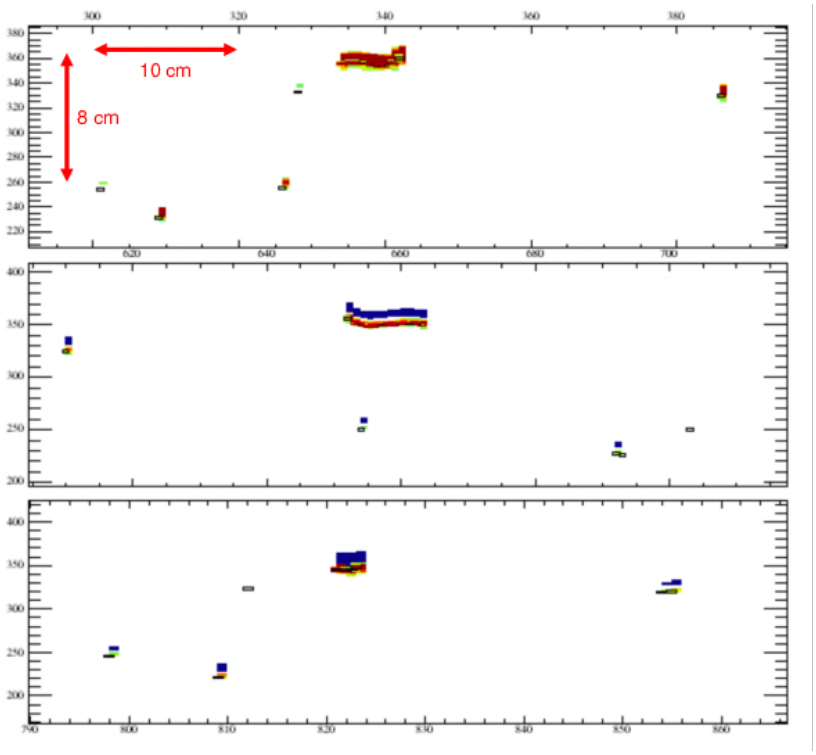


Figure 1.14: Simulated ν_e CC event with a 20.25 MeV neutrino, showing an electron track and "blips" from Compton-scattered gammas. The vertical dimension indicates time and the horizontal dimension indicates wire number. Color represents charge. The top panel shows the collection plane and the bottom panels show induction planes. The boxes represent reconstructed hits.

SNB physics and its influence on neutrino emission are not fully understood and will result in significant modulations of the event rates for different neutrino types over the few tens of seconds of the burst. DUNE's fine-grained tracking should allow significant pointing power with the most optimistic scenario of four modules and high electron neutrino fraction yielding pointing resolutions of less than 5 degrees. Other neutrino detectors worldwide will also be able to provide fast information but the DUNE information will be unique in its sensitivity to electron neutrinos. Figure 1.14 illustrates simulated signatures of SNB neutrino interactions in the far detector. The ability to produce a reasonably fast pointing signal is extremely valuable to optical astronomers doing followup, especially if the supernova were in a region where dust masks the primary optical signal. The need to be alert to supernovae and to quickly transfer and process the data imposes significant requirements on triggering, data transfer and reconstruction beyond those imposed by the beam-based oscillation physics.

The TPC and photon detector (PD) produce several signals of increasing precision. First, scintillation light is detected by the PD, yielding fast triggering information. Trigger primitives based on a subset of the LArTPC information can also be searched for supernova signatures. With the greatest resolution, a full detector readout can be mined for information across the full spatial and energy range of the detector over a period of up to 100 s. There is also research into full-chain reconstruction on highly-condensed data.

In reference [22] the performance and efficiency of fast triggering on scintillation light and TPC hits are investigated. Radiological and noise backgrounds are required to produce less than 1 false trigger/month. Both PD- and TPC-based triggers are sensitive to relatively low numbers of interactions ($\sim 10-25$) at acceptable background rates, yielding expected sensitivities out to the Large Magellanic cloud as shown in Figure 1.13.

Once the trigger system has identified a potential SNB signal the DAQ then records information for 100 s, yielding 140 TB of uncompressed information for a HD module and 180 TB for a VD module. These data must then be transferred offsite for processing with the full algorithms as the SURF site does not currently have infrastructure designed to support reconstruction of SBN candidate trigger records underground or on the surface.

Moving 320 TB (assuming the first 2 modules) of uncompressed data from SURF to processing centers will require, at minimum, 6-7 hours with the planned 100Gb/s network link. This can be accelerated by implementing onboard data compression and may require upgrades to the network when the projected third and fourth modules come online. The need to store up to 320 TB of data from a supernova candidate while also continuing normal data taking drives the size of local disk buffers at SURF and presumably requires similar reserved or rapidly-preemptable space at the centers where the data would be processed. Our ProtoDUNE experience indicates that reconstruction of LArTPC data takes between 1 and 3 sec/MB of raw data on present-day computing resources. It will thus take of order 12,000-40,000 cores to perform a preliminary reconstruction of the data as fast as it comes in. In principle, a significant fraction of the neutrino data could be processed in the 2-4 hours before a supernova becomes visible at optical frequencies.

1.3.2 Other phenomena – Solar Neutrinos and Beyond-the-Standard-Model Processes

There are multiple other measurements that can be made in massive low background detectors such as the FDs. These are described in a recent White Paper [23]. These include detection of solar neutrinos and searches for BSM processes such as neutron-anti-neutron oscillations. Detection of these rare processes will require low radiological backgrounds and trigger systems capable of reading out the detector at rates low enough to remain well below a 30 PB/year limit on data logging. They will require recording the full waveforms, perhaps over a limited physical region, to perform optimal signal extraction. In this document we concentrate on the oscillation, calibration and SNB scenarios as they are likely to dominate data rates in the near term.

1.4 Near Detector

High-precision oscillation physics requires a ND system to allow measurement of the unoscillated neutrino flux and to provide improved understanding of neutrino interaction physics. The DUNE collaboration is proposing a suite of near detectors optimized for these two goals. The proposed detectors are described in more detail in the Near Detector Conceptual Design Report [5].

The ND will be located in an enclosure in the path of the neutrino beam on the Fermilab site 574 meters from the target. Interaction rates per beam spill are expected to be very large (at 0.83 Hz), with 40-60 interactions per spill, including muons originating from interactions in material upstream of the fiducial

volumes. Figure 1.15 shows the beamline and location of the ND on the Fermilab site. There are three major subdetectors:

- A pixel readout LArTPC, ND-LAr, is the most upstream of the three subdetectors shown in Figure 1.16, where the beam propagates from right to left.
- Immediately downstream of ND-LAr is a detector for characterizing muons exiting the ND-LAr. This will be a magnetized steel range stack detector (TMS) for Phase I of the DUNE experiment. A gaseous liquid argon detector, ND-GAr, which serves as a muon spectrometer and allows more detailed study of neutrino interactions, is planned for Phase II operation when better control of neutrino interaction uncertainties is required.
- Beyond TMS/ND-GAr is the System for on-Axis Neutrino Detection (SAND) component of the ND that acts as a beam monitor.

Note that while the ND will have the same DAQ timing window flexibility as the FD, it is not foreseen that ND physics goals will require the use of varied time windows for trigger records, in contrast to the FD.

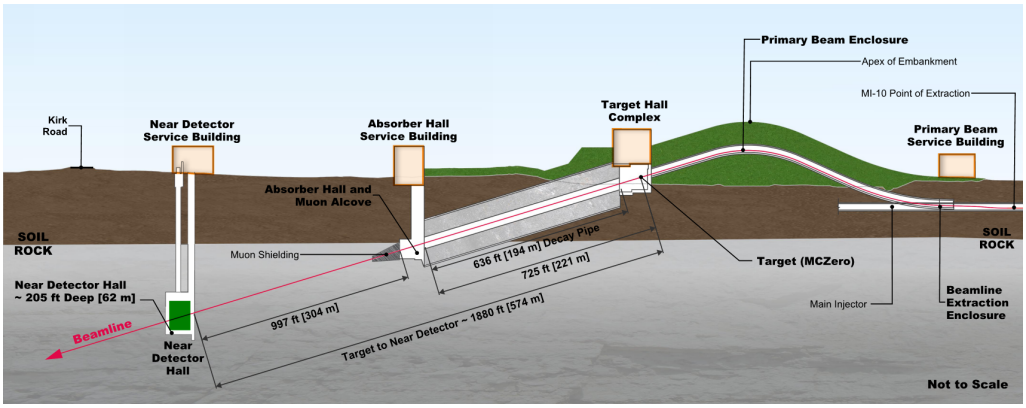


Figure 1.15: The neutrino beamline on the Fermilab site. The near detectors will be situated 574 m from the target and 62 m below grade.

1.4.1 Pixel LArTPC - ND-LAr

As the target material in the far detector is LAr, optimal cancellation of systematic uncertainties between the near and far detectors requires that the near detector include a LAr component to match the FD. However, at the intense neutrino flux and high event rate in the ND region, occupancies will be too high to allow the 2D readout provided by conventional wire planes. A new ArgonCube technology has been developed that allows pixelized charge readout and, along with modularity and highly capable light detection, provides unambiguous 3D imaging of particle interactions. The ND-LAr component of the ND is made up of a configuration of ArgonCube LArTPCs large enough to provide the required hadronic shower containment and statistics.

The pixel LAr detector is designed to have 12 million $3 \times 3\,\mathrm{mm}^2$ pixel channels and ~4200 PD channels. The LArTPC will read out only pulse times and integrals, in contrast to the far detector which reads out every time slice. The PDs will, however, read out complete wave-forms. A total of 3 MB of uncompressed data is anticipated per spill from the TPC with 5 MB from the PDs leading to an

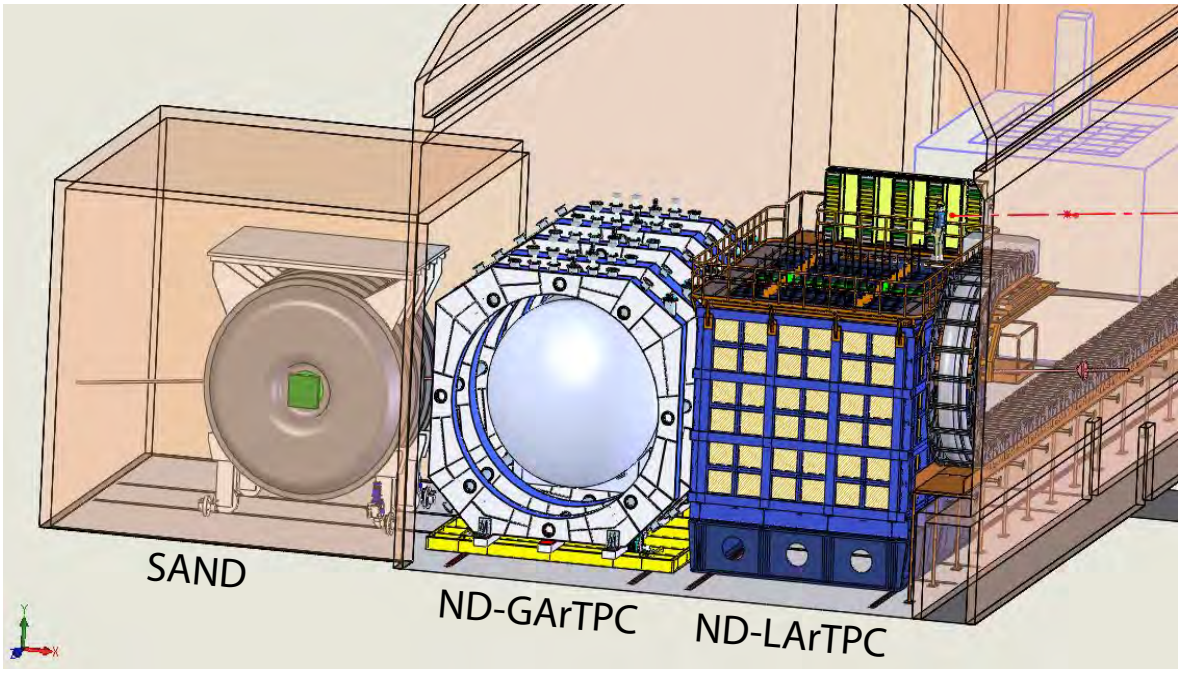


Figure 1.16: The ND systems in an on-axis configuration. The beam enters from the right in this view. The SAND scintillating beam monitor remains at beam center while the pixel ND-LAr and gaseous ND-GAr TPC detectors can be moved off-axis to make detailed studies of the neutrino flux at multiple angles.

estimate of 144 TB/year for uncompressed in-spill data. Calibrations and cosmic-ray data increase that data volume by around 20%.

1.4.2 Near Detector Phase I Muon Spectrometer (TMS)

The TMS, a low-cost, low-risk detector, will operate for the early part of the DUNE running when beam intensities are at their lowest and DUNE is not systematics limited. The TMS is a magnetized steel range stack (similar to MINOS [24]) downstream of ND-LAr. The purpose of the TMS is to measure the momentum of 1 - 5 GeV muons that exit ND-LAr by range with a momentum precision comparable to the Far Detector (taken to be 4%). The steel will be magnetized to identify the charge of muons it detects with better than 98% accuracy. These metrics are designed to ensure the DUNE physics program will function for the first 2-3 years of TMS operations with no degradation to physics output.

The TMS is placed 8.2 m downstream from the ND-LAr (center to center) and centered 2.51 m below the origin in y. The TMS consist of 200 total layers made up of alternating layers of magnetized steel plates and scintillator bars. The first 40 layers of steel are 1.5 cm thick. The rear 60 layers of steel are 4 cm thick. There is a 4 cm gap between each steel layer. The thin (thick) central steel is given a vertically oriented downward pointing field of 1.25 T (0.9 T), while the outer steel thin (thick) steel plates have their field pointing upwards with a strength of 1.5 T (1.0 T). The scintillator is implemented as a collection of vertical bars, 3.5 cm wide by 3 m tall by 1 cm thick, arranged into modules of 1.68 m wide. Four modules are placed side by side into a layer. These scintillator layers are replicated in the gap between steel layers.

The estimates for the TMS's data volume and raw processing needs are uncertain, but they are known to be less than the needs for ND-GAr. In the raw data, a TMS hit is 40-64 bits long. Assuming 64 bits per hit and 550 filled buckets per spill, each with 20 hits, one arrives at a raw data volume of 700 kb/spill, or approximately 88 kB/spill. A generous factor of four for out-of-spill data is assumed.

1.4.3 Near Detector Phase II GArTPC

The ND-GAr is a magnetized detector system consisting of a high-pressure gaseous argon time-projection chamber (GArTPC) surrounded by an electromagnetic calorimeter (ECAL) and a muon system. The ND-GAr measures the momentum and sign of charged particles exiting the ND-LAr. In addition, for neutrino interactions occurring in the ND-GAr itself, higher resolution and lower momentum thresholds can be achieved for charged particle tracks, leading to improved neutrino interaction models. This capability enables further constraints of systematic uncertainties for long-baseline neutrino oscillation analyses.

The ND-GAr is composed of 678,136 readout pads in the TPC, and approximately 3 million channels in the ECAL. Approximately one in five spills will generate an interaction in the GArTPC, but particles entering the gas from interactions in the ECAL will provide the bulk of the data volume. The readout strategy will be similar to the LArTPC, with only time and integral recorded. A data volume of 2 MB of uncompressed data per spill is expected from the TPC. The calorimeter is expected to contribute approximately 1 MB per spill of uncompressed data.

1.4.4 SAND

SAND's primary function is the primary beam flux monitor. The SAND detector comprises an active LAr target exploiting light read-out (GRAIN) placed upstream of a low-density tracker based on straw tube technology (STT) The layers of straw tubes are sandwiched between radiators of configurable composition. The vast majority of the radiator material is planned on being $(\mathrm{CH_2})^n$, though carbon radiators are also included in the design so that hydrogen measurements can be obtained by subtraction [5]. Both are surrounded by 4π-hermeticity ECAL [25] and immersed in a 0.6 T magnetic field provided by a solenoidal magnet. The ECAL is read-out by 4880 photosensors, with an estimated 5500 total hits per spill or 33 kB of packed data per spill. The STT in the current design foresees 218,000 channels and a mean number of hits per spill of about 12,500 corresponding to 75.6 kB of packed data per spill. In the current design, GRAIN will be instrumented with up to 76 (32x32) silicon photomultiplier (SiPM) matrices. The mean number of hits per spill is about 120,000 corresponding to 1.44 MB of packed data per spill. With these assumptions, the data volume from SAND comes to about 40 TB/year. The amount of data from out-of-spill cosmic rays is estimated to be 20% of that of the in-spill data, or approximately 8 TB.

1.5 Relation of Physics Goals to Offline Computing Challenges

1.5.1 Physics and sociological drivers

The DUNE physics program drives several detector characteristics that pose novel computing challenges. While the overall data volumes are smaller than those routinely handled by the large LHC experiments,

the remote detector site and unique physics goals present novel computing challenges.

Fine segmentation needed for electron-photon discrimination:
The primary goal of the DUNE long-baseline experiment is measurement of $\nu_\mu \to \nu_e$ and $\overline{\nu}_\mu \to \overline{\nu}_e$ oscillation probabilities for GeV-scale accelerator neutrinos. These oscillation probabilities are intrinsically low and are sensitive to backgrounds in which the neutral current process $\nu_\mu + A \to \nu_\mu + \gamma/\pi^0 + X$ produces a photon or π^0 meson that in turn produces electromagnetic showers that fake an oscillation signal. Fine detector segmentation is necessary to distinguish between these scenarios. Figure 1.2 illustrates this capability. The need for sub-cm-level segmentation drives the technology choice of LArTPCs and hence the number of channels.

Low-energy thresholds for astrophysical neutrinos:
Other important physics goals are the detection of astrophysical neutrinos from the sun, possible SNB neutrinos, atmospheric neutrinos and BSM signatures in the FD. Astrophysical neutrinos produce lower-energy signatures, in the 1-30 MeV range. Extracting such signals, near the noise threshold of the detector and in the presence of radiological backgrounds, requires careful attention to signal processing and zero-suppression for the FD TPC and PD waveforms. The need to optimize the low-energy threshold drives our need to carefully record waveforms with minimal processing and thus drastically increases the raw data volume.

Precise energy calibrations:
An additional challenge in oscillation physics is the need for accurate energy calibration in order to fully exploit the energy spectrum of the reconstructed neutrinos to further constrain oscillation parameters. While LAr detectors have a reputation for stability, the large volumes, complex E field configurations, liquid motion, and potential variations in electron lifetime and drift velocity make it necessary to have large calibration data samples that span the full FD detector volume. Large cosmic ray and artificial calibration samples will dominate the total data volumes from the FD.

Supernovae:
A supernova neutrino burst (SNB) candidate will generate 320 TB of (uncompressed) data across the first two modules, resulting in thousands of data files produced over a 100 s period. These data must be recorded at a low energy threshold due to the expected interaction energy range, but must also be analyzed quickly and coherently in order to measure the time evolution of neutrino emissions, which carries invaluable information about the supernova process itself. In addition, if DUNE can quickly analyze electron-scattering interactions and separate a fraction of them from CC interactions, we can provide pointing information to telescopes for optical followup [4]. Supernova physics drives the need for fast data transmission from the FD to computing facilities and for robust tracking of data movement so that a full picture of the SNB interaction can be reassembled after signal processing. The drastically different time scale of SNB physics also places requirements on the software framework.

ND integration:
While the FD modules produce large data volumes, the detectors themselves are reasonably simple, consisting of a small number of technologies and large numbers of repeating components. The ND is much more complex. The ND use case is similar to other fixed target experiments such as SBND at Fermilab and Common Muon and Proton Apparatus for Structure and Spec-

troscopy (COMPASS) at CERN. The main computing challenge for the ND will be integration of a large number of disparate detector technologies into a coherent whole. Here careful attention to simulation, detector geometry and configuration, and code management will be the major challenges.

Analysis and parameter extraction:
DUNE has over one thousand collaborators spread across five continents. Those collaborators will want to analyze our data over several decades. Fortunately, once reconstruction has been done, neutrino interaction samples are generally simpler than event records at colliders and should allow researchers to analyze them at their local institutions. However, final parameter extraction using large numbers of nuisance parameters remains a computationally intense problem and will require significant resources and efficient utilization of high-performance computing (HPC) to quickly achieve final results.

1.5.2 Resulting offline computing challenges

DUNE offline computing faces four major challenges, some of which are unique to DUNE and others shared widely by HEP experiments.

Large memory footprints - DUNE events, with multiple data objects consisting of thousands of channels with thousands of time samples, present formidable challenges for reconstruction on typical HEP processing systems. Efficient processing of DUNE data will require careful attention to data formats and, likely, substantial redesign of the processing framework to allow sequential processing of chunks of data. Chapters 3 and 4 describe the status of applications and frameworks.

Storing and processing data on heterogeneous international resources - DUNE depends on the combined resources of the collaboration for large-scale storage and processing of data. Tools for using shared resources ranging from small-scale clusters to dedicated HPC systems need to be developed and maintained. Fortunately, HEP, through the WLCG, OSG and HEP Software Foundation (HSF) has a well developed ecosystem of tools that allow reasonably transparent use of collaboration computing resources. Chapters 6, 7 and 11 describe the data volumes, computing model and data management plans.

Machine learning - Use of machine learning techniques can greatly improve simulation, reconstruction and analysis of data. However, integration of Machine Learning (ML) techniques into a software ecosystem of the size and complexity of a large HEP experiment requires substantial effort beyond the original demonstration. How is the ML trained? What special data format or processing requirements are present? How is the algorithm versioned and preserved to ensure reproducibility? Chapters 3 and 17 discuss the applications and management.

Efficient and sustainable use of resources - As discussed in Chapter 6 DUNE will use substantial computing resources. Historically the main concern has been financial - getting the most computing possible within a given budget. However, our activities also have environmental impact, through energy consumption and the creation, use, and disposal of hardware. To make our efforts more sustainable, a driving consideration in the design of our systems is efficient use of storage and CPU resources. This includes not just optimization of our major workflows (Chapters 7-13) but also documentation and training (Chapter 18) of end-users in efficient and error-resistant

practices to avoid needless reprocessing.

Keeping it all going - There is a large suite of activities, that, while not necessarily novel, still needs to be done over the full lifetime of the experiment. These activities include database design and operations, security updates, code management, documentation, training, and user support. For example, the ND presents few novel computing challenges in memory or CPU use but is highly complex in terms of the number of detector systems that must be integrated. Another example is the continuing evolution of operating systems and security requirements. These require constant modifications to working systems to maintain operations. A third activity is database design and maintenance. Here the problem is largely sociological, getting the attention of busy people to do database design and then populate and use the official collaboration databases. This requires continual engagement with reluctant stakeholders. These issues are discussed throughout this document with special reference to databases, Chapter 5, authentication Chapter 16, code management Chapter 17 and training and documentation Chapter 18

A broad suite of use cases is discussed in Chapter 3.

Chapter 2

Computing Organization

This document chiefly describes the activities of the formal Computing Consortium that concentrates on development and operations for the Deep Underground Neutrino Experiment (DUNE). The consortium provides the hardware and software computing infrastructure that allows development and implementation of algorithms and data analysis techniques. The algorithms and techniques themselves are within the purview of the collaboration physics groups. The whole of DUNE offline computing consists of three major focus areas, illustrated and described in Figure 2.1. The Consortium itself provides the underlying infrastructure and collaborates with the sites in delivering computing resources. Chapter 7 includes a description of the global sites and their hardware contributions. The Computing Consortium and the global sites meet weekly via teleconference to coordinate operations.

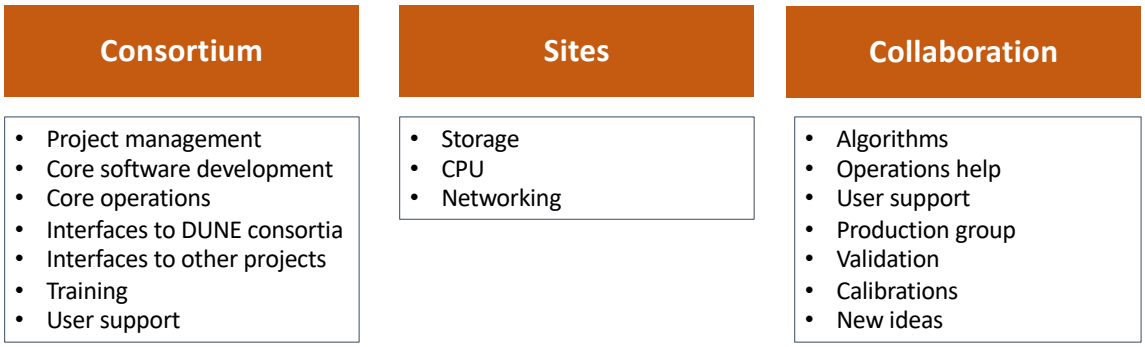


Figure 2.1: Offline computing roles. The first column shows the formal DUNE Computing Consortium, mainly comprised of experts funded to perform offline computing for DUNE. The second column represents the infrastructure contributed to computing by collaborating nations and institutions and formalized through the Computing Contributions Board (CCB). The third represents the broader contributions of the collaboration that are vital to the overall effort but not managed by the Computing Consortium.

Figure 2.3 shows the Computing Consortium organization as of September 2022. The Consortium leadership consists of a Consortium Lead, responsible for overall coordination, two technical leads, one US-based and the other from the international collaboration, and a Computing Architect. There

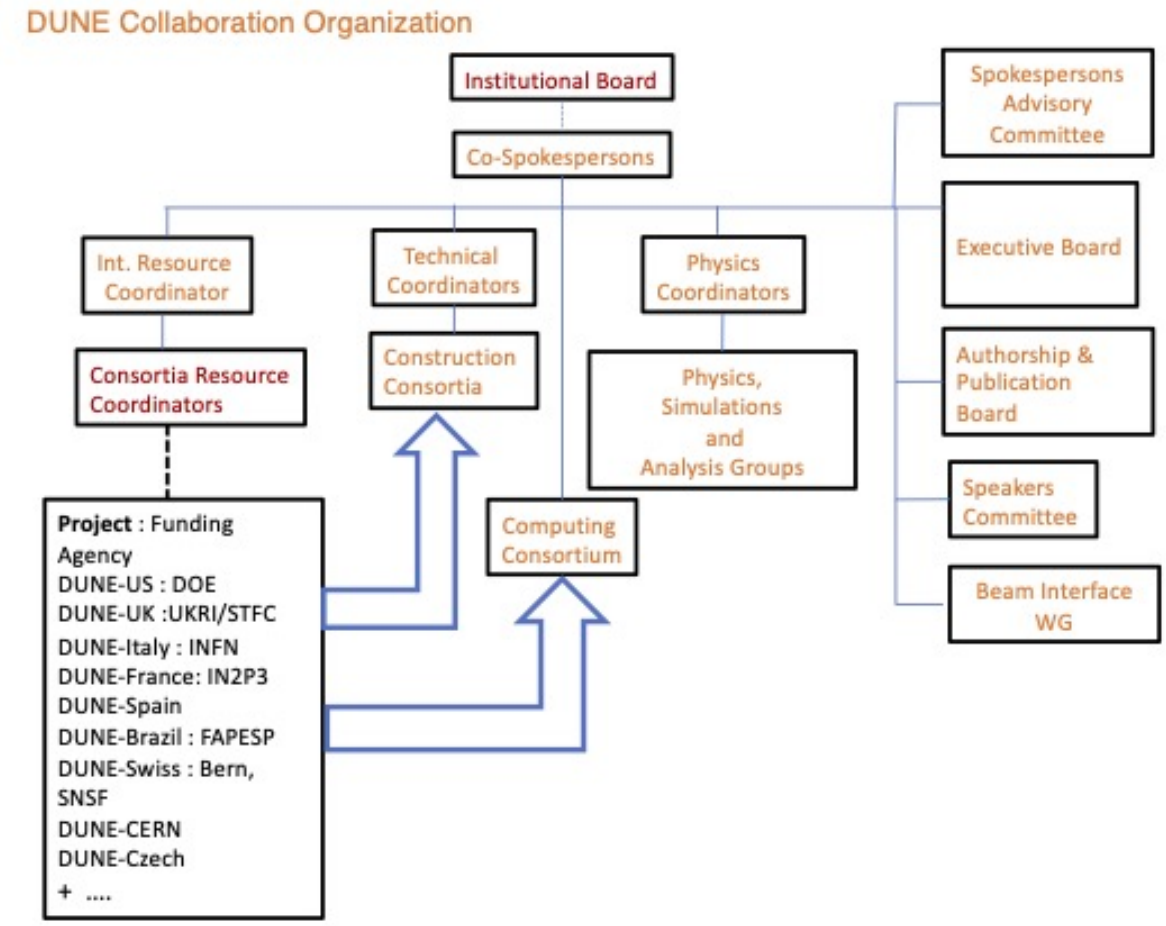


Figure 2.2: DUNE Collaboration organization chart, July 2022.

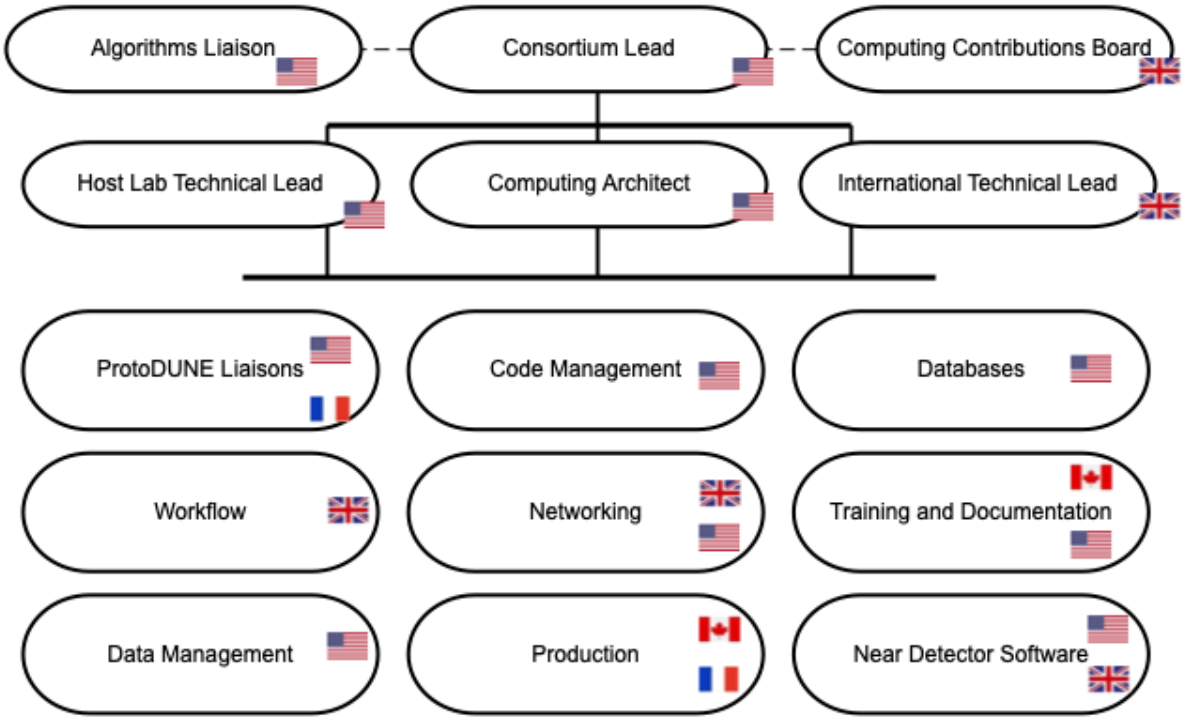


Figure 2.3: DUNE Computing Consortium organization chart, September 2022. Flags for the group leaders are shown.

is an Algorithms Liaison responsible for coordinating with the Physics groups, liaisons to the ProtoDUNE experiments and to the calibration consortium. There is also an independent CCB, described in Section 2.3, that negotiates resource contributions from international collaborators.

The Computing Consortium is not formally part of the DUNE construction project but has representation on the Executive Board. The consortium is part of the interface document matrix with the construction consortia and the interface agreements are listed in Table 2.1. The most important formal interfaces are (1) data-management and networking with the data acquisition (DAQ) and calibration groups and (2) databases with the DAQ and hardware construction groups. The Computing Consortium also has liaisons to the physics groups and the ProtoDUNE experiments.

Table 2.1: engineering document management system (EDMS) interface documents between the Joint Offline Computing (JT COM) with various DUNE hardware and physics consortia.

Document title	EDMS ID
JT COM and SP APA Consortium Interface Document	2145145 v.4
JT COM and SP PD Consortium Interface Document	2145146 v.2
JT COM and SP TPC Consortium Interface Document	2145147 v.2
JT COM and DP CRP Consortium Interface Document	2145148 v.1
JT COM and DP PDS Consortium Interface Document	2145149 v.1
JT COM and JT HV Consortium Interface Document	2145150 v.2
JT COM and JT DAQ Consortium Interface Document	2145151 v.2
JT CAL/CI and JT COM Consortium Interface Document	2145159 v.2
JT COM to Facility Interface	2145167 v.1

2.1 Internal Organization

Within the Consortium there are specific development groups responsible for particular components of the computing infrastructure. Many of these activities are described in greater detail in separate chapters.

- ProtoDUNE Liaisons - make certain that offline computing is coordinated with ProtoDUNE data taking and analysis.
- Data Management - responsible for storage, cataloging and delivery of data. See Chapters 7, 11
- Workflow - responsible for global coordination of computing resources. See Chapter 7, 13.
- Code Management - responsible for maintaining the code infrastructure and repositories and building common executables. See Chapter 17.
- Networking - Responsible for negotiating suitable networking capabilities from Sanford Underground Research Facility (SURF) to the host laboratories and from the host laboratories to the global compute resources. See Chapter 12.
- Production - Responsible for setting up and running common reconstruction and simulations jobs. This group includes both experts and collaboration volunteers who run the jobs. See Chapter 13.2 for a description of the current status.
- Databases - Responsible for the design and implementation of databases to support offline data processing. See Chapter 5.

- Training, Document and User support - Responsible for development of training materials, infrastructure for and review of documentation and for directing users to experts and documentation where needed. See Chapter 18.
- Near Detector Software - responsible for coordination of the unique near detector (ND) software and integration with the main DUNE infrastructure. See Section 3.5.

2.2 Funding Sources for Computing Development

Most personnel working on DUNE computing are supported by their institutions or national funding agencies as part of their base or DUNE project support. This support is generally not long-term and is subject to periodic requests to the relevant funding agencies. For example, a US DOE-funded consortium of four universities and three national labs is funded specifically for DUNE computing infrastructure at a level of \$1M/year through 2024, directly supporting postdocs and lab physicists, summing to ~ 6 FTE. The UK, France and European Laboratory for Particle Physics (CERN) also make significant contributions to software development through the DUNE project while many nations contribute substantial hardware capabilities. Membership in the Computing Consortium does not have any specific implication for provision of CPU and storage resources; members may contribute personnel or hardware, or both; these contributions are handled separately. Hardware commitments are made through the CCB described in the next section.

2.3 Computing Contributions Board

The CCB has been set up to formalize and recognize the contribution of DUNE partners to the computing and storage capacity required for large-scale DUNE computing. For the present, the CCB takes a "nation" as the natural unit of aggregation to set a "size metric" upon which to base requests for resources. The CCB does, however, recognize that whilst within some countries centralized coordination is natural, this is not true for others. A more natural, fair-share unit of aggregation would be based on each funding agency (as used by the Worldwide LHC Computing Grid (WLCG)), and DUNE recognizes that an evolution to this model will be required in the future. Nevertheless, at present, during the integration and construction phase of DUNE, the current model is providing sufficient resources and informal pledges with the assumption that institutions within each nation can (loosely) coordinate where needed.

The fair-share metric to which the CCB will move, once DUNE is taking data, is likely to be senior authors. During construction, the current listing of DUNE personnel in the collaboration database is a very poor proxy for the number of active contributors. Therefore, during construction all nations with a threshold number of DUNE participants, and that are capable of providing Tier-1 or large Tier-2 capacity to LHC experiments, are asked to provide a "reasonable" share (see below). We refer to these as compute-active-nations. This is very flexible and is up to each nation to decide if it wishes to be classified as a compute-active-nation.

At present the CCB is composed of a Chair, one member per compute-active-nation, one member for each of Fermi National Accelerator Laboratory (Fermilab), CERN and Brookhaven National Laboratory (BNL), and the Computing and Software Consortium Management ex officio.

The Computing Consortium management produces an overall requirements document each year[26] between November and January. The CCB receives this document, and then seeks pledges to meet these requirements. As host lab, Fermilab plans to provide $\sim$25% of the disk and CPU capacity, as well as the primary tape service. CERN also currently provides substantial capacity, including a second tape copy of raw data for ProtoDUNE. The aim is for the remaining capacity to come from other contributors according to the prevailing computing model. A substantial proportion of capacity is expected from outside of the US ($\sim$50%). National contributions of at least 5-20% are requested, depending upon the circumstance and capability of each compute-active-nation.

It is intended to use the Computing Resource Information System (CRIC) (http://wlcg-cric.cern.ch/) information system to record pledges, this work is currently in progress. In due course this process could be formalized into a (non-binding) memorandum of understanding (MoU).

The CCB may also receive requests and information from the Computing Consortium Management with respect to other non-capacity matters. The CCB may seek to help with such requests where they pertain to national contributions. This may include promotion of requests for software engineering support to be propagated within each nation.

Chapter 7.1 describes the impact of collaboration contributions in more detail.

2.4 Collaborations with other Organizations

DUNE Computing is actively engaged with multiple other computing organizations around the globe with the intent of both drawing from and contributing to the community knowledge of computing solutions. As a member of the Open Science Grid (OSG) Council, DUNE is helping define the direction of High Throughput Computing within North America. DUNE Computing is also an observing member of the WLCG and is using multiple services and solutions developed within the scope of the WLCG. We also work closely with the HEP Software Foundation (HSF) and Rucio collaborations in developing modern software solutions. DUNE intends to continue to maintain these partnerships as part of both sharing new computing developments and efficiently integrating new services into our computing model.

Part II

Single Interaction Scale

Chapter 3

Data Processing Considerations and Challenges

Before describing the large scale data model for the Deep Underground Neutrino Experiment (DUNE) we begin by describing the activities and algorithms that drive that model at the trigger record scale.

3.1 Introduction

In this chapter we describe some of the use cases for trigger record level processing, building on the experience gained from ProtoDUNE and using lessons learned from the 2018 run to understand the performance of future near and far detector modules.

In addition to the challenges faced by HL-LHC experiments, as described in the HSF White Paper[27], DUNE (along with some dark matter and astrophysics experiments) faces considerable challenges in extracting very weak signals from large, noise-dominated data volumes. This additional challenge places strains on memory management beyond those anticipated at collider experiments.

3.2 Data Acquisition and Storage

Figures 3.1, 3.2 and 3.3 outline the offline processing flow for raw data, protodune and FD simulation, and near detector (ND) simulation. Offline processing starts with the transfer of data from a disk buffer located at the experiment.

Data acquisition (DAQ) software and hardware are the responsibility of the DAQ consortium, with significant interfaces to offline computing. Important overlapping factors include the management and versioning of data format specifications, communication of calibration and configuration information between online and offline computing, high-speed networking for data, high-reliability networking for configuration and monitoring, and adequate disk and CPU resources on both ends of the connection to allow both efficient data transfer and continued data taking in the event of an extended network interruption. These efforts are coordinated by dedicated liaisons, for example between the offline databases

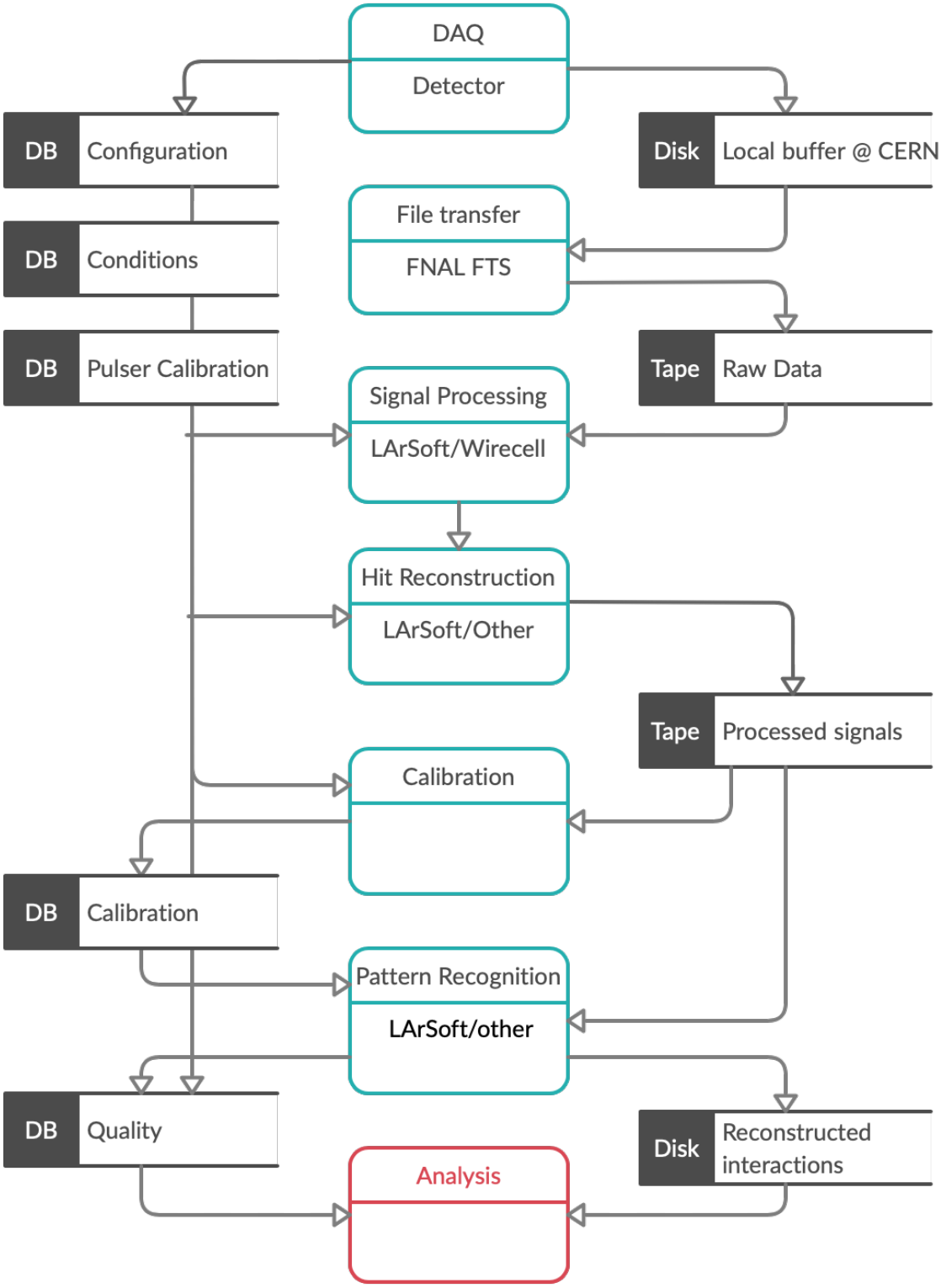


Figure 3.1: Processing diagram for standard ProtoDUNE and far detector (FD) data reconstruction. All processing and data transfers shown are part of offline processing. The central boxes show the processing steps, the right side shows the large-scale data flow, and the left side shows the auxiliary information needed for processing.

and online slow controls and formal liaisons to the ProtoDUNE-HD and ProtoDUNE-VD experimental operations groups. Informal coordination also occurs through shared Slack channels and joint sessions at collaboration meetings.

These overlaps already apply to ProtoDUNE, where data taking occurs at the European Laboratory for Particle Physics (CERN) with offline databases and archival storage in the US, and, in the future, will apply to the far and near detectors in South Dakota and Illinois, respectively. Current status and proposed solutions are discussed in Chapters 5 (Databases) and 12 (Networking).

3.2.1 Data Transfer from the Experiment

Data needs to be buffered locally and then transferred to permanent storage at the host lab(s). Data transfer activities include the generation of descriptive metadata, file transfer, writing the files to permanent storage, and integrity checks. Once data are confirmed to have successfully reached permanent storage, the local buffer may be cleared. Current specifications call for the local DAQ buffers at the FD remote site to be able to store 3-7 days worth of all triggered DAQ data or several supernova neutrino burst (SNB)'s worth (~ 640 TB) at a minimum.

The remote location of the far detector, and rack-space and power limitations at both the 4850L detector level and the surface mean that many computing tasks (for example data reformatting) may need to be performed at the host labs rather than at the experimental site. The main requirement is that there be sufficient local data buffer space (of order 1 PB, or greater than one week at normal data rates) to withstand a significant network outage or staging of supernova data.

Other challenges include the need for adequate lossless data compression to prevent data volume explosion. Overall our strategy is to move operations which can be done offsite, offsite, reserving infrastructure at the Sanford Underground Research Facility (SURF) site for activities (such as DAQ and buffering) that cannot be done elsewhere without the risk loss of data. Chapter 12 describes the status and plans for networking.

3.2.2 Control, Configuration, Conditions and Calibrations

In addition to large-scale data transfers, online data-taking configurations and conditions need to be stored and communicated to the offline systems. High-reliability network connections are needed to ensure that control signals, and configuration and monitoring information are exchanged between the remote sites, local control facilities and the host laboratory. This is the responsibility of the Fermi National Accelerator Laboratory (Fermilab) networking groups and the DAQ consortium with input and assistance from offline computing.

3.2.2.1 Slow Controls

Data from slow control and monitoring systems need to be recorded for offline use. Examples include set points and readback for high voltage (HV) systems, pressure, temperature and the outputs of purity monitors. A potential challenge: commercial SCADA systems tie DUNE to proprietary software that may have limited interfaces, require expensive licenses and need migration to new systems if the vendor stops support. Another issue of SCADA systems is the fact that they need to last for many years and are generally supported only for the version of the operating system for which they were built. It is a

challenge therefore to keep those systems running for extended periods of time, despite the progress in virtualization and containers.

3.2.2.2 Online Calibrations

Online calibration information (e.g., pedestal information for some sub-detectors and trigger time offsets relative to an absolute time standard) needs to be passed to the offline systems.

3.2.2.3 Beam Monitors

Beam monitoring information, either per spill for neutrino interactions or per beam trigger in the CERN test beams, needs to be recorded and made available for offline processing. The IFbeam system used for NuMI is already in use for ProtoDUNE.

3.2.3 Monitoring Information

Local online monitoring of data and experimental conditions will be done by the DAQ. The offline computing consortium will provide fast feedback on data movement and data quality based on the fast processing of a subset of the data transferred offsite.

3.2.4 Offsite High-level Data Filtering

This is not yet part of the data plan but there may be a need for offsite filtering of data before it is written to permanent storage. Initiation of a high-level trigger offsite remains a possibility if the need arises. Local space, power and cooling issues make this difficult to do at the SURF.

3.2.5 Data Compression

The data volumes and rates anticipated are significant. Storage and network resources can be optimized with zero-suppression (lossless or lossy) and data compression. Multiple avenues for data reduction are being explored, including on-board algorithms in the readout boards or downstream DAQ systems, or before data are written to archival storage. Considerations include the computational load of compression/decompression, especially of the very large data objects the far detector can produce, and irrevocable data loss in the cases of lossy methods.

In ProtoDUNE-SP lossless compression was performed as part of the DAQ. For ProtoDUNE-HD this is not planned due to the need to keep processing overheads low in the DAQ. Lossless compression of FD time projection chamber (TPC) data will need to be applied later in the processing chain, most likely using the intrinsic compression methods of the chosen data format(s). ROOT allows implementation of lossless compression, as does Hierarchical Data Format (HDF5). The data model assumes that for ProtoDUNE-HD and ProtoDUNE-VD the same level of compression will be achievable as in ProtoDUNE-SP. The NDs currently plan to take advantage of their higher signal/noise ratio and do lossy zero-suppression.

3.2.6 Outputs from DAQ and Monitoring

The end result of data acquisition includes the transfer of the data themselves and of the metadata and configuration/calibration/beam information needed for offline processing. Only when the data and

metadata have been successfully transferred and stored can they be deleted from the local DAQ storage. The computing consortium is responsible for developing the software for transferring data from the DAQ disk buffer and deleting stale data, while the DAQ consortium is responsible for providing the hardware resources (CPU, memory and disk) to run the data transfer software, as well as provide alerts to the running data transfer program when new data are available [28].

3.3 Simulation Chain

The DUNE simulation chain involves a large number of steps including beam simulation, particle interaction or decay simulation, simulation of energy deposition and simulation of detector and electronics response and noise. FD and ProtoDUNE share a common simulation framework, based on art and Liquid Argon Software (LArSoft), with other Fermilab liquid argon time-projection chamber (LArTPC) experiments while the ND simulation framework is still under development. Figures 3.2 and 3.3 illustrate the main processing flow for the photon detector (PD)PD and ND.

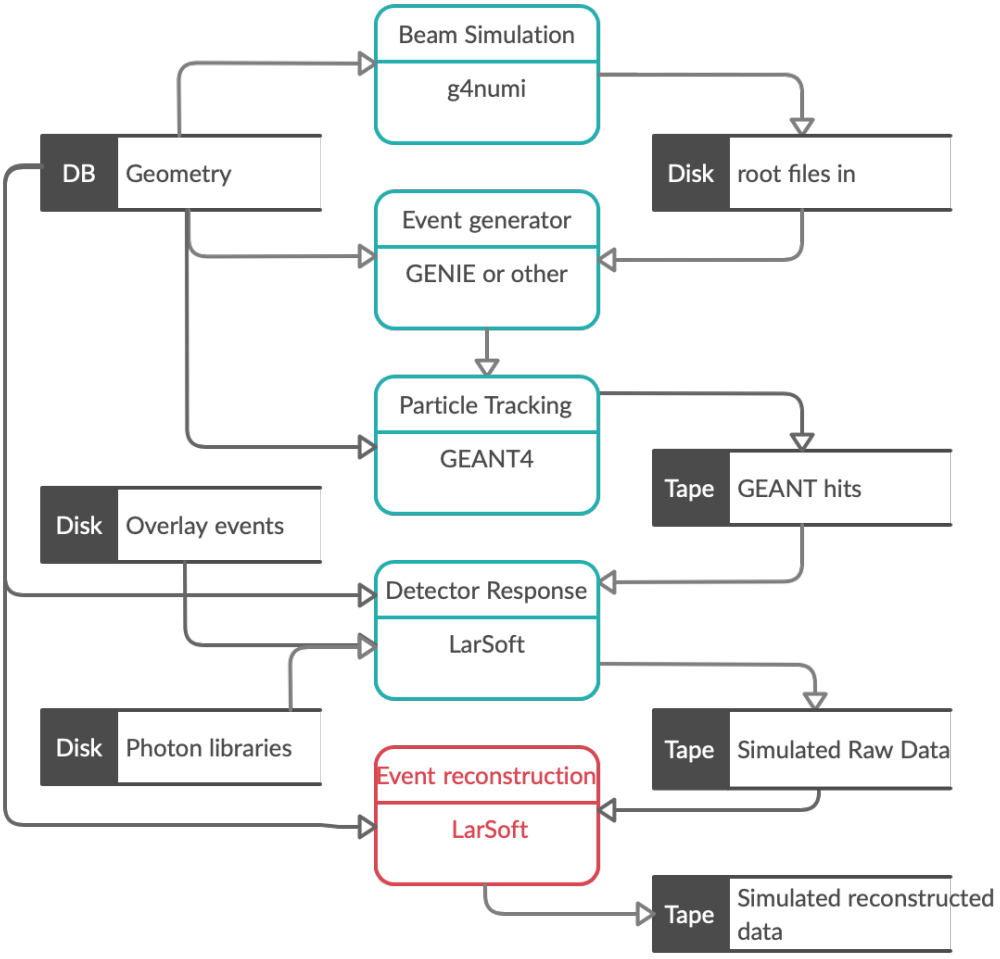


Figure 3.2: This diagram shows the components of the LarSoft based Far Detector and ProtoDUNE simulation workflow. Major external inputs are shown on the left while the main data movement is shown on the right. The reconstruction chain is illustrated in 3.1.

In the following sections we break out in more detail the portions of this simulation chain, and enumerate

the computational challenges and technical approaches that are associated with them.

3.3.1 Supporting Simulation Volumes

The simulation needs and scale of Monte Carlo generation that are required for DUNE are driven by uncertainty budgets of different analysis results that we are pursuing. In the case of the long baseline oscillation analyses, the small signal rates and sources of potential backgrounds from cosmic ray induced events and from radiogenic backgrounds set the scale of the simulation that is needed to cover different systematic uncertainties. In this respect, the amount of supporting Monte Carlo that is required for far detector portion of the long baseline oscillation measurements is estimated at approximately a factor of 10x of the experimental exposure, as measured in kT*kW*s, for the signal channel [beam] samples, and a similar 10x or greater simulation of the background channels (cosmic and radiogenic) as determined by the beam live time of detectors measured in seconds. These estimates are based upon similar approaches taken by earlier neutrino experiments (NOvA, MINOS) which have been statistically signal limited by their far detectors and for which cosmic ray backgrounds can contribute significantly to the backgrounds. However, it should be noted that unlike NOvA which was surface detector with only 3 m earth equivalent overburden, DUNE has the advantage of a highly suppressed cosmic ray flux owing to its deep underground siting. This may allow for background Monte Carlo samples to be smaller than our initial estimates but also may require more advanced computational techniques to generate the attenuated spectra over the large flux surfaces of the detector modules.

In contrast to the far detector simulations, the near detector complex also will require significant supporting simulation to drive the beam flux predictions for the combined near to far oscillation calculations and for the large suite of near detector based cross section physics measurements. Similar to the far detector simulation volumes we estimate that, based on similar oscillation and cross section programs from past neutrino experiments, we will require supporting simulation at a level of 10-20x the integrated beam flux, as measured in kW*seconds for the near detector cavern complex. In particular this supporting simulation need is driven by the beam flux uncertainties in targets and particle production models, and by neutrino induced interactions in the rock surrounding the detector cavern that result in muon fluxes through the detector volumes which overlap with signal interactions in the main detector volumes.

We also recognize that these estimates are essential minimal estimates, and that in almost all cases the DUNE program benefits from additional Monte Carlo in terms of providing lower statistical uncertainties on the Monte Carlo and from the single event sensitivities that can be reached for rare event interactions and interaction topologies. In practice, the supporting Monte Carlo sample sizes will be driven by the DUNE physics groups and their specific analyses. This is especially true if the data are categorized into many sparsely-populated categories, or if they are binned in multi-dimensional histograms as is common in differential (and doubly-differential) cross section measurements. In these cases, even if the number of data events is very small (or even zero) in portions of the spectra, the predictions of rates in each category or bin must still be reliable in order for parameters to be extracted properly and sensitivities calculated. In these cases, we expect the supporting Monte Carlo factor may need to be higher, in the range of 10x to 100x of that of the beam interaction sample in order to cover the low rate portions of the parameter spaces.

Similarly nucleon-decay searches and other analyses seeking rare events or limits on rare processes will require larger samples of simulated cosmic-ray events or radiogenic backgrounds than are required by

the core oscillation measurements. These samples may also differ from the core oscillation samples in the method and regions that they are generated over, both inside and near the FD, in order to simulate the very small fractions of events that may leak into the fiducial detector volumes and be potentially selected. Strategies to reduce the CPU and storage requirements for generating these rare search samples, such as rejecting events at the generator level if they are known not to pass exotic-search requirements no matter what happens in the detector, may be useful in optimizing the production of these samples and reducing the computing requirements and budget they represent.

As mentioned in Section 3.3.4, some, but not all, generator systematic uncertainties can be evaluated using reweighting techniques without having to generate, simulate, and reconstruct additional samples. Between five and ten versions of the FD simulation samples however will be needed to address generator systematics. As these uncertainties are expected to factorize from detector systematics, shortcuts, such as zero-suppression and simulation in smaller geometries can be taken in order to reduce the CPU, memory and storage impact of the need to study generator systematics.

The simulated data size for the FD is expected to consist of a different mixture of physics data types than the raw data. The raw data volume is expected to be dominated by cosmic rays, supernova triggers, and calibrations, with a much smaller amount of data from atmospheric neutrinos and beam neutrino interactions. These latter categories will require large multiples of the data in simulations, while the cosmic-ray sample and the supernova triggers will require a smaller simulation to raw data volume. Various strategies can be taken especially in the supernova simulations, such as placing the stubs much more densely in the detector than in the expected raw data, to reduce the number of empty waveforms simulated.

3.3.2 Beam Simulations

The MARS[29] simulation is used in the design of the beamline and near detector systems and for safety and environmental calculations. It is a significant user of CPU time. The MARS codes are not available outside of the US due to export controls which poses some challenges in balancing compute tasks.

Neutrino production in the Long-Baseline Neutrino Facility (LBNF) beamline is simulated using the g4lbnf[30] framework. The ProtoDUNE beamlines were designed and simulated using the MAD-X framework used at CERN[31]. Inputs include random number seeds and the beamline and target region geometry and materials. Outputs are flux files containing simulated particle trajectories. The flux files contain sufficient information to allow reweighting based on interaction cross sections. These flux files must be cataloged and made available to offline simulation jobs. The size and versioning of these files lends the distribution of flux files to remote sites to be performed using StashCache [32]. See Section 13.2.2 for more details on I/O handling for offline jobs.

3.3.3 Detector Geometry Description

Offline event generation, simulation, reconstruction and visualization tasks require descriptions of the geometries of the relevant detectors. Similar to some dark-matter and neutrino-less double-beta decay experiments, a neutrino experiment's detector is the interaction target. Therefore, the event generator must be aware of the spatial distribution of significant amounts of material, listed separately for each element's contribution, and the local densities. Neutrino detectors contain some elements for which the neutrino-scattering cross sections are poorly known; when these do not contribute significantly to the

detector volume, a simplified geometry containing only the key components is often useful.

Once the primary interaction has been generated, Geant4 [33] requires a representation of the spatial locations of materials in order to simulate the subsequent propagation and energy depositions of all particles except neutrinos. As was the case for generating primary interactions, a simplified geometry description is needed since a detailed geometry with a cylinder of CuBe for each wire in a horizontal-drift anode plane assembly (APA) would use a large amount of memory and CPU for a very small amount of non-fiducial interaction modeling.

The common language used in DUNE for geometry description is geometry description markup language (GDML) [34]. It is stored in human-readable text files. Versions of each detector's geometry are saved alongside software source code in the code repository. When a release is built, these GDML files are copied into directories that are visible to running jobs. At the time of writing, CERN VM File System (CVMFS) is the chosen distribution method for pre-compiled code and small auxiliary files like GDML files. Both Geant4 and ROOT have functionality for reading GDML files.

One notable requirement for code versioning is that all supported versions of the GDML files for a detector must be available in each release. If a geometry description is updated for a release of the software, the older versions must be kept, with the old GDML filenames, in order to ensure backwards compatibility, as data files are stored on disk and tape with the presumption of the old geometry. DUNE thus uses a simple additional versioning system by adding a version number to each detector's GDML filename. LArSoft and Gaseous Argon Software (GArSoft) jobs allow for FHICL configuration of the detector geometry GDML file. A LArSoft job, if run as input with data from an earlier LArSoft job, will check the compatibility of the requested GDML file and that used for the input. By default, if there is a mismatch, an exception is thrown and the job stops. Sometimes, however, it is desired to simulate and reconstruct Monte Carlo (MC) samples with different geometries, in order to simulate misalignments, for example. In order to accommodate these cases, the geometry consistency check can be disabled by setting an appropriate FHICL parameter.

Some software components, such as the DUNE plug-in components to the LArSoft geometry service, have different behaviors depending on which detector is being simulated or reconstructed. If an update to a geometry description for a detector requires a corresponding update to the software components, then a new name must be assigned to that version of the detector geometry to distinguish it at run time so the appropriate routines can be called. Software routines supporting older geometry versions will be maintained as long as the older geometry is supported.

GDML files include many similar code blocks to allow simulation of detectors with repeated, similar structures. The GDML files for the horizontal drift technology (FD1-HD) and vertical drift technology (FD2-VD) far detector modules and the ProtoDUNEs are made with Perl scripts and a number of small shell scripts to make final adjustments. Some of the inputs are not repetitive, such as material mixture definitions, and these fragments are hand-edited. GDML files for the near detectors are made with DUNENDGGD [35][1]. DUNENDGGD takes as input Python classes called "builders" and produces GDML files. The near detector GDML file creation requires additional steps, as two of the detectors move off-axis while a third remains stationary. Particles travel from ND-LAr to The Muon Spectrometer (TMS) and thus they must be simulated together, with a single GDML file. In addition to a full hall

[1] https://github.com/dune/duneggd

description, detector working groups may prefer to work with a geometry that only has one detector element in it for computational convenience.

Computational convenience also drives the need for workspace geometries for the far detector modules. A subset with only 12 anode plane assemblies instead of the full complement of 150 is a useful geometry to use for physics simulations, though care must be exercised regarding differences in event containment between the workspace geometry and the full geometry.

3.3.4 Neutrino Event Generation

Extracting physics results from the DUNE experiment requires comparing the observed data with simulation, which includes detailed simulations of the physics processes under study as well as of the response of the detectors. These physics processes can be simulated by any of several 'neutrino event generators', including GENIE [36], NuWro [37], GiBUU [38], NEUT [39], and others.

Neutrino event generators are used to simulate neutrino interactions within a detector volume. Input requirements are the neutrino beam flux, random number seeds and the detailed detector geometry. The output consists of four-vectors for final-state particles, where the final state of the interaction includes the decay of any short-lived particles and the subsequent particles produced. DUNE currently uses GENIE [40] v2.12.10 as the default, with the `DefaultPlusValenciaMEC` physics list. An upgrade to GENIE v3 is in progress. Other event generators may also be used.

A large fraction of DUNE's analysis effort will go into studying the effects of the neutrino-nucleus interaction models encoded in the generators [4]. Central values and systematic uncertainties for predictions of cross sections as functions of neutrino flavor and energy for each analyzed final state are needed for physics analyses. Some systematic uncertainties can be approximately evaluated by reweighting a central default sample generated with one generator with a specific choice of steering parameter values to mimic the output of the generator with a different choice of steering parameter values, or to that of a different generator entirely. This is not possible in all cases, due to underpopulation of some regions of phase space in the default generator. As has been the case with other high-statistics neutrino experiments, DUNE's physics working groups will participate actively in generator validation and tuning. It is not yet known how many full-statistics sets of simulated data will be needed in order to cover the non-reweightable generator systematic uncertainties, but an estimate is that between five and ten such sets will be needed.

3.3.5 Non-Beam Interaction Simulations

Simulations of non-standard model physics (e.g., neutron decay) and of cosmic ray and rock muon backgrounds are also used. Cosmic-ray simulations are performed with CORSIKA [41, 42] for detectors on the surface, and MUSUN/MUSIC for detectors deep underground [43, 44]. Radiological decays are modeled with BXDECAY0 [45] and LArSoft's radiological generator.

Factorization of the simulation into a generation stage and a detector simulation stage is a common practice in collider experiments, such as ATLAS and CMS. The fact that the interactions simulated by generators for collider physics happen inside an evacuated beampipe means that the details of the detector geometry and materials are not relevant for most event generation, and lists of four-vectors of particles emerging from a primary vertex will suffice. In a neutrino experiment, however,

the detector material is the target material, and hence the generators must be aware of the detector geometry and materials, which affects the structure and performance of the generator code. Currently GENIE, CORSIKA, MUSUN/MUSIC, BXDECAY0 are integrated with LArSoft. LArSoft also has custom generators for radiological decays and a text-file interface providing arbitrary four-vectors as inputs to the simulation. GENIE is also integrated with GArSoft.

DUNE can rely on strong international efforts to produce standard generators, but it should be noted that these efforts are in general external to the DUNE Collaboration. One challenge is the complexity of integrating multiple simulation code versions into our framework to ensure that there is timely evolution of the generators used as the physics understanding evolves.

3.4 Far Detector and ProtoDUNE Detector Simulations

A large amount of code re-use and sharing via the design of LArSoft has allowed the shared development of simulation algorithms for ProtoDUNE-SP, ProtoDUNE-DP, ProtoDUNE-HD and the FD2-VD detector proposals. Only the geometry, the field description, and the anode plane channel models need to be updated; the rest of the simulation chain is re-used. It should be noted that modules and workflows may need to be developed in order to apply detector-specific calibrations or physical effects, but that the general framework appears to be flexible enough to handle all such needs in simulation. The ND simulation differs substantially and is discussed below in Section 3.5.

Two classes of detector simulation for DUNE exist at the time of writing: parameterized fast simulations and full detector simulations that combine particle propagation codes such as Geant4 that record energy deposits in active elements with detector response simulations of electron drift trajectories, photon paths and electronics performance. The output of both the full simulation and the fast simulation can be used as input to make Common Analysis Format (CAF) files. The Common Analysis Format consists of a simplified ROOT-based ntuple and corresponding library of analysis tools, CAFAna, that was developed by the NOvA collaboration and is currently used by NOvA, DUNE, and SBND for the final stage of physics analysis.

3.4.1 Fast simulations

Parameterized, or "fast" detector simulations involve smearing truth-level physics quantities based on expected detector performance metrics, such as acceptance and energy resolution. These fast simulations are useful when optimizing detector designs, and for engaging physicists outside of the DUNE collaboration.

3.4.2 Particle Propagation Simulation

Full simulations are based on detailed geometry models and Geant4 [46, 47], and are needed for computation of precise physics sensitivities.

A detector simulation such as Geant4 [47] or FLUKA [48] is used to simulate the interaction with the detector material and signals produce by the particles described by the four-vectors produced by the event generator. Simulation input requires the geometry and 4-vectors, while the outputs consists of the true interactions and energy deposits in the active detector materials. Energy thresholds need

to be quite low O(100s keV), as the detector systems are sensitive to physics processes at MeV-scale and above. The large uniform detector volumes in liquid argon (LAr) detectors compensate for the low thresholds by requiring fewer Geant4-volume boundary crossings in a typical particle trajectory. Hadronic cross sections on LAr are not yet fully understood. DUNE has developed a reweighting framework Geant4Reweight [49] to allow reuse of existing simulation as interaction rates are refined. Due to the high granularity and low energy thresholds, interaction records can become very large. Simulated ProtoDUNE-SP single drift-time trigger records for 6 APAs are currently 200-300 MB, substantially larger than the raw waveforms coming from the actual detector.

3.4.3 Detector Response Simulation

The propagation of low-energy drifting electrons and scintillation photons, signal induction and electronics response are then simulated in a separate step.

LArTPCs require detailed models of electron trajectories through a potentially charged fluid. Photon detection requires ray tracing over long distances. Once electrons and photons have been propagated to the detection systems, simulation of the electrical response must also be done. Inputs to the detector response simulation are the output of the detector simulation, charge distribution in the liquid, absorption parameters and the detailed readout system geometry. Outputs are simulated streams of bits in the electronics.

Drifting electrons are simulated parametrically using a model based on the measured drift velocity, longitudinal and transverse diffusion coefficients, and a parameterized model of space charge. This last effect is particularly pronounced at ProtoDUNE-SP and ProtoDUNE-DP due to the large number of cosmic rays crossing the detector volume, giving rise to distortions in the apparent positions of particles of up to 30 cm. Based on the experience gained from unintentionally grounded electron diverters in ProtoDUNE-SP, external imperfections need to be simulated as they can be the cause of field distortions.

Once the electrons drift to the anode plane in wire-based LArTPCs in the simulation, a detailed 2D model of the wire responses is applied using Wire-Cell Toolkit (WCT) [50, 51, 52]. The two dimensions are wire number and time, and the effects of induced currents on neighboring wires are included in the simulation. The electronics response function is folded in to a final model of the observed waveforms. Simulated waveforms have been compared with real ones in ProtoDUNE-SP and are found to be very similar. In the pixel-based ND-LAr and ND-GAr, the electronics simulation is at a simpler level, as the electronics have not been fully demonstrated.

ProtoDUNE experience indicates that, although simulation is reasonably fast relative to reconstruction (3 to 1), memory utilization, even for ProtoDUNE, is very high in simulation jobs. Process sizes of 5-6 GB in memory are typical for trigger records in which six anode plane assemblies are read out. High energy FD interactions are expected to span more detector volume (although not the full size of the detector) and thus require even more memory. Multi-threaded solutions are being investigated, but a large fraction of a job's memory footprint is occupied by trigger record data and not shareable memory such as code segments and geometry description. Efforts are focused on reducing the per-thread memory requirements. Multiple simulation and reconstruction passes may be needed since calibration of the electron drift model incorporating space-charge effects requires a reconstructed dataset, and the output of this calibration is used for subsequent simulations and reconstruction. A similar calibration is required for the recombination model.

3.4.4 Photon Detector Simulation

Photon simulation in large detectors is known to be highly computationally intensive due to the need to trace large numbers of photons over large distances. DUNE has several approaches to this problem.

The sequence for photon simulation is:

1. perform particle track simulation in Geant4 to produce the energy deposits along the track;
2. calculate the number of photon/electron emissions at each vertex where energy is deposited;
3. simulate the photon transport in the detector (either full simulation or fast simulation); and
4. reconstruct the photons into offline objects such as individual PDS hits and collective flashes of hits.

In LArSoft, after particles are propagated using Geant4, energy deposits along tracks are recorded so that the number of ionization electrons and scintillation photons generated at each step can be estimated using the available ionization and scintillation methods. Once the number of photons is determined, the fraction of those photons that will actually reach a given photon detector is usually estimated using one of several fast light-simulation methods. This procedure can be followed for both 128 nm photons (Ar scintillation) and 176 nm photons (Xe scintillation) to account for the wavelength-dependent Rayleigh scattering in the simulation to cover the possibility of Xe-doped LAr. Since a copious number of scintillation photons (25000 γ/MeV at 500 V/cm) is produced in LAr, it is very demanding computationally to propagate all photons individually using Geant4. Instead, fast simulations for photons are implemented. The full Geant4 photon simulation CPU time per event depends on the energy deposited. In the fast simulation methods, using a library, semi-analytic methods or machine-learning, the CPU time depends on the granularity of the detector. Generally speaking, the performance of the three fast simulations methods is at the same level and the usage of any particular choice will depend on the physics requirements of the sample.

3.4.4.1 Optical Library Method

This method consists of dividing the cryostat volume into smaller parallelepiped-shaped regions called voxels and creating a lookup table (the optical library) that stores the visibility of each photon detector to photons generated within a given voxel. This optical library is created using the full Geant4 simulation to generate photons anywhere inside a given voxel, with random direction and polarization, and then store the fraction of those photons that land on the optically sensitive region (visibility) of a given photon detector, identified within LArSoft by its optical channel. When using the fast simulation, LArSoft will retrieve the Optical Library and store its information to directly transform the number of photons generated in a given step along a particle's track into the number of photons landing on each optical channel. This method can satisfactorily be used as a fast simulation method, but nevertheless, its performance greatly depends on the size of the voxel and the number of photons being generated per voxel. Increasing the number of voxels in a library will improve the description and reduce the bias at the cost of a large increase in memory consumption. Increasing the number of photons per voxel will provide much better statistics and also largely increase the amount of time dedicated to generating the optical library.

3.4.4.2 Fast Simulation with Generative Neural Networks

This method relies on a generative neural network trained on the photon detection system. The input to the network is the vertex where the photons are emitted, and the output is the mutual visibility of each photon detector/emitter pair. The generative model can be trained ahead of time using a full Geant4 optical photon simulation with photons emitted from random vertices in the detector, after which it can be frozen to a computable graph and deployed to the production environment (LArSoft framework).

When the computable graph is loaded in LArSoft, it quickly emulates photon transport by computing the visibility of each photon detector calculated from the photon emission vertex along the particle's track. This method is 20 to 50 times faster than the Geant4 simulation while keeping the same level of detail for particle tracks, such as the number of energy depositions and the precision. The model inference also requires a relatively small amount of memory. The samples for candidate far detector-like geometries show the required memory for the model inference is around 15% of the Geant4 simulation. Further, this memory use is not directly correlated to the size of the detectors.

3.4.4.3 Semi-Analytical Photon Detection

In these methods, large numbers of photons are generated at different points within the cryostat volume and propagated using Geant4. Once those propagation distributions have been simulated, Gaisser-Hillas functions are fitted to the number of photons reaching the photon detectors as a function of source-detector distance and relative angle. The resulting parameters are used during event simulation in order to extract the fraction of photons produced at a certain point that arrives at a given sensor.

3.4.4.4 Comparison Between Fast Simulation Methods

A semi-analytic model and the use of optical library models for fast simulation have been compared to the full light simulation in Geant4 by the SBND experiment. Their studies used a highly segmented and large-photon-count optical library, created with ∼1.6M voxels, with 0.5M photons being generated in each voxel (total of 7.9×10^{11} photons), resulting in a 1.2 GB file size. A similar optical library for the DUNE $1 \times 2 \times 6$ geometry (volume of $7 \times 12 \times 13.9$ m^3) would be prohibitively large.

All DUNE optical libraries produced so far have larger voxels and fewer total photons simulated per voxel in comparison to SBND, one optical library of which was used as reference for the comparison of the two modes. It has been reported that the optical library struggles to properly describe light signals generated closer to the detectors and more on-axis (up to $\sim 50\,\mathrm{deg}$). This is a known issue caused by the intrinsic discontinuity of the voxelization schemes. In the ProtoDUNE optical library, the visibilities are smoothed using neighboring voxels to minimize this effect.

However, SBND, observes better performance from the semi-analytic model, with improved resolution close-on axis (3.6% vs 5.6%), and less than 1% bias. In contrast, the optical library is systematically biased (2.5-4.9%), in particular for the larger/closer signals. These differences, together with the very high memory consumption (of several extra GB) during simulations when using an optical library, motivate the choice of the semi-analytic model as the default for fast simulations in the DUNE FD.

3.5 Near Detector Simulations

3.5.1 Common Simulation Tools

The near detector simulation chain is diagrammed in Figure 3.3. The flux, geometry and generator stages are common across the ND detectors. The model used to predict the neutrino flux at the near detector is a Geant4-based simulation of the incident proton beam, the target, the focusing horns, decay pipe and the hadron absorber, called g4lbnf [53], and the dk2nu package. The same simulation is used to predict the flux at the far detector, which differs from that at the near detector due to the angular and position acceptance of the near detector, as well as neutrino oscillations. Uncertainties on the flux are estimated using the Package to Predict the FluX (PPFX) framework developed by the MINERvA collaboration [54, 55]. The detector geometries are generated with the DUNENDGGD package which produces a detector suite in the hall in GDML format, as described in Section 3.3.3. Neutrino interactions are simulated with GENIE version 2.12.10, with the `DefaultPlusValenciaMEC` physics list. [36]. An upgrade to GENIE 3 is in progress. Particles exiting the struck nucleus are propagated through the detectors using a Geant4-based model called edep-sim [56], which produces ROOT trees containing the simulated energy deposit information and associated truth labels. Detector-response simulations are handled independently for each subdetector as described below.

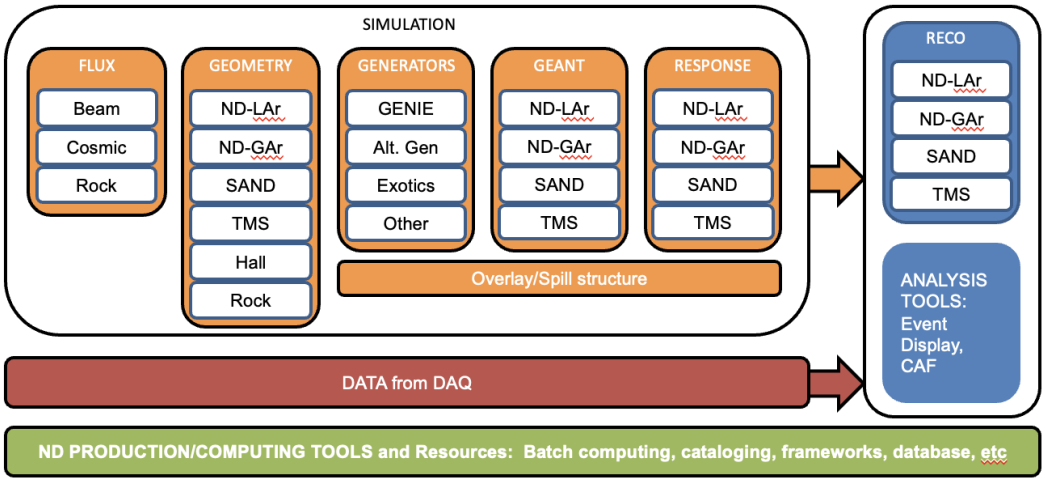


Figure 3.3: This diagram shows the components of the Near Detector simulation and reconstruction workflow. Common tools include flux models, geometry specification tools, event generators, and GEANT simulation. The detector response and reconstruction is special to the subdetectors, and the reconstructed data are collected in common analysis format (CAF) files. Data are to be reconstructed in the same way as simulated data.

3.5.2 ND-LAr Detector Simulation

The ND-LAr detector simulation is performed through a set of fully GPU-optimized algorithms. The software, called `larnd-sim` [57], is written entirely in Python and compiled on the GPU using the Numba wrapper for CUDA. This highly-parallelized implementation provides orders of magnitude processing speed improvements compared to a classic CPU one.

The software takes as input energy deposits in the active volume as simulated by `edep-sim`. The

edep-sim output ROOT files are converted into the HDF5 format by a dedicated script. The simulation takes into account quenching and drifting effects for the ionization electrons in the liquid argon and calculates the current induced on the pixels placed at the anode of each module. Subsequently, the electronics simulation produces the analog-to-digital counts and timestamps in the same format as the LArPix ASICs [58]. The number of photons reaching the light readout system is calculated using a GPU-optimized lookup table. The output is saved in the HDF5 format.

The high flexibility of the software allows easy implementation of different detector geometries. The software has already been employed to perform comparisons between the simulation and the data coming from the ArgonCube Module-0 and Module-1 runs [59] and it has been successfully used to produce full-spill ND-LAr simulations.

3.5.3 TMS Detector Simulation

Currently detector response is handled with resolutions incorporated at the hit level by smearing reconstructed quantities. A full detector response simulation is in progress.

3.5.4 ND-GAr Detector Simulation

The GArSoft suite of art-based modules is used to simulate the ND-GAr detector. It performs similar functions to LArSoft's Geant4 and detector simulation modules. GArSoft includes interfaces to the following event generators: GENIE, CRY [60, 61], and a text-file generator. It also contains a LArSoft-like interface to Geant4, called garg4. In order to be used in conjunction with the other near detectors in a full hall simulation, GArSoft also includes a module to import energy deposits from edep-sim, and format them internally as if they had been made by garg4.

Once the energy deposits are available, a module that simulates the division between ionization and scintillation quanta is invoked. Electron drift through the gaseous medium is parameterized, and longitudinal and transverse diffusion are modeled by sampling from Gaussian distributions of widths that grow with the square root of the drift time. The effect of attenuation due to attachment of drifting electrons on electronegative impurities in the gas is also simulated at this time.

The anode planes are modeled as re-purposed ALICE inner readout chambers (IROCs) and outer readout chambers (OROCs) [62]. Charge is amplified by avalanches on the anode wires, and induced charge on the nearby readout pads provides the charge detection. The pad response functions are taken from Ref. [62].

The response of the calorimeter and the muon systems are parameterized versions of the energy deposit information, localized to the detector cells in which the energy is deposited.

3.5.5 SAND Detector Simulation

3.5.5.1 SAND Simulation Components

The SAND detector response simulation is performed by several tools developed in C++ and Python. The digitized detector response waveforms are built starting from the energy deposits provided by edep-sim, after the simulation of the detector response.

- ECAL: The energy deposits in the scintillating fibers are converted into the number of generated photons randomly extracted by a Poisson distribution with mean of 25 photons per MeV of energy deposited. The number of photo-electrons is obtained taking into account the attenuation according to the fiber length traveled by the photons. The arrival time to the photo-sensors is assigned to each photo-electron taking into account the scintillation decay time, the propagation time along the fiber and a random jitter. The amplitude of the signal of the photo-sensor is proportional to the number of photo-electrons collected in a certain time window and the time of the signal is obtained simulating a 15% constant fraction discriminator.
- STT: For each energy deposit in a straw tube, the electronic signal arrival time at the read-out is evaluated considering the electron drift time to the wire, the electric signal propagation along the wire and a random jitter. The timing of the signal is given by the earliest electric signal reaching the tube read-out. The amplitude of the signal is proportional to the sum of the energy deposit in a certain time window.
- GRAIN: Argon scintillation in GRAIN is simulated using Geant4. For each energy deposit, a random number is extracted from a Gaussian distribution, with the mean value determined by the argon light yield and the fraction of the energy deposit that ends up producing photons (already calculated by edep-sim), and a dispersion given by the square root of this number. If the mean is less than 20, a Poisson distribution is used instead. For each photon, a random momentum, polarization, position along the step, time of emission, and energy are determined. The properties of argon are included: the singlet to triplet ratio [63], the fast and slow scintillation time constants [63], Rayleigh scattering [64, 65, 66], and the absorption length [67]. The photons are then collected by cameras. A camera is composed of an optical system (either mask or lens) and a (32x32) silicon photomultiplier (SiPM) matrix as photo-sensor. The simulation includes the geometry of the cameras, whether being lens-based or mask-based. The number of photons collected in each channel in a given time window and the time of the first photon impinging in each channel is stored for producing a two-dimensional image for each camera. The photo-sensor response is simulated with custom software written in Python and OpenCL for GPU-accelerated computing. The SiPM matrix response waveform is computed with a heuristic function from the timing of impinging photons on the matrix surface. The SiPM characteristics such as PDE, crosstalk, afterpulses, and dark count rate are included in the simulation. A small quantity of white and rf noise is added as well. The waveform is then quantized and integrated over a fixed time window to simulate a DAQ composed by an ADC and a TDC for timing over a fixed threshold.

3.5.6 Overlays

In order to build full beam spills simulated events on rock, detector fiducial volumes and the corresponding anti-fiducial sample are overlaid to reproduce expected beam timing and intensity. The OverlayGenie package is the current overlay tool in use but alternatives which are more integrated are being developed. More details are listed in 3.7.1.

3.6 ProtoDUNE Simulation Experience

The FD/ProtoDUNE LArSoft-based detector simulation framework has already been tested successfully in ProtoDUNE and is described more fully in references [14, 17].

The ProtoDUNE-SP simulation includes beam particles, cosmic ray interactions and radiological backgrounds. The beam particle species and momentum distributions are from the Geant4 simulation of the H4-VLE beam line at CERN, which consists of e^+, π^+, p, and K^+ particles at 0.3, 0.5, 1, 2, 3, 6, and 7 GeV/c. Cosmic ray interactions are produced with the CORSIKA generator. Radiological backgrounds, including ^{39}Ar, ^{42}Ar, ^{222}Rn, and ^{85}Kr, are also simulated using the RadioGen module in LArSoft. The primary particles are tracked in the LAr using the Geant4 package. The ionization electrons are drifted towards the wire planes and the effects of recombination, attenuation and diffusion are simulated. The accumulation of positive ions in the detector modifies the trajectory of ionization electrons and the strength of the E field, which is known as "space charge effects"; this is simulated using the measured distortion and E field maps. The electronic signal is simulated by convolving the number of electrons reaching each wire plane with the field response and electronics response. The field response is modeled using the Garfield [68] package. The electronics response uses the parameterization from a Simulation Program with Integrated Circuit Emphasis (SPICE) simulation with the average gain and shaping time measured in the ProtoDUNE charge injection system.

3.7 General Simulation Considerations

In addition to the specific needs for particular detectors there are additional common considerations for simulation.

3.7.1 Overlays or Mixing

An efficient method for accurately simulating backgrounds in high-rate or high-noise environments is to overlay the simulated interaction information on real data. For example, the environment in ProtoDUNE or the near detector includes multiple beam and rock muon interactions in the same readout as the interactions of interest. Libraries of properly-formatted raw data can be combined with simulated hits from a simulated event to yield a realistic simulation of an interaction in the real detector. This requires a library of such interactions and careful matching of the running conditions of the external raw events to the desired simulated events. It is also important to include the same electron lifetime and space-charge effects in the signal simulation as are present in the overlaid data, in order for the combined data to be consistent.

DUNE has not yet integrated overlays into the ProtoDUNE or FD frameworks. To do this we will need to distribute large volumes of overlay data to remote sites while maintaining randomness. Other experiments such as MicroBooNE and MINERvA have done this successfully.

3.7.2 Reweighting

If intermediate simulation steps are stored they can be used to reweight interactions to reflect new knowledge about the underlying cross sections, neutrino flux, detector materials or detector properties. Examples include the PPFX beamline simulation reweighting system and the implementation of different cross section and final-state interaction models in event generators. In the case that a particular portion of phase space is under-represented by a generator, or not represented at all, but nonetheless is populated by a desired alternative generator, additional samples must be generated with the alternative generator.

3.7.3 Outputs from Simulation

At this point we have simulated data that mimic the raw data coming from the detector, and the next steps are calibration and reconstruction. In all cases, descriptive metadata need to be produced that facilitate locating the simulation samples on persistent media, understanding their contents, and reproducing them if need be.

3.8 ProtoDUNE and Far Detector Reconstruction

A detailed description of the ProtoDUNE-SP and Far Detector reconstruction algorithms is given in Ref. [53]. This section outlines those aspects of the reconstruction processing that are directly related to software and computing issues. Figure 3.1 shows the data flow for regular reconstruction in ProtoDUNE. There are, in principle, multiple stages in reconstruction that are each well suited to different computer architectures. We expect the full FD data processing to follow a similar path to that used by ProtoDUNE.

Figure 3.4 illustrates readout structures for full DUNE FD modules. A readout consists of a large number (up to 150) of 30 MB APA readout fragments and a number of smaller readout fragments that are much smaller than a 30 MB APA fragment. Details of trigger record volume can be found in Tables 6.2 and 6.4. These need to be processed and recombined into a single readout record.

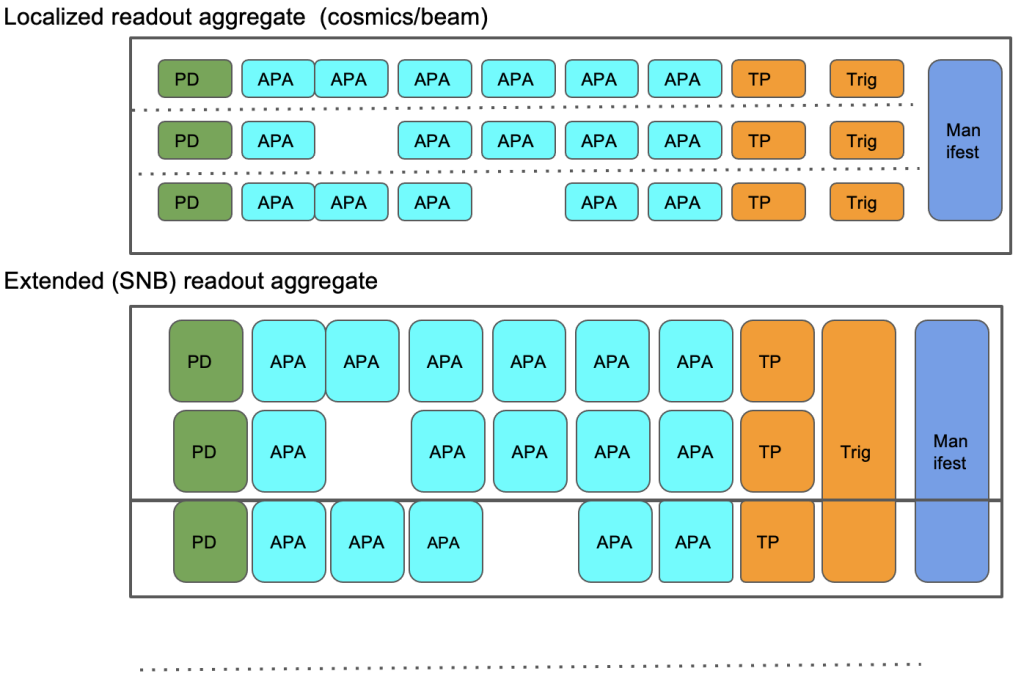


Figure 3.4: Data aggregation cases for the far detector. The top case shows information for normal beam or calibration readouts. A single file of ~ 10 GB size contains several complete trigger records with their boundaries designated by the dashed lines. TPC APA's, PDs, trigger primitives and a trigger are recorded for each trigger readout. In addition, a manifest, which describes the relations between the data, is stored either in the file or in external metadata. The bottom case is a SNB readout, in which thousands of 5-10 ms time slices must be read out over 100 s. The solid lines denote file boundaries. How data are ordered, by geographical position or by time, is not yet specified.

Each trigger record contains a large number of waveforms, one for each readout channel. In ProtoDUNE-SP, the waveforms typically were 6000 time ticks long, with one analog-to-digital converter (ADC) sample per time tick. A detailed description of the reconstruction procedures, starting with these waveforms, is given in [52], and summarized briefly here.

3.8.1 Signal Processing

The initial offline processing stage, labeled "signal processing" in Figure 3.1, holds particular significance and challenges. Such a stage must be applied to data from both the PD and TPC readouts. It is the latter which poses the bulk of the challenges and here it is the focus. The source of the challenges is simple. The processing is relatively intensive in terms of CPU and memory as evaluated on a per-trigger basis and it is data intensive as it must be applied to the entirety of the TPC readouts. Unlike subsequent stages, there is no reasonable pre-selection criteria to reduce the input to this stage which has not already been exploited by the DAQ. The initial stage consists of a sequence of two primary data transformations.

The first is known as data preparation or noise filtering. The raw data from the DAQ is input after being decoded and loaded into dense per-channel ADC waveform arrays. The transformation then attempts to remove features unrelated to those caused by ionization electron signal while at the same time producing no significant alteration to those signal components. One important goal of this processing is to attenuate non-thermal noise such as that due to RF emissions in the cryostat and which induce waveforms coherently across many channels. Also mitigated are unwanted waveform features such as ADC bits with error codes ($\sim$1%) and transient artifacts. Some general-purpose algorithms have been developed by the LArTPC community but substantial effort is required to understand and invent solutions to problems specific to each given detector. The output of this first step is effectively still in the form of full, dense ADC waveforms.

The second major transform is known as signal processing. It produces "signal regions-of-interest" (signal-ROI). It does so in two tightly coupled steps. First, a 2D model (in the space of channel vs sample time) of the per-plane detector response is deconvolved from the prepared, filtered ADC waveforms. This produces waveforms which are in units of number of ionized electrons in the region near a wire and per time sample. However, due to the bipolar nature of the detector response on the induction planes, this deconvolution necessarily amplifies low frequency noise. Left alone, this noise would overwhelm the resulting signal waveforms. To combat this, another algorithm operating in the time domain identifies small, regions-of-interest where signal is found to be above the low-frequency noise. Within each signal region of interest, the baseline of the waveform fragment is recalculated, thus effectively operating as an adaptive high-pass filter. The resulting data consists of very sparse, unipolar waveform fragments and substantial data reduction is achieved while retaining almost all signal information. This is a key operation for any induction-based LArTPC detector and the current best incarnation was developed and vetted in MicroBooNE [69, 70] and applied to ProtoDUNE [71] and has since been applied to many other detectors including the prototypes for DUNE FD2-VD. These latter, strip-based detectors pose new challenges related to the required precision of the detector response model. This is an area of ongoing study but initial results show good performance.

This initial stage (both major transformations) can run in parallel, multi-thread form at the level of data from one APA (or equivalent detector module unit). In particular, the WCT [50, 51] in which the signal-processing and some forms of noise filtering are developed, implements an execution model that

supports this level of parallelism. Furthermore, the Fast Fourier Transform (FFT) and other algorithms used heavily in this stage benefit from SIMD acceleration [72] such as provided by GPU devices. These forms are also supported by and being further developed in WCT implementations. This execution model provides flexibility to scale this key stage from ProtoDUNE to DUNE by mapping these two very different data volumes to different classes of computing facilities.

3.8.2 Reconstruction Strategies

Two primary reconstruction strategies have been developed in the LArTPC community, have been applied to ProtoDUNE and are expected to be applied to DUNE FD. They may be classified as "view-first" vs "slice-first". Note that representations of ProtoDUNE-SP trigger records of different approaches can be seen in Figures 3.9 and 3.10.

In "view-first", data from each wire plane view is initially processed. Peaks are found in the waveform fragments output by the signal-processing and each is fit to a Gaussian model in a process called "hit-finding". The Gaussian model is convenient as a baseline for hits, but it should be mentioned that not all waveforms will be Gaussian in shape and that there are limitations to this algorithm. These hits are then used as inputs to reconstruction algorithms, such as the SpacePointSolver [53], Pandora [73], TrajCluster [74], and Projection Matching Algorithm (PMA) [75], which identify clusters, tracks and showers in 3D by associating objects in the three 2D views. The calorimetry modules sum up and calibrate the charge deposits for use in energy reconstruction and particle ID (PID). The parameters of the clusters, tracks and showers are stored in ROOT trees that end users can analyze rapidly and repeatedly.

In "slice-first", data from one time slice across all wire plane views is initially processed. The slice is chosen to span four to six samples in time which exploits the fact that signal processing naturally produces over-sampled waveforms and thus the data may be further reduced with no information loss. Next, a tomographic inversion is performed for each slice with an algorithm called "wire-cell" [76], which is provided by the Wire Cell Toolkit (WCT). Stacking the per-slice inversions over the full readout results in sparse 3D regions of the detector volume that localize ionization electrons in a manner consistent with the original signal waveforms. The ambiguities inherent in the tomographic nature of the TPC also leads to portions of the solution not truly containing any ionization. Further constraints, such as consistent charge in all three views and simple connectivity between regions of neighboring slices, are applied to remove or reduce these ambiguities. This is done in a model-free manner in order to not bias the results. When bias-free constraints are exhausted, a local track-like model is used to allow collapsing regions of ambiguity, making the results yet more sparse and more tightly surrounding regions containing ionization. These results finally enable very accurate reconstruction of quantities such locations of interaction vertices, track branch points, and dQ/dx and from there dE/dx along tracks. These algorithms were initially developed for MicroBooNE [77] where they have been found to outperform others [78, 79]. There is an ongoing effort to generalize and "port" these algorithms into WCT for use by DUNE and other LArTPC detectors.

Either strategy results in a 3D model of the ionization distribution which is composed of a number of spatial clusters connected in some way through the 3D space of the detector volume. With these clusters as input, a process called "flash matching" attempts to associate processed PD signals with the TPC clusters in order to determine when in time or equivalently where along the drift axis the cluster of ionization occurred. The offline software must arrange for these two data flows to merge on

a per-trigger-record basis.

From this relatively low-level reconstruction, various high-level reconstruction techniques follow. For example, separating track-like energy deposits from shower-like deposits is a key part of many DUNE analyses. In one case, this is accomplished with a convolutional visual network (CVN), `EmTrackMichelID`, listed in Table 3.1. The signal processing is run in two modules: "caldata" includes all but 2D deconvolution and signal-finding which are handled in "wclsdatasp". It is one of the most CPU-intensive operations in the ProtoDUNE-SP reconstruction chain, when the algorithm is run on a grid node lacking a GPU. Recently, however, a GPU As A Service (GPUAAS) technique has been developed [80], enabling a speed-up on the order of a factor of ten, though it depends on the ratio of CPU-only nodes to GPU resources.

Table 3.1: Wall-clock module execution times for the reconstruction of a typical ProtoDUNE-SP event, in seconds on a 2020 vintage processor with an HS06 rating of 11 [81]. Processes taking less than 1 second are not shown. The event is a data event from Run 5809, a 1 GeV beam run. Note that the electromagnetic shower reconstruction module EmTrackMichelId dominates.

Module Label	time/event (sec)
RootInput(read)	0.14
beamevent:BeamEvent	1.6
caldata:DataPrepByApaModule	83.0
wclsdatasp:WireCellToolkit	79.9
gaushit:GausHitFinder	1.6
reco3d:SpacePointSolver	9.4
hitpdune:DisambigFromSpacePoints	1.5
pandora:StandardPandora	39.9
pandoraTrack:LArPandoraTrackCreation	4.6
pandoraShower:LArPandoraShowerCreation	3.7
pandoracalo:Calorimetry	2.1
pandoracalonosce:Calorimetry	1.9
pandoraShowercalo:ShowerCalorimetry	3.3
pandoraShowercalonosce:ShowerCalorimetry	3.2
emtrkmichelid:EmTrackMichelId	233.8
canodepiercerst0:T0RecoAnodePiercers	1.1
pandora2Track:LArPandoraTrackCreation	11.6
pandora2calo:Calorimetry	4.9
pandora2calonosce:Calorimetry	4.5
pandora2Shower:LArPandoraShowerCreation	4.2
pandora2Showercalo:ShowerCalorimetry	4.2
pandora2Showercalonosce:ShowerCalorimetry	3.8
RootOutput(write)	2.8
Total:	**507.9**

3.9 Near Detector Reconstruction

The ND Reconstruction primary exists as separate reconstructions for each individual ND sub-detector (described below). The ND software group are developing a common data model which will allow inter-detector software connection. For instance, an event matching algorithm between the ND-LAr and TMS has been developed. Output from the Reconstruction will be written to common analysis files (CAFs) that will enable physics analysis (Figure 3.5). The current ND data model is outlined in the DUNE ND Data Model document [82].

3.9.1 Near Detector Liquid Ar (NDLAr)

Reconstruction algorithms for the ND-LAr include both traditional and machine-learning based chains. The primary reconstruction is an automated machine-learning based package which has been demon-

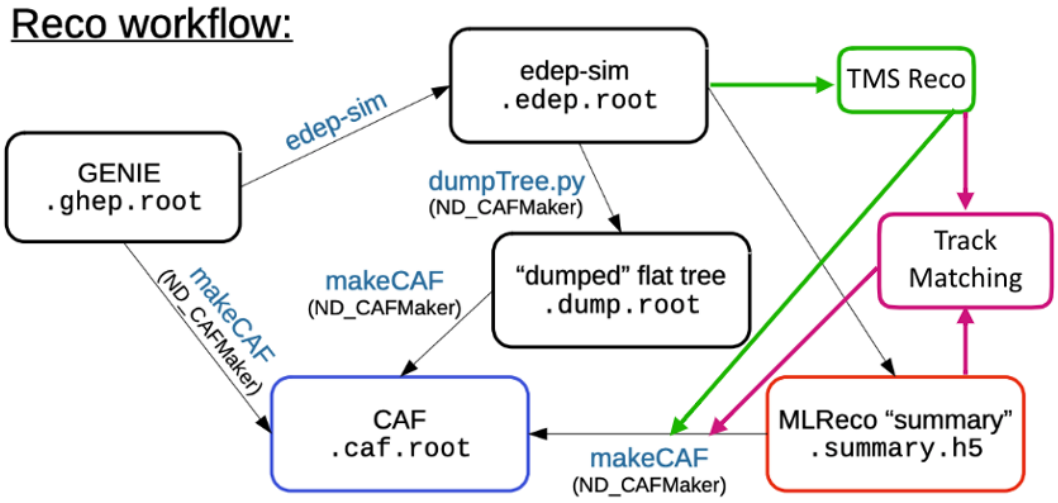


Figure 3.5: An example ND reconstruction workflow is shown for simulation with TMS and LAr (MLReco) data producing track matching information and stored in CAF files.

strated on full-spill events simulated in the native 3D detector readout. A parallel Pandora based reconstruction is also being prepared. These reconstruction algorithms are currently being tested on both simulated and prototype data. Rough estimates for CPU processing time are listed in Table 6.7.

3.9.1.1 ND LArTPC ML Reconstruction

A full data reconstruction chain for the LArTPC at DUNE ND consists of multiple machine learning algorithms optimized for multiple reconstruction tasks. An overall software design can be found in [83] and details of individual algorithm can be found in [84, 85, 86, 87, 88], including evaluations performed on a public LArTPC simulation dataset [89]. This reconstruction software is based on earlier development of applying powerful deep learning methods to LArTPC experiments [90, 91, 92]. In particular, these algorithms include deep convolutional and graph neural networks as well as analytical physics models (e.g. Multiple Coulomb Scattering for estimating momentum). The reconstruction chain takes voxelized 3D images (i.e. data recorded by the detector) as an input, and produces output that organizes the following information:

- A list of neutrino interactions
- A list of particles in each neutrino interaction
- Attributes of individual particles
  - Particle type and kinematic information
  - A cluster of pixels that resemble the particle's trajectory
  - Particle hierarchy (e.g. a parentage information for a particle produced through a decay process)
  - Calibrated uncertainty quantification for (most of) the attributes listed above

This reconstruction chain is end-to-end optimizable: all algorithms can be optimized by a gradient-based technique [93] simultaneously in a completely automated fashion, unlike traditional software, without requiring any human interventions. This saves weeks and months of manual tuning process required for

a traditional counterpart. The software is developed on top of the CUDA programming ecosystem [94] and the PyTorch machine learning library [95], which allow the use of NVIDIA Graphics Processing Units (GPUs) to accelerate computation. Furthermore, the implementation allows multi-GPU parallelization and is also compatible to distributed computing (i.e. inter-node parallelization exploiting fast network interconnects such as infiniband to expand parallelization beyond a single server) to unlock the power of modern infrastructures including High Performance Computing (HPC) clusters. While many deep neural networks require excess amount of computing power and are thus often impractical to execute on CPUs, this software can be run on CPUs within a reasonable time thanks to an implementation of innovative sparse matrix multiplication that exploits the sparsity of LArTPC image data. In terms of computing needs, the process of optimization is typically performed on a single NVIDIA A100 GPU and takes about a week starting from scratch. It takes a day or two for a typical fine-tuning of the chain when the underlying detector physics modeling changes. Discussion of access to and resource estimates for GPU-based algorithms is discussed in more detail in Section 7.6.3.

3.9.1.2 Pandora based Reconstruction

Pandora is a multi-algorithm approach to particle reconstruction that has been used effectively for other LArTPC detectors, including the MicroBooNE detector and the ProtoDUNE detectors. As it is currently a leading reconstruction option for the DUNE Far Detector, it is thus ideal that it can also be adapted to work with the DUNE Near Detectors. Currently the effectiveness of the pre-existing 3x2D reconstruction chain when used with ND-LAr is being assessed, with the plan being to develop a native 3D approach and tailored algorithms as the next stage of work.

3.9.2 TMS

The output of the TMS reconstruction will inform the ND-LAr reconstruction as to which tracks exited the ND-LAr and were reconstructed in the TMS. A stand-alone TMS-reconstruction also enables monitoring of the inclusive neutrino event rate, using neutrinos interacting on the TMS fiducial volume.

The current TMS reconstruction approach is running the track finding and reconstruction simultaneously through a Kalman filter, or having a separate track finding algorithm identify track candidates which it feeds to the Kalman track reconstruction.

3.9.2.1 Track finding

For the track finding stage, we are currently investigating three approaches. They all run in each view separately, and the 2D to 3D track matching and merging is done at a later stage.

- Perform a Hough transform [96] in each view separately, and find which hits intersect the primary Hough line. Traverse the hits and group neighboring hits into a track candidate. Take remaining hits (not belonging to a candidate) and repeat process to see if there is a N^{th} track candidate. Due to the bending in the $x-z$ view, the Hough transform is often more successful in the $y-z$ view, and the neighbor clustering in $x-z$ is critically important.
- Sort the hits in z, and run a custom A* [2] path finding algorithm from first to last hit in z. The intent is to find the shortest path between the first and last hit, only traversing cells that have

[2] https://en.wikipedia.org/wiki/A*_search_algorithm

been hit. If no path is found from start to finish, repeat A* path finding by running the algorithm until the hit with second largest z, and continue.

- Sort the hits in z, and run a Kalman filter from first to last hit in z. For this to succeed, the Kalman filter needs to account for the energy-loss, bending and multiple-scattering happening in the iron and scintillator layers.

3.9.2.2 Track reconstruction

Currently, the lepton candidate is assumed to be the longest track in the track finding. After the reconstruction stage, it is envisioned the lepton candidate is assumed to be the most energetic track that fits a muon hypothesis. Depending on the final design of the TMS, the track reconstruction will operate on 2D or be combined into 3D views.

Preliminary studies show track length is by far the most important feature for a good muon KE measurement, with the energy deposits and track bending being secondary contributions. The current Kalman implementation in the TMS software includes effects from energy loss and multiple scattering, inspired by similar implementations in MINOS, MINERvA, and T2K. Efforts towards integration of the ND-LAr and TMS reconstructions are also ongoing.

3.9.3 Pixel-Based Gaseous Argon TPC Reconstruction

The ND-GAr consists of two primary detectors – a copy of the ALICE pixel-based TPC in a 10-bar gas consisting predominantly of argon, surrounded by a calorimeter and a superconducting magnetic coil. A muon system is envisaged outside of the calorimeter but is not included in the simulation or the reconstruction at the time of writing. Data are unpacked and hits are found on the per-readout-pad waveforms, similarly to how the initial stages of reconstruction for a LArTPC are followed. Hits are then clustered into TPC clusters, which reduces the memory and CPU usage of subsequent steps, and also increases the spatial resolution of individual clusters. Vector hits are found by grouping TPC clusters into short line segments, and the vector hits themselves are grouped into track candidates by a pattern recognition module. A track fit based on a Kalman filter then finds the best estimates of the track parameters. Vertices are then found using tracks with nearby endpoints. Because there is a cathode in the middle of the drift volume in the nominal ND-GAr design, a cathode stitch module is then run to associate track segments on either side of the cathode, bring them together by solving for the best interaction time that lines the segments up, and also moves the associated tracks and vertices. The calorimeter consists of a mixture of strips and pads. Calorimeter hits are found in the SiPM waveforms provided by the calorimeter DAQ, and they are clustered together to form reconstructed three-dimensional energy deposit objects with positions, directions, and energies. Calorimeter clusters are associated with tracks, and they must match both in position and in time. Before association with a calorimeter cluster, a track's position along the drift direction is uncertain due to the unknown interaction time. The nanosecond-scale timing resolution of the calorimeter allows the updating of track positions along the drift direction.

3.9.4 SAND

For neutrino interactions in the Straw Tube Tracker (STT), a first rough estimation of the vertex position is based on the STT digit spread in both the XZ and YZ views, where Y is the vertical axis, Z is the projection of the beam direction on the horizontal plane and X is perpendicular to the first two in

a right-handed coordinate system. The vertex position is then used to apply a conformal transformation to the coordinates of the YZ digits $(y, z \rightarrow u, v)$. The YZ tracks are then identified as clusters of digits in the v/u distribution. The particle momentum in the bending plane is estimated with a circular fit of the YZ digits. To measure the dip angle, a transformation is first applied to the coordinates of the XZ digits $(x, z \rightarrow x, \rho)$ to linearize the trajectory, followed by a linear fit. A Kalman Filter based track reconstruction for the STT is under development. The longitudinal position and timing of the energy deposit in an ECAL cell is obtained by comparing the measured timing in the photo-sensors on both ends of the cell. Adjacent cells are grouped into a cluster and the position, timing and energy deposit of each cell is used to obtain the position, timing, total energy deposit and direction of the cluster. The reconstruction for GRAIN lens-based light readout takes as input the two-dimensional images produced by the sub-set of the cameras which have observed each event. The first step of the reconstruction is aimed at identifying the track projections in the images and fitting them. The algorithm is based on the Hough transform [96], multi-Otsu thresholding [97] and DBSCAN [98] clustering to isolate, select and fit the tracks in each image. The second step combines the two-dimensional information to obtain a three-dimensional reconstruction of the event. Assuming the pinhole approximation, pixels associated to a reconstructed track are projected back into 3D space to find a volume compatible with their 2D projections. The final outputs are then the reconstructed tracks, an estimate of the energy deposited from each track based on the amount of collected light, and the 3D vertex candidate. The direct 3D reconstruction for GRAIN with Hadamard masks consists of a custom reconstruction algorithm that obtains a 3D map of the deposited energy. It is based on a probabilistic and combinatorial approach: each hit on the camera pixel is propagated in the detector volume through each mask hole, with an appropriate weight assigned to the segmented volume units (voxels), which measure 1 cm^3. The algorithm is designed to be run on GPUs to provide the required performance. The event analysis in case of mask-based light readout is not so different from the lens-based case. Clustering algorithms and fit procedures are used for the 2D reconstruction of the tracks. The single-pinhole approximation is assumed to combine the signals from different masks and to get the 3D view of the event. A new algorithm to avoid wrong associations of light pixels on different masks is under study. Also, for the masks, the final results are the vertex identification looking at the most probable 3D voxels and the calorimetric measurement.

3.9.5 Common ND analysis files

The final output format used to conduct physics analysis will be a simplified *analysis tree* file whose format has not yet been fully defined, which contains a reduced subset of the full simulation and reconstruction output. The first version of the analysis tree format is based on the ND CAFs, which are produced by applying a parametric response model to the Geant4-level ND simulation for use in long-baseline and other analyses. It is recognized that a more detailed definition of the simplified analysis will be needed in the future, but given the preliminary nature of the ND design greater definition is not possible right now.

3.10 Calibration

Before final pattern recognition can be applied to physics events, the processed hits from calibration samples (subsets of the full data, sometimes with special conditions) are run through specialized pattern recognition and used to derive high-quality calibration constants which are stored in the conditions database for future use. Inputs include processed hit data but also detailed information about the

Table 3.2: Average wall-clock processing time in seconds for reconstruction modules for GArSoft events consisting of just one interaction in the gas. The processor had an HS06 rating of ∼11. An actual spill will contain of order 60 interactions, mostly in the calorimeter, though many tracks will pass through the gas. Reconstruction of more complex events with future software is expected to take more CPU per event than shown below.

Module Label	time/event (sec)
RootInput(read)	0.00
init:EventInit	1.31275e-05
hit:CompressedHitFinder	0.00488308
tpcclusterpass1:TPCHitCluster	0.0091922
vechit:tpcvechitfinder2	0.0103787
patrec:tpcpatrec2	0.0130245
trackpass1:tpctrackfit2	0.014211
vertexpass1:vertexfinder1	0.000851081
tpccluster:tpccathodestitch	0.0269436
track:tpctrackfit2	0.0135842
vertex:vertexfinder1	6.19847e-05
veefinder1:veefinder1	9.96417e-05
sipmhit:SiPMHitFinder	0.00153375
sscalohit:CaloStripSplitter	0.0547754
calocluster:CaloClustering	0.00492599
trkecalassn:TPCECALAssociation	0.000237503
TriggerResults:TriggerResultInserter	2.36397e-05
RootOutput	3.6783e-06
RootOutput(write)	0.214077
Total	**0.369802**

configuration of the calibration system. This step will likely be done many times, especially at the start of the experiment. Calibration samples may be taken quickly and may be very large, for example 500 TB for a full laser calibration of the far detector. They will require occasional fast processing of PB-scale data samples.

The large size of some calibration samples means that fast processing for monitoring and fast application will require peaks in data storage and processing at rates considerably higher than normal data taking.

3.11 Visualization

Both raw and reconstructed data from each of DUNE's detectors must be visualized graphically in multiple ways in order to successfully design, commission and operate the detectors, and to extract meaningful physics results. While reconstruction is automated and hand-scanning is no longer used to extract physics results, event displays are critical for a large number of necessary steps:

1. **Detector Design Development** While designing detectors, it is important to visualize interactions in the design in order to ascertain which interactions are easy to identify and measure and which ones are difficult, in order to refine the design.

2. **Optimizing reconstruction algorithms** Comparisons of true particle trajectories and energies with reconstructed versions thereof in simulated interactions provides guidance for improving reconstruction algorithms. Quantitative metrics such as completeness and purity can be optimized, but it is easier to do so when viewing an event display that shows which pieces of which interactions have failed to reconstruct well.

3. **Operating prototypes and learning from them** Data from prototypes, such as the cold boxes, the ProtoDUNEs, and ICEBERG R&D cryostat and electronics (ICEBERG) must be analyzed quickly to ascertain what the noise level is and to search for the presence of signals. This is part of an iterative experimental process to commission the prototypes by searching for noise sources, checking HV and cryogenic performance, drift medium purity, etc. Dead channels or other artifacts such as long-range induction effects are plainly visible on raw data displays. Various artifacts may be addressable with software, such as correcting for front-end amplifier artifacts, correlated noise, and ADC issues. These are all discovered first with event visualization.

4. **Interpreting calibration data** Laser calibrations, neutron source data and radioactive source data must be inspected first before use, as unanticipated artifacts may confound the intended calibration use.

5. **Providing educational and public outreach materials** Introducing new students to DUNE's physics program involves showing them what interactions look like in each of our detectors. Materials for DUNE's public-facing web sites and press releases also require high-quality data visualizations.

Each of these uses places different requirements on the visualization software. Furthermore, the event display programs must be responsive and easy to use, though specialist functions may require additional configuration.

3.11.1 Two-Dimensional Event Displays

The data from the FD1-HD and FD2-VD modules are inherently two dimensional, with ADC values read out per wire (or strip) for each sample time. Inspecting the raw waveforms side by side in a grayscale or colored map for each of the three views is critical for understanding the features of the raw data. LArSoft provides such a display, optimized for X-Window connections. Users can zoom in on rectangular subsets of the data and clicking on the image brings up a one-dimensional waveform plot for the corresponding channel. An example of such a display for a ProtoDUNE-SP event with cosmic rays is given in Figure 3.6. An example of a two-dimensional display of reconstructed tracks, vertices and showers is given in Figure 3.8.

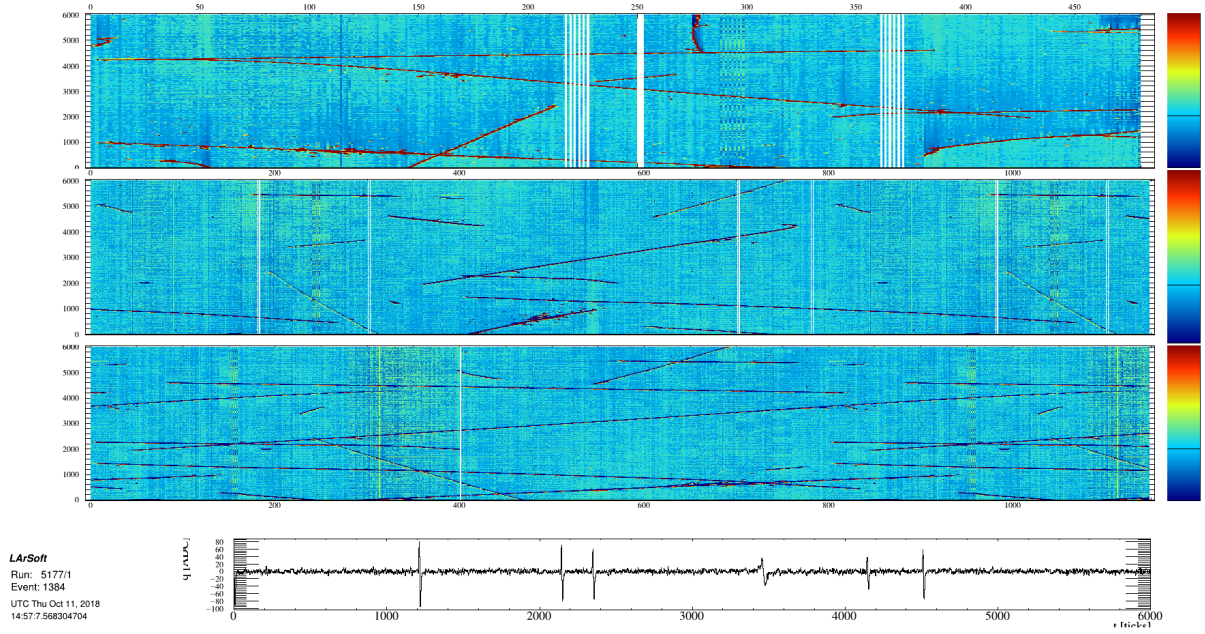


Figure 3.6: Example raw data display produced with LArSoft. One APA's worth of data are shown for a ProtoDUNE-SP trigger record collected in October 2018. The top panel shows the collection-plane data, the next panel V-plane data, and the bottom colored panel shows U-plane data. Below those three is a single channel's waveform. The colored panels show the pedestal-subtracted ADC values for each channel on the horizontal axes with sample time in 500 ns ticks on the vertical axes. The time between samples is 500 ns. White vertical lines in the collection plane correspond to channels that have been flagged as noisy or dead.

The raw data display provided by LArSoft is best run interactively. The user can step through the events in a file, forwards or backwards, or skip to a desired event number. This event display is not built for non-interactive use. There are use cases in which large numbers of raw event displays need to be produced in a batch. For this purpose, dedicated modules have been written in DUNE's LArSoft-based software that provide two-dimensional displays of raw data and data that have been processed through each step of data preparation. Examples are shown in Figure 3.7 after four stages of data processing. While there is definitely room for development of features and improved ease-of-use, this is currently the best resource for raw data viewing.

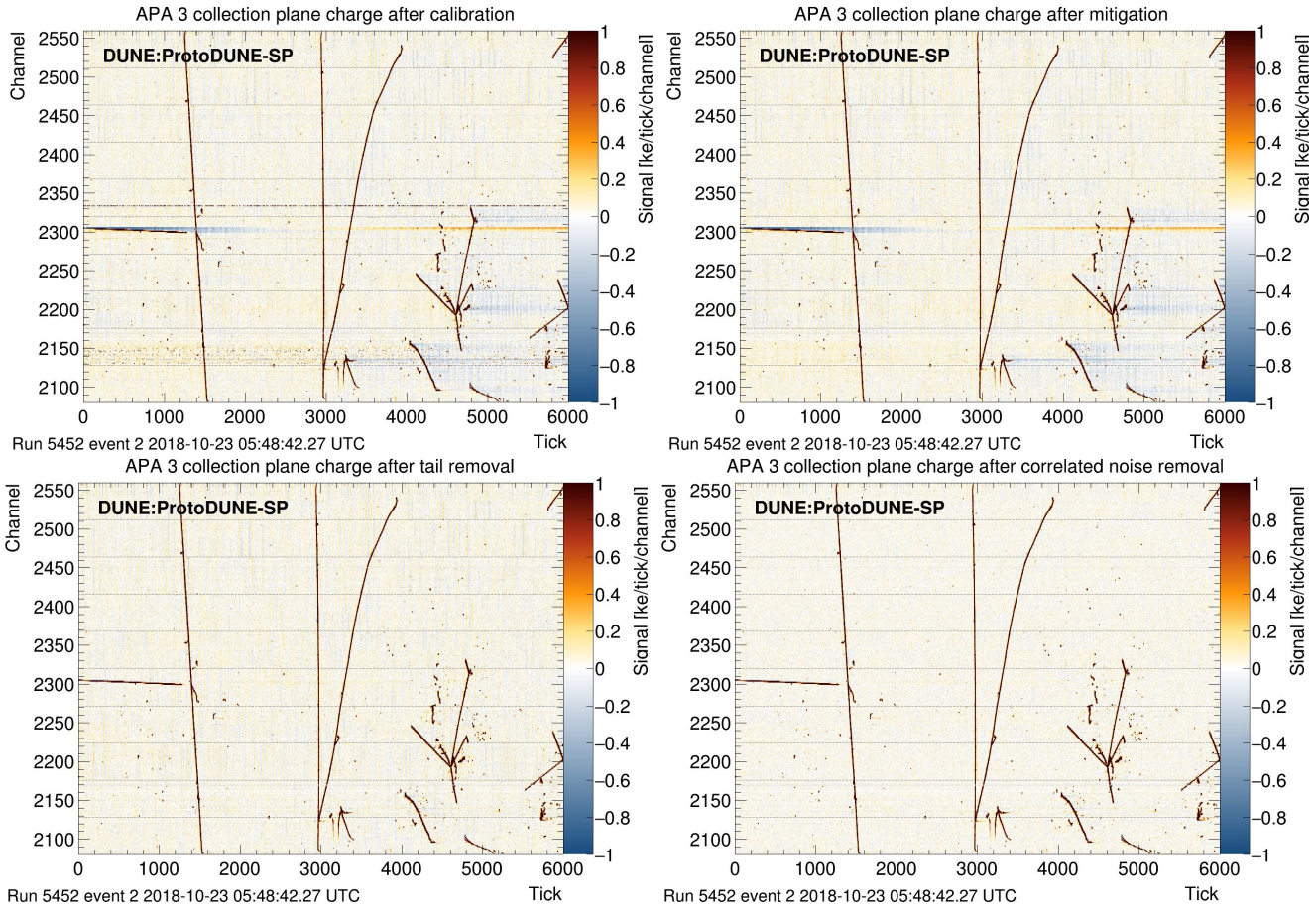


Figure 3.7: Example batch-mode event displays for a collection plane showing background reduction in successive stages of data processing. For each plot, the horizontal axis is the readout sample time in 500 ns ticks, and the vertical axis is the channel number. The color scale represents the charge for each channel averaged over five ticks with the range chosen to make the noise visible. Signals from charged tracks appear mostly in black and are off scale, well above the noise level. From Ref. [17].

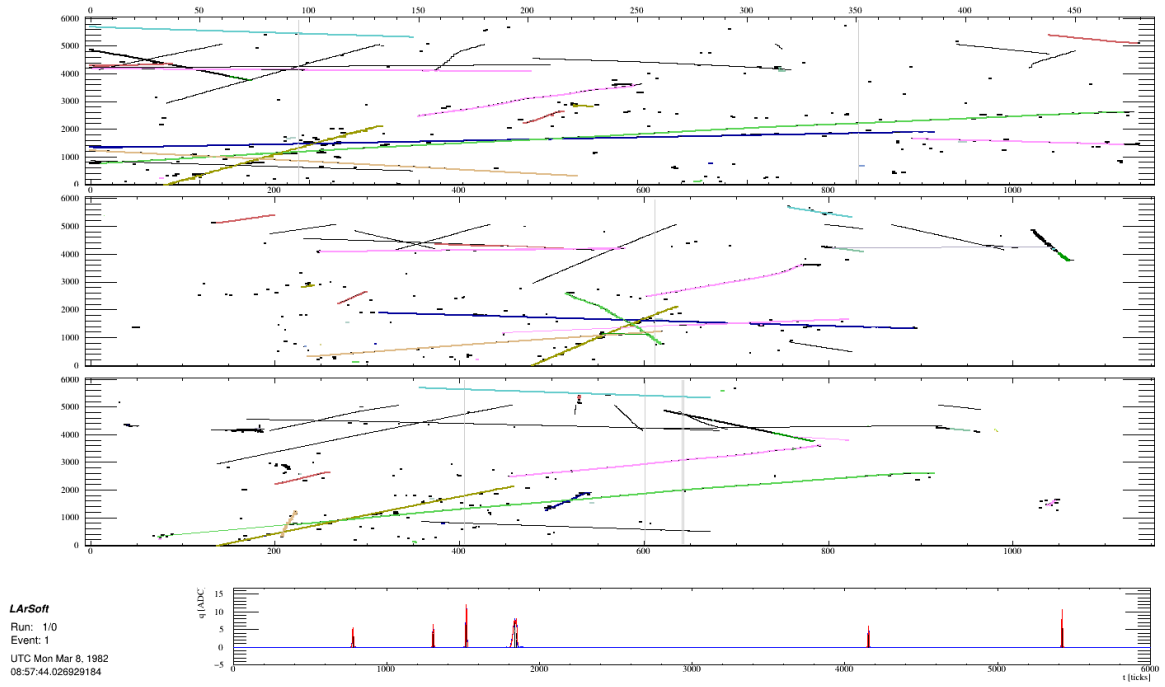


Figure 3.8: Example reconstructed data display produced with LArSoft. One APA's worth of data are shown for a simulated ProtoDUNE-SP trigger record. The top panel shows the collection-plane data, the next panel V-plane data, and the bottom colored panel shows U-plane data. Below those three is a panel showing a single channel's deconvolved waveform and fitted hits. Reconstructed hits, space points and tracks are displayed in the top three panels.

3.11.2 Three-Dimensional Event Displays

LArSoft also provides a three-dimensional event display, showing the same reconstructed objects. An example is given in Figure 3.9. Only reconstructed objects are available in three dimensions as the data in the ProtoDUNEs and the Far Detector modules are two-dimensional.

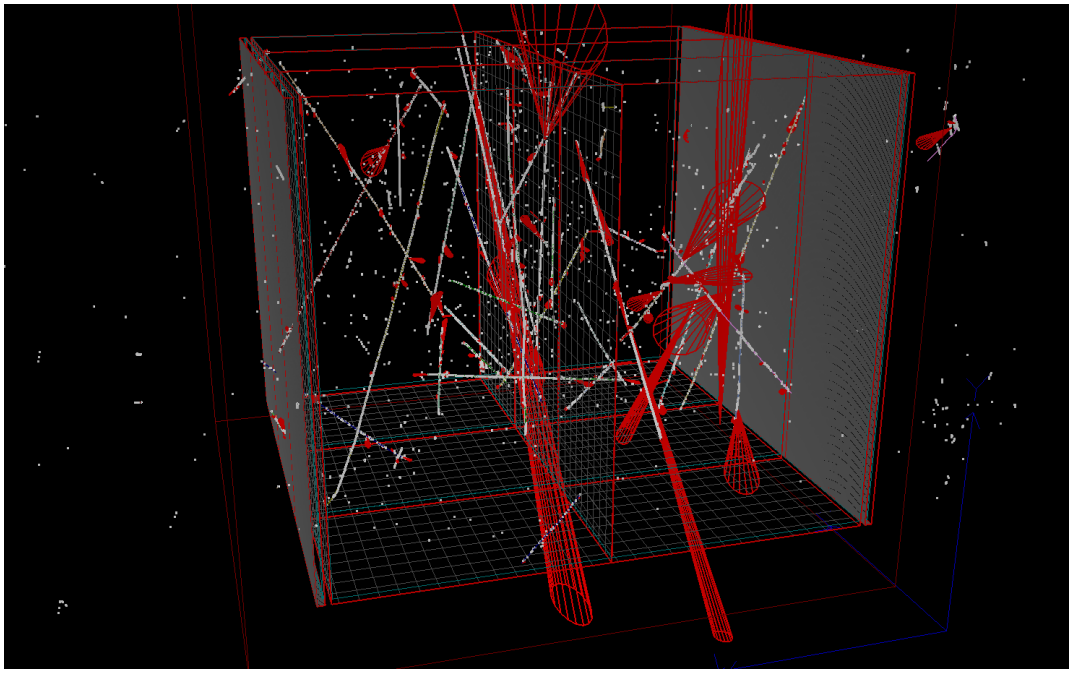


Figure 3.9: Example three-dimensional reconstructed data display produced with LArSoft, showing data from a ProtoDUNE-SP trigger record, for all anode plane assemblies. Reconstructed space points are shown in gray while shower cones are drawn in red. The display can pan, zoom, and rotate under user control. The LArSoft 3D event viewer can also produce animated gifs.

3.11.3 Web-Based Event Displays

In addition to the LArSoft-based event displays, DUNE has multiple web-based event displays. One, called the Bee event display, is made for displaying ProtoDUNE-SP events in a web browser. The server is hosted at Brookhaven National Laboratory and contains a short list of typical ProtoDUNE-SP events, one of which is shown in Figure 3.10. No software needs to be installed on the user's computer, and no data files need to be copied. The user only needs a modern web browser and the experience is enhanced with a capable graphics card.

Another web-based event display is called WebEVD. It runs as a LArSoft module, and it sets up a local web server owned by the user process on an interactive computer. The user then must ssh in to the interactive computer running the web server, with a specified port forwarded. The user then connects to that port with a local web browser and interacts with the event display with it. This user-owned web server which is only visible via authenticated ssh has the advantage that it provides the user with full control over which data are to be displayed, as the web server is not centrally managed. An example display of two-dimensional deconvolved waveforms for a simulated ν_eCC event in a FD workspace geometry run is shown in Figure 3.11. An example three-dimensional event display from a ProtoDUNE-SP data run produce using WebEVD is shown in Figure 3.12.

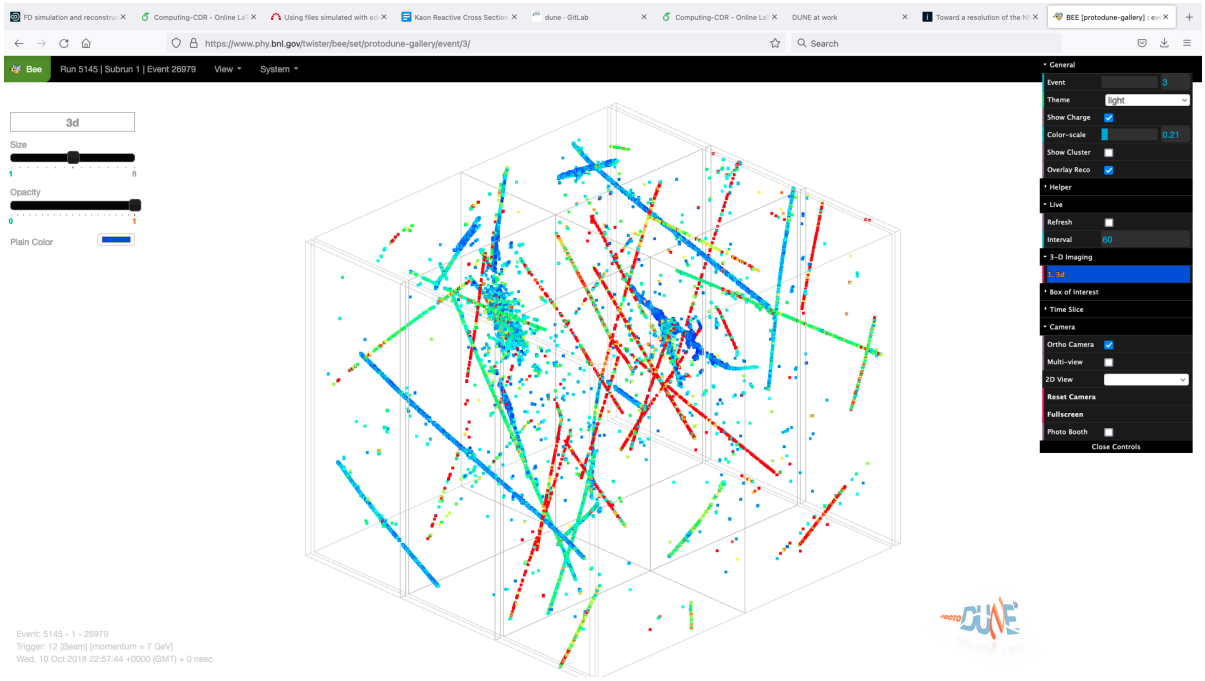


Figure 3.10: Example three-dimensional reconstructed data display for a ProtoDUNE-SP data trigger record produced with the web-based Bee event display. Users interact with the data needing only a modern web browser on their computer. They can pan, zoom, rotate, and change display colors, point sizes, and transparency.

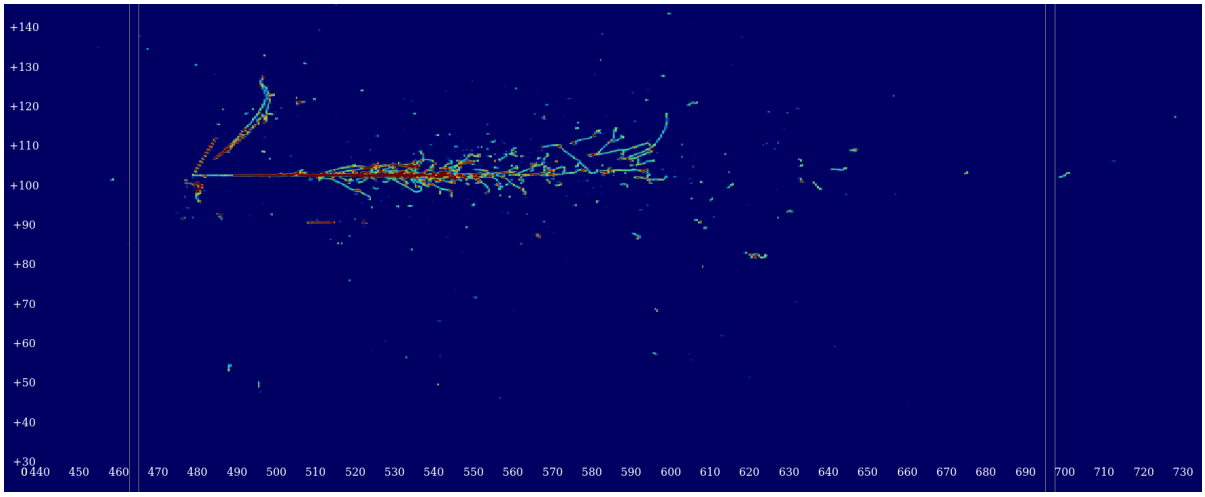


Figure 3.11: A simulated ν_eCC interaction in a far detector workspace geometry run. The display is of charge reconstructed on collection wires (recob::Wire). The horizontal axis is the z-axis (beam direction) in the detector. The vertical axis corresponds to drift time (x position in the detector). Both axes are labeled in cm. Rendered with WebEVD.

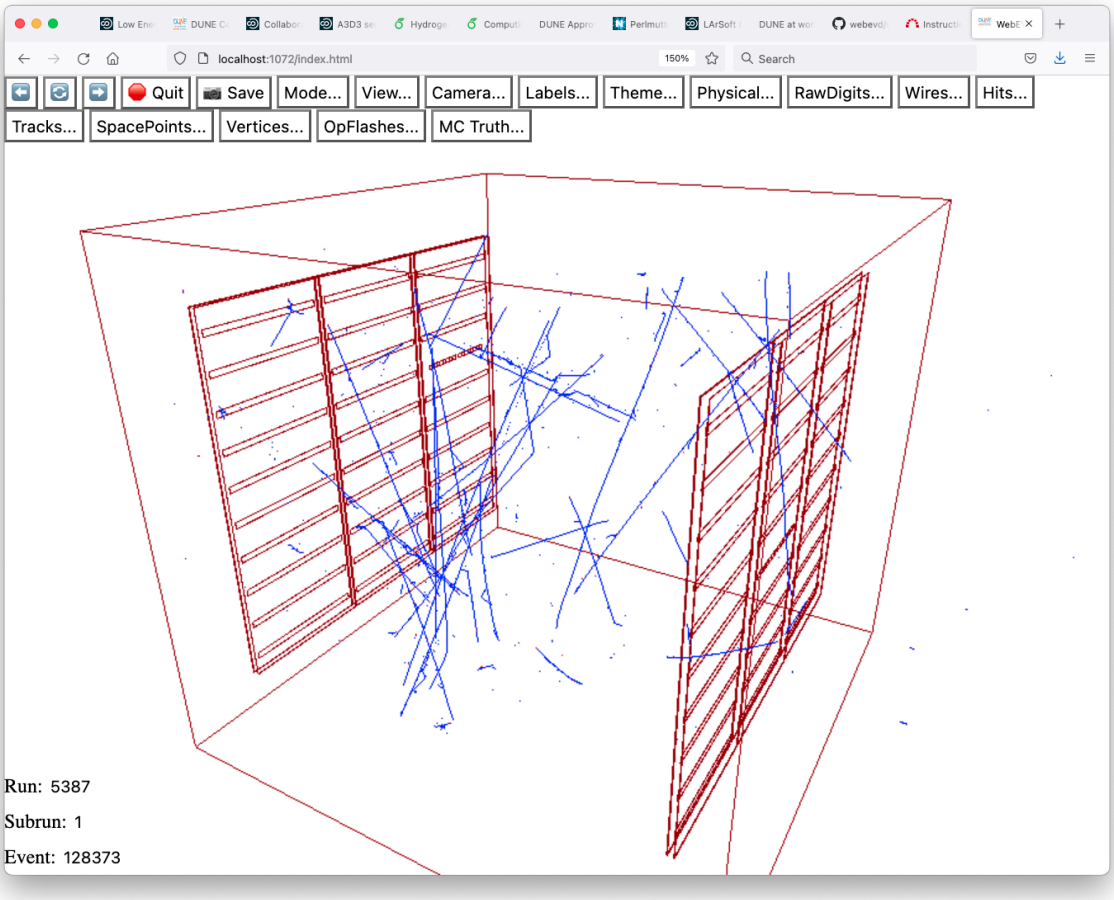


Figure 3.12: Example three-dimensional reconstructed data display for a ProtoDUNE-SP data trigger record, rendered with WebEVD. Reconstructed tracks and space points are shown in blue, and detector elements are shown in red.

3.11.4 Near Detector Event Displays

An example of a reconstructed event in ND-GAr is shown in Figure 3.13 rendered with the ROOT-based event display in GArSoft. The same event is shown in Figure 3.14, rendered with the TEve-based event display in GArSoft.

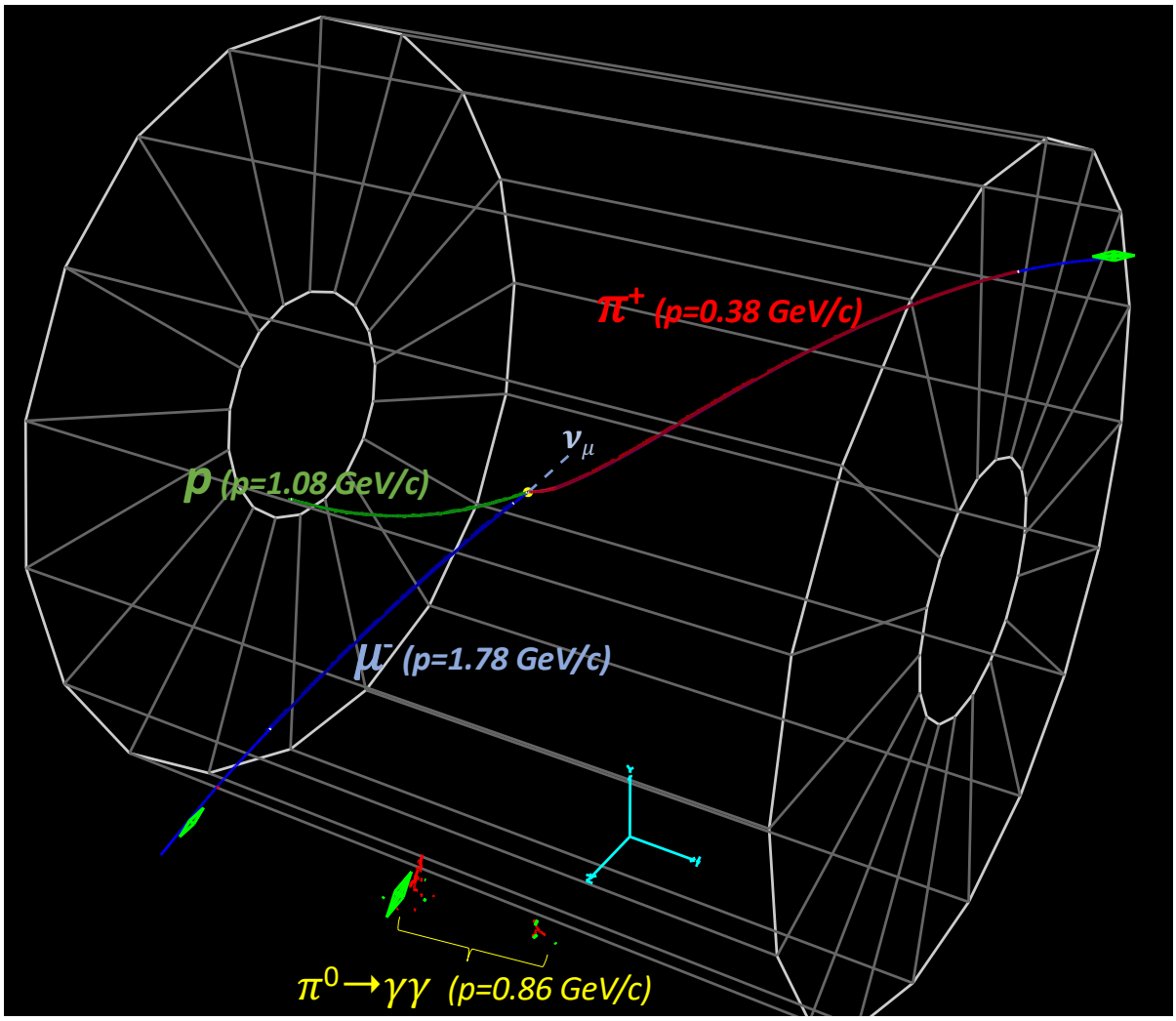


Figure 3.13: Example reconstructed ν_μCC event produced with GArSoft, using an ALICE-inspired detector geometry. The reconstructed vertex is indicated with a yellow dot, and tracks are shown in green, blue, and red. Reconstructed calorimeter clusters are shown with green tetrahedra. The annotations giving the MC particle identities were added by hand.

The third component of the near detector, System for on-Axis Neutrino Detection (SAND), consists of a straw-tube tracker and a calorimeter inside a solenoidal magnet. A small cryostat containing liquid argon, called GRAIN, is to be installed upstream of the straw-tube tracker inside the electromagnetic calorimeter (ECAL). Design and performance studies are underway. Basic event visualization tools have been developed to aid in the development of the reconstruction algorithms. An example display of simulated and reconstructed particles in SAND is shown in Figure 3.15.

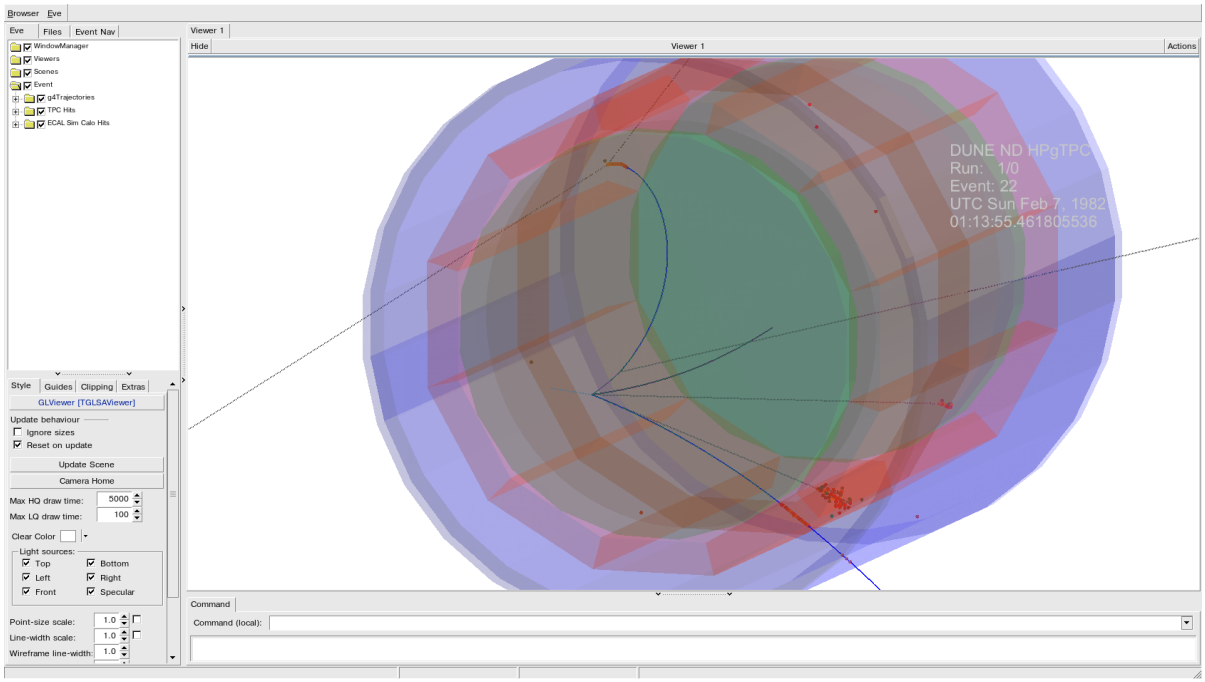


Figure 3.14: The same ND-GAr event as shown in Figure 3.13, rendered here with GArSoft's TEve-based event display.

3.12 Analysis of Reduced Data Samples

3.12.1 Analysis Sample Production

At some point in the processing, specialized datasets and streams based on trigger type, reconstructed event information, and intended use need to be defined and produced.

The interaction data, which are in the output format supplied by the full reconstruction are then reduced and reconfigured into analysis formats for use by users. In the long run, this processing will be done as coherent production steps but may also be done by small groups while the data formats and procedures are being developed.

CAF (Common Analysis Files), the standard analysis-level file, is an example of a reduced data sample. DUNE's primary analysis framework for oscillation physics is currently the CAFAna framework, originally developed for the NOvA experiment [99, 100]. The High Level Analysis at a Neutrino Detector (HighLAND) framework used by T2K is also used for some analyses.

This phase of processing is I/O limited and can put considerable strain on storage and network resources. The CAF framework and calibration drive the need to put most of the reconstructed data and simulation on disk at distributed sites as described in Part III. A typical ProtoDUNE-SP data or simulation pass produces 10,000–100,000 2-8 GB reconstructed files and then produces much smaller tuple outputs for analysis. These reduction jobs stream using xrootd and are I/O bound. Preliminary monitoring studies indicate that average input rates of 5-30 MB/sec per process can be achieved when the source Rucio Storage Element (RSE) and processing are co-located (as is possible at the largest facilities) and the data can be streamed via the Local Area Network (LAN) within the site, with aggregate rates of up to

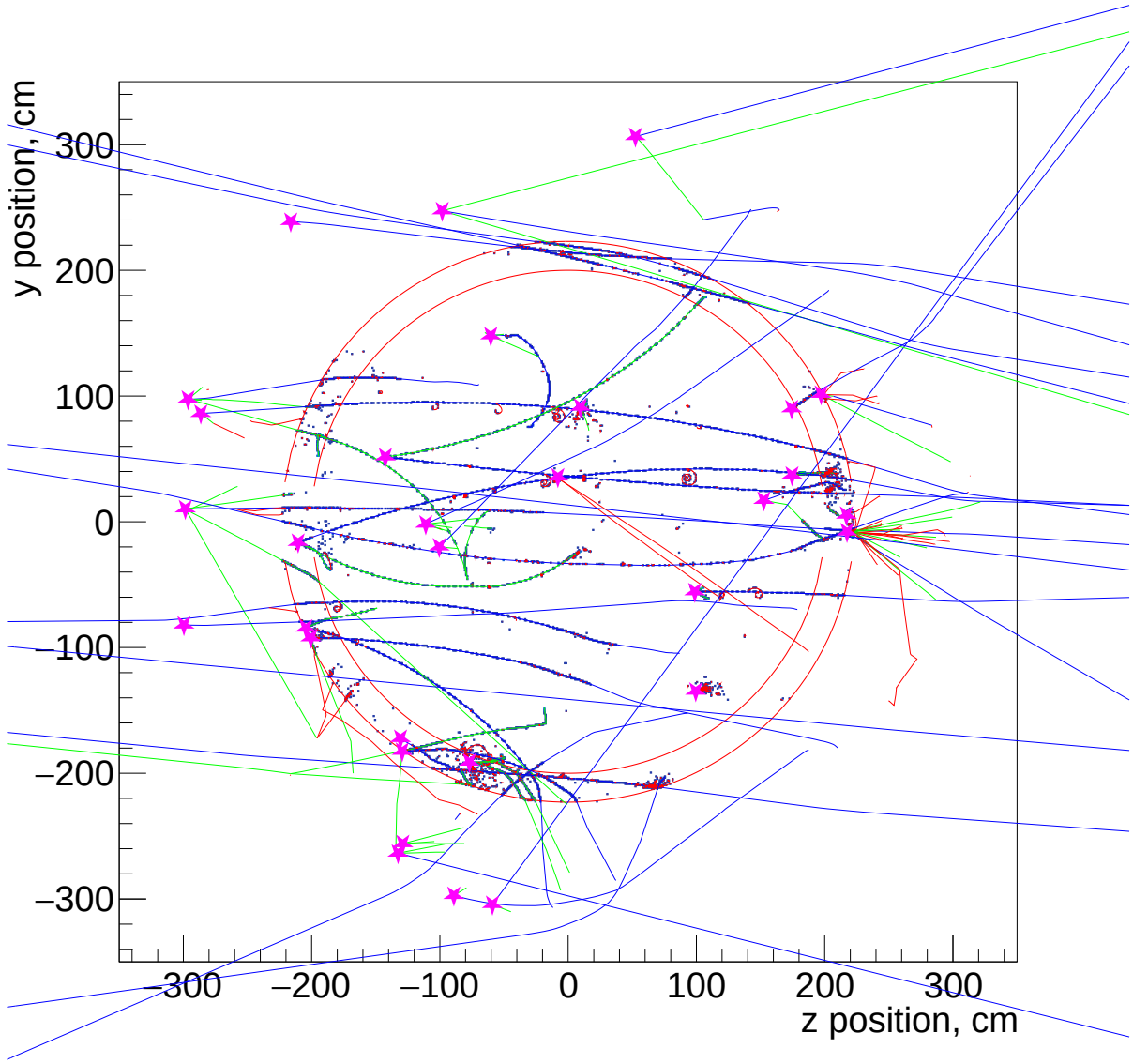


Figure 3.15: A rendering of a simulated full spill of neutrino interactions in the SAND detector. Magenta stars indicate true neutrino interaction vertices, blue curves indicate true simulated muon tracks, and thick green curves show proton tracks Thin red lines indicate electrons and photons. The red circles show the outline of the ECAL. Blue dots indicate calorimeter hits.

several GB/sec observed when many batch processes run at a given site. We are currently mining data access records to measure rates and reliability as a function of source and sink.

3.12.2 Reduced Analysis Samples

Reduced data analysis samples are the samples users see and analyze when they are not developing new reconstruction or calibration algorithms. Analysis samples should be useful and as small as possible. Analysis codes should not need to read from the central databases but may need to access small local replicas.

Data analysis may be done on local clusters, on collaboration grid facilities using shared data samples, or in dedicated analysis facilities that offer advanced access methods such as the Columnar Analysis framework Columnar Object Framework For Effective Analysis (COFFEA) [101]. This phase of processing is typically very I/O bound and requires fast access to smaller data samples.

An important question is what auxiliary information is needed and how it will be delivered. In particular, geometry and run-quality information are often needed in final analysis stages while direct access to detailed electronics calibrations is less often needed.

3.12.3 Current Practices

A survey of data analysis users was done in February 2022. Analyzers were contacted through the physics groups and analysis Slack channels. We received 29 responses with users from institutions in eight countries dominated by the US(14) and UK(8). Responses were spread across multiple physics efforts: ProtoDUNE (13), Near Detector (5), Low Energy FD (9), High Energy FD (7) and Calibration (5).

Figure 3.16 summarizes the results. LArSoft is used by most users (presumably for small scale tests or tuple creation). Users then rely on their own ROOT-based C++ or Python code for the final analysis steps. A small number of users use shared analysis frameworks such as CAFAna, Nuisance [102] and HighLAND. Jupyter notebooks are becoming popular.

Users analyze their reduced samples (which range in size between 5 GB and 10 TB) on the Fermilab interactive machines, via the grid batch systems and on local clusters and desktops.

Although the physics groups tend to use the same technologies to analyze their data, the particular data content of their tuples differs. Many users generate their own formats (at considerable expense in time and effort). Physics groups are beginning to move to common shared production as the content choices stabilize. A single common format with common content is unlikely to emerge given the wide range of content different use cases (calibration, far detector low-energy signal, near detector, ProtoDUNE studies) that are already present.

The current goal for computing infrastructure is to support a wide range of formats and analysis strategies while encouraging shared, well-documented, cataloged samples.

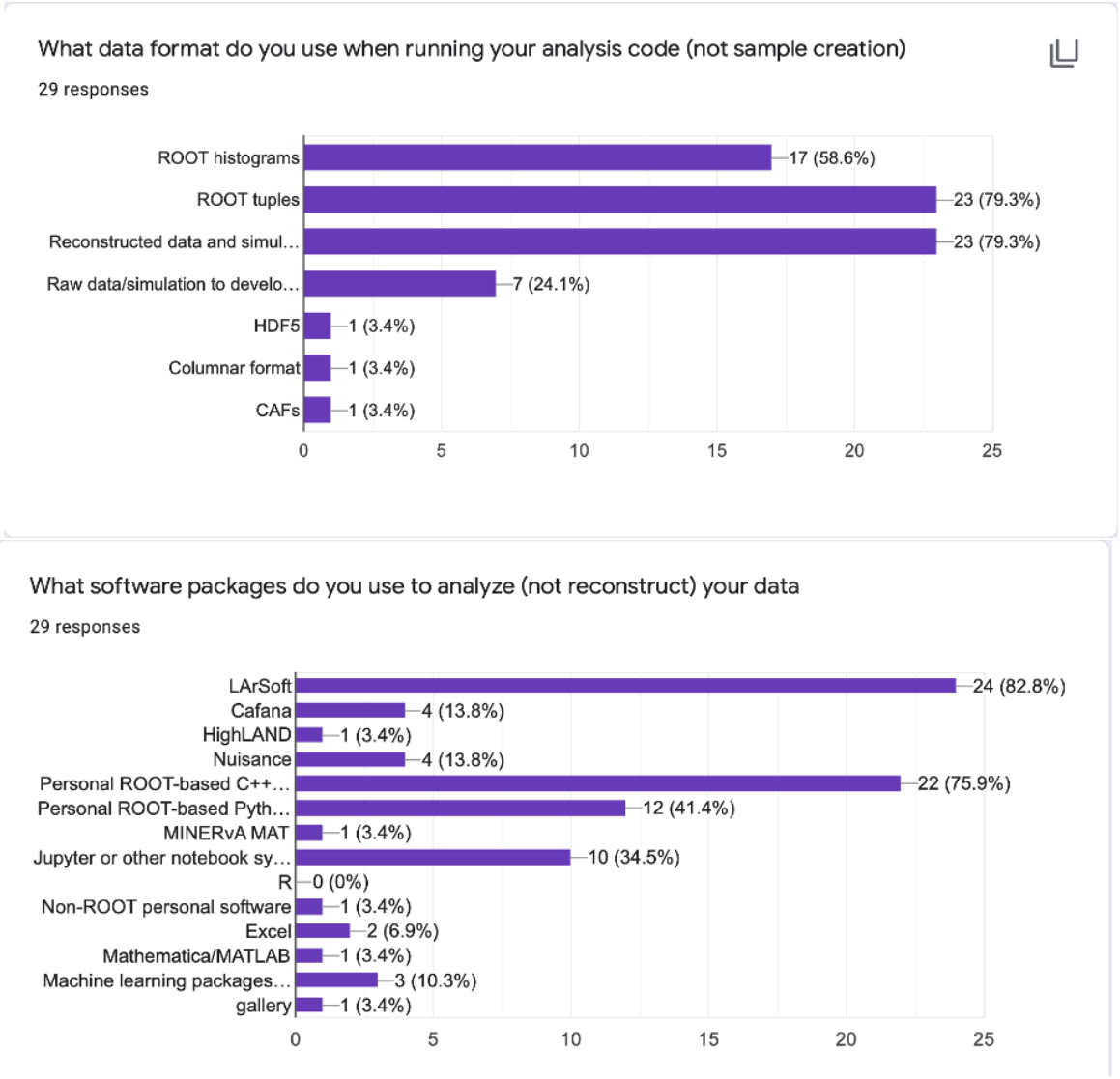


Figure 3.16: User responses to the analysis survey. Panels show the data formats used, the analysis frameworks used, the locations where the jobs are run, and the locations where code is kept.

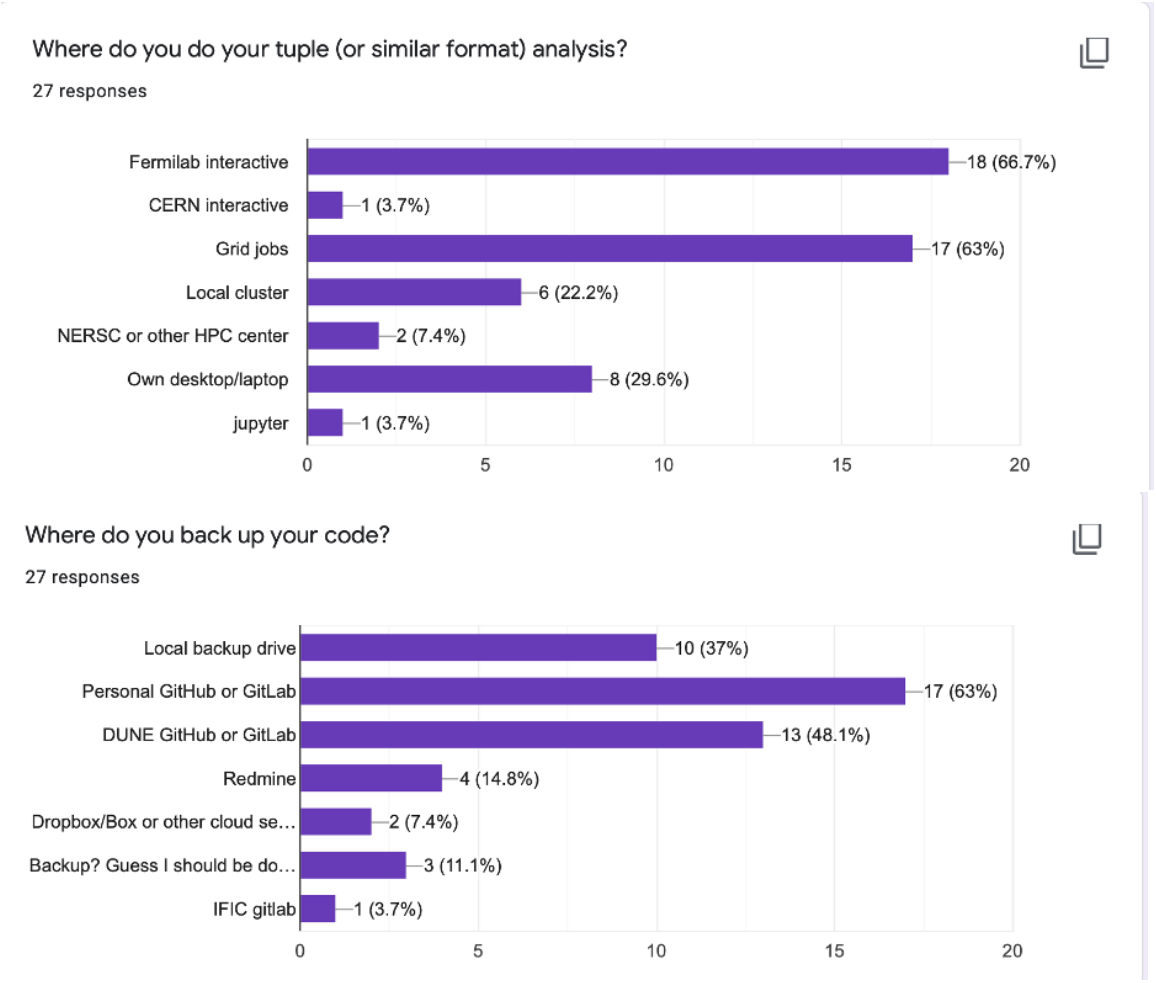


Figure 3.17: User responses to the analysis survey (continued). Panels show the data formats used, the analysis frameworks used, the locations where the jobs are run, and the locations where code is kept.

3.13 Machine Learning Training and Implementation

Neutrino science, and HEP more generally, has been increasingly relying on Machine Learning (ML) methods for track and shower reconstruction, particle ID, event selection and classification, and more. Many simulation and reconstruction algorithms in the DUNE software stack can profitably use ML algorithms, and some already do. While the inference stages of these algorithms can run quickly on each event, the training stages require significant resources. In some cases, the inference stages run much more efficiently on specialized hardware. Some non-ML algorithms also benefit from specialized hardware. Multiple groups are working on implementing some of DUNE's algorithms on GPUs. A notable example is reference [80] in which ProtoDUNE reconstruction workflows on simulated far detector neutrino interactions are implemented via the Services for Optimized Network Inference on Coprocessors (SONIC) framework developed at Fermilab. This hybrid framework enables the use of remote GPUs as call-outs within the normal *art*/LArSoft workflow. In particular, the electromagnetic shower algorithm, `EmTrackMichelId`, runs a convolutional neural network (CNN) inference step. When run on a CPU, it is the algorithm that takes the most CPU time, as seen in Table 3.1. To speed it up, this algorithm was run on commercial (Google Cloud) remote GPUs. Patches of detector data were sent to the remote processors over TCP/IP and inference results were returned. In the initial test, event processing, which is dominated by that step, was sped up by a factor of 17. Similar approaches are very promising for the future.

Other ML algorithms export data from *art*/LArSoft jobs in formats that can be read by external ML packages, and the training is done by a separate program. Algorithms for reconstructing vertices, ν_μCC and ν_eCC events are written as CNNs and trained in this way. The CNN parameters from the training step are stored in a file that is then loaded by an *art*/LArSoft job that then calls TensorFlow to run the CNN inference during the reconstruction phase. An experimental infill algorithm that interpolates data for dead or broken channels also operates in this way.

3.14 Parameter Estimation

Once the experimental observations are made (i.e. the number of and energies of different charged current interactions in the near and far detectors), these observations must be linked back to the underlying physics models explored through a formal process of parameter estimation. In its most basic form, parameter estimation takes the form of a "fitting" procedure where by *observed* spectra are fit to *prediction* spectra based upon PMNS or other oscillation models. However, the statistical nature of the observations combined with the experimental uncertainties intrinsic to the measurements, and the natural degeneracies that are present in different neutrino interaction and oscillation models, leads to a statistically ill defined backwards transformation problem (i.e. the mapping from the observation space to the model space is not a single valued transform). In particular the formal process of parameter estimation relies on a number of statistical techniques that determine the most likely regions of parameter space and their neighborhoods which would result in the observation that was made by the detectors.

In the case of the neutrino oscillation measurements, the combination of the low statistics of many of the observations (i.e. the ν_e-CC and $\bar{\nu}_e$-CC events in the far detector) along with the periodic form of the transition probabilities of the oscillation models, lead to a violation of Wick's theorem and prevents the use of Gaussian statistical approximations for the interpretation of χ^2 distributions and other frequentist statistical techniques. Instead, neutrino oscillation parameter estimations from experiments

such as NOvA [103] have relied on Feldman-Cousins based techniques which profile over nuisance parameters, and are able to obtain proper statistical coverage of multi-dimensional parameter spaces. Parameter estimates from other experiments such as T2K have used Bayesian techniques combined with Hamiltonian Monte Carlo integration techniques [104] to probe similar parameter spaces.

The issues with these techniques, which will become even more challenging for DUNE, is that their dimensional scaling runs with both the dimensionality of the fundamental parameters of interest being estimated, and also with the dimensionality of the auxiliary or nuisance parameters that are included in the computations. In the case of the modern results from NOvA, pairwise estimation of PMNS parameters (i.e. $\sin^2\theta_{23}$ and Δm^2_{32} or $\sin^2\theta_{23}$ and δ_{CP}) are extracted against a field of $\mathcal{O}(50)$ nuisance parameters, which are profiled over. These computations require repeated fitting/minimization computations against pseudo-experiment distributions that are "thrown" using Monte Carlo techniques to provide adequate coverage of the underlying model and observation spaces (i.e. both the neutrino interaction model spaces and the smearing and intrinsic resolutions of the detector apparatus spaces need to be varied to provided realistic coverage of the actual observation.) In the case of experiments like NOvA, which are exploring the parameter spaces at $2-3\ \sigma$ significance levels at $< 5\%$ induced computational uncertainty this can result in the need to generate and fit in excess of 9k pseudo-experiments per point in the primary parameter space near the 3σ confidence boundaries.

Taken as a whole, parameter estimation has proven to be a significant portion of the total computational budget for modern neutrino experiments. In this respect the parameter estimation portion of NOvA results required 73 million core hours of computation in 2018, 78.5M in 2019, 127M in 2020 and most recently the 2022 neutral current results required 55 million core hours (although this result was only report to the 2σ significance level). In all of these cases, the experiment was able to obtain these resources from the HPC facilities at NERSC and were able to transition their codes to run first on Edison, then Cori-Haswell, Cori-KNL, and most recently the Perlmutter system.

Techniques and infrastructure for executing these computations using high performance computing platforms (HPCs) have been developed by the ASCR-supported SciDAC program, in conjunction with NOvA and SBN. These codes have been used by NOvA and SBN to obtain their published results starting in 2018 and have become the *de facto* standard for parameter estimation in these experiments. It is expected that DUNE will adopt this approach, and build upon the performance improvements that have been so far pioneered by these projects.

3.15 Summary: Characteristics of Large-Scale Processing Tasks

A large number of complicated tasks have been listed above. Table 3.3 summarizes the computing characteristics of the most prominent ones per MB of data. In combination with data volumes for any given task, these estimates can be used to predict processing needs as described in Chapter 6 and to optimize data and job placement as described in Chapters 11 and 13.

- Simulation requires flux files (and possibly overlay libraries) as input but produces much larger outputs. The impact on networks is negligible but the memory needs are substantial.
- Reconstruction of data is assumed to be done by streaming the input raw data to most likely a computing site that is separate from the raw data storage. The processing time/MB is large enough that the impact of streaming raw data on networks is minimal.

- Ntuple creation and calibration require multiple passes over the reconstructed samples. Either local data stores or streaming from "nearby" sources is optimal as network speeds become important. Section 7.2 describes studies of job throughput as a function of disk and CPU locality.
- Parameter estimation has been done using the high-performance computing (HPC) resources at NERSC, and it is anticipated that additional HPC sites will be available in the future. In this case, a small amount of input data is processed simultaneously with systematic variations across a large number of cores.

Table 3.3: Summary of resources needed per file for compute intensive tasks. The total CPU needed for a task can be determined from the total size of the sample and these numbers. Here the time is in seconds on a processor with rating around 11 HS06.

Use case	memory	input file size	output file size	CPU time	input	cores/job
units	GB	MB	MB	sec	MB/s	
Simulation+reco	6	100	2000	27000	0.00	1
data reco	4	8000	4000	60000	0.13	1
tuple creation	3	4000	1	100	40.00	1
calibration	3	4000	1	100	40.00	1
Parameter estimation	1	400	1	600000	0.00	68000

Chapter 4

Frameworks

The Deep Underground Neutrino Experiment (DUNE) presents a unique challenge for data analysis and data processing software frameworks. DUNE has an ambitious physics program that spans numerous physics topics including precision neutrino oscillation measurements, searches for proton decay, sensitivity to nearby supernova explosions, and more. Moreover, these physics interactions occur at vastly different timescales, from nanoseconds to 100s of seconds, and software frameworks that are fully capable of adapting to these varied timescales efficiently will be very important. Furthermore, accomplishing timely results will require that data from the DUNE far and near detectors be efficiently processed using modern computing techniques. Given evolving computing architectures, traditional HEP software must be adapted to run on new generations of advanced and accelerated computing resources such as high-performance computing (HPC), Graphical Processing Units (GPUs) and other novel computing architectures.

Development of this type of modern computing pipeline and analysis structure will require reoptimization and extensions of the existing software frameworks used by DUNE, and shared to a large extent by the greater neutrino community to which DUNE and its computing ecosystem belong. In this Chapter, we provide a brief overview of our existing code base followed by a description of our process for designing the optimal frameworks for data processing and analysis that will be needed by the time DUNE begins operations.

4.1 Defining a Framework

4.1.1 The Data Atom

Modern HEP experiments generate large volumes of primary detector data, as well as auxiliary and ancillary data from secondary systems. Along with detector data, vast amounts of simulation which mimic these detectors and their responses provide estimates and predictions for the different physics processes that may be observed in the experiments. The management, flow, transformations, analysis and interaction of these data and their algorithms require a well-defined framework for governing and sequencing these tasks. In particular, traditional HEP analyses data processing and analysis frameworks are designed to provide "event loops" which allow experimenters to step through the data and simulation

in a manner that can break up the information into discrete chunks which represent physical processes. In this chapter, these discrete chunks are referred to as "data atoms" and contain a fundamental assumption that they can be treated as independent units of work. In a neutrino experiment, such as photon detector (PD) or the DUNE near detector, this “atom" of data may well contain several neutrino or hadron interactions of direct interest in addition to 50-100 cosmic rays or background muons.

Historically in HEP experiments, the data atoms have been associated with well defined external time structures which govern the data production mechanism and/or collection process. For collider experiments these are often associated with a particle accelerator beam or bunch crossing, while for fixed target experiments they have often been associated with pulsed beam extraction to target stations (often referred to as a "spill"), or substructure within the beam spills defined by the data acquisition and triggering systems (i.e. the data atoms are the individual triggers).

In the case of the DUNE physics mission and the resultant system designed to accomplish that mission, the definition of these data atoms is not as simple or as static. In particular in the DUNE vernacular, the data atom does not map in a one-to-one manner with traditional “event” terminology that is used in collider physics. Rather the data atom represents a spatial and temporal extent of interest and can vary depending on the contextual nature of where within the processing chain we are focused. This is not a new concept within the neutrino community, and predecessors of DUNE have used varying terminology to refer to the subsetting of triggered and free running readouts, whether the NOvA “slice”, the MINOS “snarl” or other such ill-named terms for the simple, contextual subsetting of time windows into variable collections of data. In particular, this approach is needed because while neutrino beam interactions have natural timescales of nanoseconds, the timescale of supernova neutrino bursts, proton decay searches, and long-lived, BSM particle searches span up to 100s of seconds. The nature of these varying times scales within the DUNE physics program results in data atoms which vary in size and structure by orders of magnitude, and may be spread across numerous organizational units, data structures and even permanent storage records such as files. Furthermore, apart from the neutrino beam interaction case, which should have a limited number of predetermined readout times, the data production mechanism is not provided by a deterministic external source and is random in nature. DUNE data atoms could represent a) an arbitrary time window representing a time period of interest b) a time window representing the drift of ionization across the detector volume c) a spatially and temporally isolated region of the detector readout d) a sub-region of interest within a spatially/temporally distinct portion of the detector or e) other segmentations of the DUNE detector readouts corresponding to potential physical signatures with time structures defined by the physical process. Having a framework that can adapt to varying time structures, multiple input files, and data structures is a very important feature for DUNE to accomplish its vast physics goals.

4.1.2 Key Software Framework Concepts

A software framework is a software structure and engine that can ingest and apply transformations and filters to the individual data atoms regardless of their underlying nature and related time structure. The framework has a responsibility for controlling and managing the I/O associated with ingesting the data atoms, for sequencing and scheduling the algorithms and transforms that run on the data, and for providing logging, provenance and other bookkeeping tasks associated with describing and enumerating how the data was handled and new information derived.

To these ends, the framework provides mechanisms (such as "plug-in" structures) for experimenters to

develop modular algorithms which can be included or excluded from a given analysis chain and be run either serially or in parallel depending on the data context. The framework also provides mechanisms for data to be passed between different algorithmic portions of the data processing and analysis chain while maintaining the data's coherence and integrity from the standpoint of the computing platform. In some framework implementations this data passing mechanism is provided by a *data store* which includes controlled forms of data access and locking. Frameworks which use these types of methods for data passing need to then additionally manage memory footprints of the stores to ensure that they can operate within the technological memory capacities provided by the platforms on which the codes run. Other frameworks can provide less rigid data pipelines for information passing which can be advantageous when dealing with large data streams and which can better accommodate ingress and egress data paths with respect to the bulk memory footprints, but which may not be compatible with certain types of parallel processing and parallel algorithms, or with algorithm scheduling systems.

The other key feature that modern frameworks provide, is the ability to configure and re-configure the content and flow of the execution modules without the need to rewrite the underlying algorithms. This aspect of a framework as a configuration driven entity is key to their flexibility and their ability to satisfy the needs of the many different and varied analyses that are proposed for DUNE. Without a robust configurable framework, different data processing tasks and analysis chains could be spread across a wide variety of codes, likely with significant duplication, which could severely hamper the collaboration's ability to understand the derivation of results and ensure their integrity during scientific reviews. A robust configuration-driven framework system allows users to make changes to the values of individual parameters in order to investigate the data or simulation's response to those changes, and it allows for reordering, restructuring and inclusion/removal of algorithmic blocks without the need to edit or recompile the core physics codes. This in turn promotes code stability, reuse, and facilitates modern code design and debugging principles.

Another key concept that defines modern data processing frameworks is their construction and organization as advanced state machines. In contrast to older, linear execution techniques, this design allows for the execution of more complicated workflow topologies, parallel analysis chain topologies, and dynamic execution models. This state machine organization and behavior also allows the frameworks to emulate better the behavior of other systems that are used throughout HEP, such as those encountered in the data acquisition systems realm, and to adapt more readily to parallel and asynchronous data access systems where the coherence of the system's "state" can be monitored and maintained between framework controlled state transitions (i.e. between configuration, initialization, algorithm execution, data serialization, and finalization stages).

It should be noted that in modern framework definitions and designs, there is typically an interplay and hand off between the framework which is executing the algorithm code, and the higher level WFMS layers which are responsible for the batch scheduling and delivery of the framework "jobs" to compute resources at different computing sites or on different types of computing hardware. Modern frameworks are typically designed to be aware of this macro level scheduling layer, and will often provide hooks to allow for the management layers to either transmit or receive information from the framework jobs. Information regarding framework configuration and various types of diagnostics can help both the framework and the WFMS layers more efficiently execute their missions, e.g. a framework will often provide monitoring diagnostics regarding the overall progress of the job which then allows for the management layer to stage queued work or initiate data transfers between sites. Equally, the WFMS may provide "late binding" configuration information to the framework which allows it to determine at

runtime sources of input data, or locations for output data which may be site specific or coordinated by the higher level management tools.

Lastly, modern frameworks as used by HEP today are also designed and responsible for adapting to heterogeneous computing resources. Modern frameworks must, due to the push towards exascale computing platforms, be able to execute their codes or bind/link their codes across not only different operating systems, but across different hardware architectures. In the case of DUNE this will be an enabling technology that allows DUNE algorithms to run on platforms ranging from commodity desktop/laptop systems, to large scale grid computing centers, and across the exascale era leadership computing facilities being built by the DOE and other major national and international computing centers. These facilities' architectures will span from the classic x86 CPU architectures to advanced, many-core GPU and AI/ML tuned accelerator systems. The framework's ability to deal with this diversity is a requirement from DUNE and is discussed in detail in the following sections.

4.2 Current status

Currently a set of frameworks has been used for the simulation of the DUNE detectors, and processing of data from the ProtoDUNE detectors. These frameworks have been developed or adopted based on the immediate needs of the sensitivity studies, simulation studies and ProtoDUNE running. These frameworks are based on earlier frameworks that have been used in the neutrino community and reflect in part the features that are needed for long and short baseline neutrino analysis.

For the far detector (FD) simulation and reconstruction efforts, the ProtoDUNE detectors and ND-GAr studies, the art framework which was developed at Fermi National Accelerator Laboratory (Fermilab) is the primary framework being used throughout the collaboration. The art framework, was originally developed based on the CMSSW framework used by the CMS experiment, but was designed to meet the specific needs of the neutrino and muon science communities. The art framework was designed as a multi-experiment framework starting in 2011 [105], and is currently used by 11 different major experiments in the HEP community including DUNE, NOvA, MicroBooNE, ICARUS, SBND, Muon g-2 and Mu2e. It is developed, maintained and supported by Fermilab's scientific computing division (SCD) and through a stakeholders committee consisting of the experiments using the framework.

The framework provides the data processing loop, manages memory, interfaces to I/O tools, defines uniform mechanisms for defining, associating, and persisting data products, provides a uniform mechanism for job configuration, stores job configuration information in its output files, and manages messages, random numbers and exceptions. The framework is highly modular and common modules which are shared across experiments have been developed for common infrastructure components such as the accelerator beam information systems at Fermilab, various data management systems and data cataloging systems, and database systems for calibration and conditions data access. Use of the art framework has allowed DUNE collaborators and other experimenters to quickly interface with Fermilab facilities, and with other facilities hosted by European Laboratory for Particle Physics (CERN) and other sites, in a common and consistent manner. The framework also allows for dynamic (runtime) configuration of the code modules, which has allowed additional common code libraries to be developed and run on top of the art base.

In particular, the Liquid Argon Software (LArSoft) toolkit is a collection of art plug-ins and associated

algorithm code, configuration files, static data such as geometry specification files and photon visibility maps which has been developed for the liquid argon time-projection chamber (LArTPC) detector community. LArSoft provides the interface to neutrino event generators such as Generates Events for Neutrino Interaction Experiments (GENIE) and cosmic ray generators and simulations such as COsmic Ray SImulations for KAscade (CORSIKA) and CRY, detector simulation via Geant4, custom simulation and reconstruction software, event displays and tutorials. Experiment-specific metadata and configuration database plug-ins assist in batch workflow organization. Like art, LArSoft is supported by Fermilab's SCD, and through collaboration and contributions from participating experiments.

It is worth noting that while art has some multithreading capabilities, these features are not extensively utilized by either DUNE or LArSoft. A significant amount of work is needed to transition DUNE modules, LArSoft modules, and most implementations of art services into thread-safe software so that the multithreading capabilities of art would become the standard workflow within DUNE.

For the gaseous detector simulations, the Gaseous Argon Software (GArSoft) software package is used. GArSoft is patterned on LArSoft as a layer of common algorithms which runs on top of the art framework. It provides a robust toolkit for simulating and reconstructing data from the ND-GAr concept detector or for other gaseous argon detectors. Like LArSoft, it provides interfaces to event generators and Geant4, custom simulation and reconstruction software, and event displays. It also provides simulation and reconstruction for ND-GAr-Lite. Unlike LArSoft, GArSoft is written, maintained, and supported only by the DUNE collaboration and is not shared with other experiments in the neutrino or HEP community.

In addition to the ND-GAr software, the DUNE collaboration's near detector groups have developed different tool suites for simulation and analysis of the proposed near detector designs. The ND-LAr software, which represents a group of standalone tools for simulating and reconstructing pixel-based LArTPC data. These tools and toolchains have been developed for simulating and analyzing SingleCube prototype data. The System for on-Axis Neutrino Detection (SAND) software efforts are based on collaborators' experience with the KLOE detector and its software. Portions of the SAND software are being used to provide simulation for the magnet and calorimeter systems, while new software is currently being developed for the 3D scintillator tracker (3DST) and other components of SAND. These software stacks are specifically being developed with maximum flexibility due to the design stage of the SAND concept. This software flexibility is important at this stage in the design in order to allow studies of different detector designs which can be used between the 3DST and the calorimeter. These tools are currently only used by the near detector group and have an independent software structure from the long baseline groups.

For higher level analysis tasks, sensitivity studies, and event selection, DUNE is currently leveraging the CAFAna framework. CAFAna is a high level framework based on the ROOT analysis software stack, which is designed to work with ROOT TTree's (ntuples). As mentioned in the previous chapter, ROOT will be transitioning to RNtuple [106, 107] as a replacement for TTree ntuples on the DUNE timescale, and so CAFAna will also need to transition. CAFAna is designed to provide bookkeeping facilities for neutrino flux information and exposures, which are needed for long baseline oscillation measurements and for short baseline cross section measurements. CAFAna is designed to work with output from data that has been processed with art and LArSoft and provides a more interactive environment for experimenters to explore the data, by using the ROOT scripting and interpreter interfaces, with specific CAFAna libraries and functions. CAFAna provides facilities for high level event selection, normalization of distributions, re-weighting distributions, and more.

The distinguishing features of the two frameworks, art and CAFAna, are that art works at the individual event record level, while the CAFAna framework allows for high level event selection but operates primarily on the resulting ensemble level distributions.

4.3 Framework Requirements

4.3.1 Requirements Process

Given the unique challenges that DUNE data pose, in 2020, the collaboration assembled a task force to examine and define needs and requirements for the DUNE software frameworks that are needed to accomplish the core physics missions of DUNE. The task force was charged with identifying "physics use cases" which could then be translated into software requirements that could be imposed on the current framework system, and applied to any new software systems[108]. The collaboration then approached the HEP Software Foundation (HSF) who assembled a panel of software framework experts from various experimental backgrounds to review these requirements and assess their compatibility with existing HEP software. The report [109] produced by this panel was discussed in a workshop involving the panelists and task force members, which resulted in a followup workshop to address the outstanding issues, mostly around the topic of concurrency and mapping DUNE software and data processing to HPCs and leadership scale computing facilities. The findings of the panelists, and the summary of the concurrency workshop [110], have been incorporated into the DUNE software framework requirements documented here. In particular a number of issues unique to DUNE were identified which must be addressed as DUNE and its software moves forward. We present these in the following sections with details regarding their need and their impact to the DUNE computing model.

4.3.2 Summary of Software Framework Use Cases

The complete set of use cases that was identified for DUNE has been enumerated, annotated and reviewed in detailed in [108] as well as the HSF report of section 4.3.1. As a result we present and discuss here the major requirements form those reviews that impact the computing R&D that DUNE will need to address and the use cases which have driven (listed in Table 4.1) the collaboration to place them upon the framework develop plans.

Table 4.1: Summary of resources needed per trigger readout for compute intensive tasks in ProtoDUNE and the FD. These numbers assume that reconstruction will load individual subunits such as anode plane assemblies (APAs) into memory one at a time, possibly in parallel across multiple cores. Note that the output size of SNB reconstruction is not listed because a the time of writing this CDR there was no consensus estimate for algorithm output.

Use case	memory	input size	output size	CPU time	input/core	cores/readout
units	GB	MB	MB	sec	MB/s	
Simulation+reco	6	100	2000	2700	0.00	1 to 150
FD reco	4	80	4000	600	0.13	up to 150
SNB reco	2	3.2×10^8	**	$3\text{-}10\times10^8$	0.13	10-30,000

One striking feature of the use cases that DUNE needs to support is the diversity in scale of the data atom. Somewhat counter-intuitively, although the raw trigger records are themselves very large, the

data atom (a single APA or charge-readout plane (CRP)) is relatively small [O(10 MB) per ROI], and compatible with the same high-throughput computing workflows adopted by most HEP experiments. Meanwhile at the analysis level although the quantity of data related to a single trigger record is small, nuisance parameter extraction requires correlations across all trigger records in an analysis dataset, which is the effective data atom. This is a very good fit to high performance computing, particularly if the analysis framework can take advantage of many HPC nodes at the same time. Experience from the HSF panelists encouraged DUNE to separate the analysis and production use cases when considering the software framework design.

4.3.3 Unique Software Framework Requirements for DUNE

While section 4.3.2 provides an overview of use cases, certain unique use cases drive the needs of the DUNE software framework to be fully compatible with the physics missions. We summarize these driving requirements here as a summary of the major work that needs to be executed for DUNE to have a software framework capable of supporting its physics program.

Driving use cases and consequences on framework development:

- Far detector partial region simulations and subsetting of temporal and spatial data.

While the actual visible energy signatures from the simulation of physics processes in the FD are likely to span only a subset of a detector module, as individual interactions are confined to a reasonably small volume and a prompt temporal footprint, the generation of these simulated interactions is not. This is due to the manner by which the current generation of detector geometries and simulations treat the relevant volumes, the overlaying of noise, cosmic ray induced activity, and other background processes. Without substantial effort to restructure how these processes are accurately simulated across the detector volume, while preserving the detector response, all of the simulated hits regardless of locality (within the sub-volume, typically 10-20% of the total module, spanned by the interaction) have to remain in memory at once during the initial simulation stages (i.e. we have to generate all activity across the detector as a whole and then after the full event topology along with its background overlays are determine, we can subset the event the discard information that falls outside of the primary regions of interest). This memory footprint for the full detector is substantial and presents difficulties for scaling to current and future computing platforms. The framework will need to be able to handle the management of these simulations and in particular will need to be able to subset them efficiently and effectively "page" them in and out of the active memory in order to propagate information between stages. Simulations of a 2x6 APA region in the HD detector have already been demonstrated, but this needs to be extended and made a native function of the framework's memory management and handling systems. This is also an area where auxiliary detectors, and in particular the photon detectors may add substantially to the memory footprint as discussed in section 3.4.4.

- Far detector partial region reconstruction and subsetting trigger records for data processing

Reconstruction algorithms running within the art framework, are already running into memory limitations when processing the 6 anode plane assemblies ProtoDUNE data trigger records. While most FD Trigger Records, whether initiated by cosmic rays, beam neutrino interactions, or atmospheric neutrino interactions, will result in activity in only one FD module, there are still extensive air shower events and

other physics events, even at the 4850' level of Sanford Underground Research Facility (SURF), which results in correlated activity spread across multiple modules. Processing data from a full FD module, with hundreds of thousands of channels and thousands of samples/channel will require sophisticated memory management to handle not only ingesting the data, but working with it through the different algorithmic transforms that are required for the digital signal processing, the reconstruction tasks, and for writing out the resulting information. There is however natural parallelism in the DUNE detector corresponding to the APAs. A framework designed to operate with knowledge of this parallelism, would be able to work with the individual readout subunits such as the anode plane assemblies or CRPs and process them either individually, sequentially, or in parallel in such a manner as to manage the total memory footprint while maximizing processing throughput. In contrast current frameworks, such as art, are not designed in this manner and can not natively handle this type of subsetting and scheduling.

- Temporal and spatial "stitching" of readout trigger records into new extended windows

In the event of a supernova burst, or other multi-messenger or long time duration events, the data from multiple files (or storage objects) will need to be combined across file boundaries to form complete views of the interactions and, perhaps more crucially, to avoid edge effects in the Fast Fourier Transform (FFT) (see section 3.8.1). In the case of a SNB Trigger Record, the data from ten's of seconds of readout will be 100's of TB in size and may have thousands of physical interactions spread across the active volume. Proper analysis of extended events like this require that data from different portions of the underlying DAQ readout be "stitched" across boundaries in both the detector spatial dimensions and also in time. However, current HEP frameworks make the implicit assumption that data atoms are independent of all other data atoms and that the temporal ordering of data atoms can be ignored when scheduling data processing tasks. To overcome this, the framework that DUNE uses will need to handle data that has a well-defined temporal ordering, and combine data from adjacent readouts, taking into account overlap regions, effective sidebands, and duplication of data products. This is a highly non-trivial task and one for which R&D is required for developing the methods for scheduling these types of merger operations within the context of a parallel or distributed computing environment.

- Contextual switching of primary data atom types for driving event loops

In contrast to the far detector supernova readout, the DUNE near detector due to its proximity to the target station, high fluxes, and cavern siting, will have many neutrino induced interactions within or crossing the detector volume during each LBNF spill window. The Gaseous Argon Near Detector Component (ND-GAr) expects approximately 60 overlaid interactions per spill. Most of these interactions will originate in its calorimeter, which has fast timing capabilities that can be used to disambiguate the interactions. Disambiguating these interactions will allow for a single trigger window to be effectively unfolded into the particle content of each interaction. Similarly, the LArTPC near detector is expected to have on the order of 20 interactions per spill which can also be disambiguated and separated out. From the framework perspective this subsetting of a data window into what has been colloquially referred to as "slices" changes the primary context of items that need to be looped over and analyzed. The DUNE framework will need to be able to handle this type of a contextual switch, so that it can run appropriate reconstruction and identification algorithms over each of the interactions, instead of over the original readout window. In much the same way that current frameworks are not designed to handle "stitching" of data atoms, current frameworks similarly are not designed to break data atoms down into smaller atoms while retaining the required bookkeeping information, and then looping and

scheduling algorithm modules over these.

Given the potential for pressure on memory resources, a requirement on the framework is:

- The framework must provide fine-grained control of memory, managing the memory needed by data products in the *data store* or its equivalent.

4.3.4 Modular Framework Design

For developers, highly modular code must be encouraged, allowing evolution or replacement of sub-algorithms that lend themselves to particular approaches. The codebase will therefore likely contain several alternatives for (sub)algorithms, the choice of which would depend on the available hardware. The framework will need to run in heterogeneous and potentially dynamic environments where the availability and type of co-processors may be known late. While the design of this technically challenging aspect is left to framework developers, it is worth pointing out the added challenge of developing algorithms for diverse hardware if the framework does not easily allow it. Therefore:

- The framework must separate the persistent data representation from the in-memory representation seen by algorithms
- The framework must separate data and algorithms

For developers, fast turnaround time is crucial:

- The framework should allow a rapid development setup with minimal overhead both in terms of writing code, compilation and startup time (including configuration)

It is noted that a sufficiently advanced Configuration system (see section 4.3.6.1) would be able to achieve this.

4.3.5 I/O Requirements for the Simulation/Reconstruction Framework

Given the uncertainty in the choice of data format:

- the framework must support reading and writing different persistent data formats
- The framework I/O read functionality must be backward compatible across versions.
- The framework I/O layer needs to provide a mechanism for user-defined schema evolution of data products.

The DUNE data model allows for Trigger Record data to be stored in persistable representations which are generated by customized hardware or which are optimized for specific acceleration hardware or computing systems. As a result, the data model expects that data will have custom "packed" representations that do not conform to 32-bit or 64-bit little-endian words. Furthermore, compression of the raw waveform data will be performed in the DAQ, though some data may arrive uncompressed. Some highly compressible data products may benefit from dedicated compression algorithms scheduled to run before output. Therefore,

- the framework I/O layer must provide a mechanism to register/associate and to run/apply custom serialization/deserialization and compression/decompression algorithms to data on write/read of that data from a persistable form.
- The framework I/O layer should support compression on output data in a manner that is transparent to users and is configurable. It must be possible to disable the automatic compression of output data.

Experience shows that it is highly desirable to be able to configure a maximum file size such that output files are the correct size for efficient storage and for units of data processing; currently a file size of several GBs is considered optimal. As this requires the closure of an existing output file and creation and opening of a new file (with sensible filename) then this needs to be addressed at the framework level.

- The framework I/O layer should allow a configurable maximum output file size and provide appropriate file-handling functionality.

Frameworks need to provide configurable, flexible I/O access so that experiments can control the output of their jobs in fine-grained detail. This is needed to save on storage and also experimenter time, as smaller datasets take less time to analyze than larger ones.

- The framework needs to support skimming/slimming/thinning for data reduction.

Similarly, processing and analysis is made much more convenient (or even possible) if the skimmed/ slimmed/ thinned output streams can be associated with information in other streams that may be stored separately. Some analyzers may need the auxiliary data streams while others may not, and so a framework job that produces outputs for collaboration use would need to read all of the necessary streams. This also allows efficient use of storage, as data does not need to be co-located in the same file to be available for processing. In the case of writing, I/O cost can be very efficiently amortized if several output streams can be written based on one input file.

- The framework needs to be able to read and write several parallel data streams. Labeling of data objects across streams should be intuitive and not error prone. Provenance information should support correlating related data objects across streams.

A common offline job need that goes beyond the $1 \rightarrow 1$ input to output data model is mixing real Trigger Record data with Monte Carlo simulation. Monte Carlo simulation is often not sufficiently realistic, perhaps it fails to capture the time dependence of detector conditions or the generator or detector simulation simply lacks sufficient accuracy for the physics use case. In this case, Monte Carlo simulation can be augmented by adding actual detector data, e.g. to embed single tracks or entire Trigger Records into simulated data and reconstruct them as if they were actual Trigger Records. This can lead to a $2 \rightarrow$ many input output situation with asynchrony in both input and output.

- The framework needs to allow experiment code to mix simulation and (overlay) data.

4.3.6 Reproducibility and Configuration

The reproducibility of physics results and the knowledge of how physics results were obtained is essential to DUNE and to the neutrino community as a whole. It must be possible both to replicate physics results using identical input data, or to repeat an analysis using a different set of input data with an identical sequence of identically configured algorithms.

For the purpose of these requirements, we consider data to be reproducible if for integer fields the products in a given event record are bitwise identical. Floating point numbers in a given event record are the same to within the floating point precision of the machine they were generated on. The ordering of data products within an event record are the same. These criteria for reproducibility are designed to allow DUNE to simulate "rare" events with well defined single event sensitivities, and to perform numeric transform operations on data values across different hardware platforms and runtime concurrency topologies.

- It is highly desirable that the framework broker access to random number generators and seeds in order to guarantee reproducibility.
- The framework must provide a full provenance chain for any and all data products which must include enough information to reproduce identically every persistent data product. By definition, the chain will also need to include sufficient information to reproduce the transient data passed between algorithm modules, even though the product is not persisted in the final output.

This will include the computing architecture, including any specialized hardware used, on which the application is run as well as the runtime environment, execution model and concurrency level that the application used. It is likely that a full picture of the necessary metadata will also require information only known to the WFMS. In such a complex environment, it is highly desirable that the framework provide support to allow the effect of configuration changes and computing environments to be easily understood.

4.3.6.1 Configuration

Configuration is distinct from initialization of the framework objects; configuration happens first and it is particularly important for multi-threaded frameworks to have a fully configured framework instance in the initialization phase. Given the abundance of variations that make up HEP workflows a robust and easily programmable configuration system is a foundational component of all modern frameworks. Some use a strictly declarative language and some use a Turing complete language. The former must be augmented with scripts that write the declarations in a Turing complete language (usually it is part of the WFMS) because it turns out that control flow is a requirement. Given this, there is a requirement that:

- The framework should provide a configuration language as a foundational component so that it can ensure coherence of its configuration.

As discussed in section 4.3.3, the WFMS needs to be able to supply framework configuration parameters such as input file(s) or random number seeds to each framework application instance, which it should do using the framework's configuration language. To minimize errors, these parameters should be

self-describing and validatable, and so:

- the framework should provide a suitable API so that algorithm writers can ensure their required parameters are self-describing and validatable.
- The framework should provide the concept of parameter sets that are nestable. The set of all parameter sets that define a framework application instance should be identifiable, referred to here as a FrameworkConfigID. Tracking that identity is one of the ingredients necessary to ensure scientific reproducibility. However some parameters do not (and should not) change the algorithmic results, such as a debug print flag. Independent of the state of such a flag it should be possible to define equivalence between FrameworkConfigIDs.
- The framework configuration system needs to have a robust persistency and versioning system that makes it easy to document and reproduce previous results. It must be possible to create, tag, check-sum, store and compare configurations. This configuration management system should be external to the framework or data files so that configurations can reliably be reused and audited.
- The resulting state of the configured framework and its components should be deterministic and reproducible given a set of environmental conditions which include the available hardware, operating system, input data, etc.

Related to the discussion on rapid turn-around in section 4.3.4, but also important central features of the configuration system:

- it is desirable that it should be possible to only configure those framework components required for a particular data processing use case.
- it must be possible to derive the input data requirements for any algorithm, in order to define the sequence of all algorithms needed for a particular use case.

4.3.6.2 Conditions Configuration

In addition to the Trigger Record data, data processing requires access to other data from various sources, for example slow controls, detector status, and beam component status. Such data is referred to generically as conditions metadata, introduced in section 5.1.1, and also includes e.g. detector calibrations and any data external to the Trigger Record. The time granularity or "interval of validity" of this data varies by source and is typically of much coarser granularity than individual Trigger Records, e.g. calibrations may be valid for months of data taking. Meanwhile there are often several versions of conditions metadata and correlations between conditions metadata is not uncommon, making the coherent management of conditions metadata a challenge in itself. For this reason, conditions metadata management should be external to the framework.

Access to the external conditions data should preferably proceed via REST interfaces that support loose coupling of the framework and the external conditions metadata management system. As conditions metadata may need to be transformed from its persistent format into a format required by an algorithm, and as multi-threading makes the cache validity of conditions metadata complicated:

- the framework must provide a thread-safe conditions service that is a single point of access to conditions metadata.

- The configuration of the framework conditions service should ideally be via one configuration parameter (a global tag).

Developers must not hard-code conditions data in their algorithms, although

- it must be possible to override a subset of global tag configured conditions for testing purposes.

Developers usually find it convenient if such alternative conditions payloads can be provided outside of the main managed conditions system, e.g. via a local file.

4.3.6.3 Concurrency

The intrinsically parallel structure of the DUNE detector data, and the types of digital signal processing and machine learning algorithms that are being used to analyze it, benefit highly from both the multi-core CPU architectures and new hardware based accelerators such as GPUs, Tensor processor units and other many-core architectures. However, while there is significant benefit to using these heterogeneous architectures, there is also a cost in terms of the complexity of needing to manage the concurrent operations and ensure that they are computationally safe and reproducible.

In the context of DUNE, the framework, in being able to schedule module work, will need to be both thread aware and accelerator aware. By this we mean that the framework will need to ensure that the modules it schedules and executes in parallel threads, or through offload to accelerators, preserve the coherent state of the memory, data, I/O and other resources that the framework controls and serves as a gateway to.

While the goal of the DUNE collaboration will be to write modern, high parallelized, thread- and accelerator-safe algorithm codes, we recognize that there will be code within the DUNE code base that will rely on serial programming techniques. We also acknowledge that portions of the DUNE code base will leverage external libraries which may or may not be thread safe, or which may need exclusive access to accelerators. For these reasons we impose upon our framework the following requirement:

- The framework must be able to schedule thread-safe and non-thread-safe task modules, while ensuring coherence of the data, memory and I/O systems.

- The framework must be able to schedule tasks modules which require access to hardware accelerators, while ensuring coherence of the data, memory and I/O systems.

At the same time the framework must be able to perform this scheduling in the context of different thread pools or memory maps which are commonly controlled through different parallelism toolkits (i.e. OpenMP, Intel TBB, Intel OneAPI, OpenACC, etc...). This needs to be done due to the way in which different toolkits can interact and result in oversubscription of resources. For DUNE, where researchers may be leveraging major data science, machine learning or other commercial libraries, this is particularly important because it may imply that the task modules need to provide “hints” which will help the framework schedule the modules or change the sharing model that is used during portions of the code’s execution.

- The framework should be compatible with the use of [external] multi-threaded data processing techniques and access to co-processors resources. The framework should be capable of scheduling these resources in an efficient manner with respect to a given runtime environment.

This type of parallel scheduling and access problem is an active topic across more domains than just HEP. Developing a framework system that meets DUNE's needs will require both significant development work and not-insignificant R&D with regards to the computer science and data science techniques which will ultimately be used. This investment is however has the potential to reap massive benefits to DUNE in terms of the execution times of the algorithm codes, and the ability of the collaboration to use the exa-scale computing platforms that will be dominate and supported across the DoE leadership computing facilities.

4.3.6.4 Externals including Machine Learning

Machine learning is already heavily used in analysis of ProtoDUNE data and has become an integral part of scientific analysis across HEP. The developed by DUNE framework should give special attention to machine learning inference in the design, both to allow simple exchanges of inference backends and to record the provenance of those backends and all necessary versioning information. In particular the framework needs to support the injection of AI/ML model configuration and be able to describe not only the foundational model but the training state of the models or inference systems. It is tempting to require that the framework natively support a foundational model description language, such as the Open Neural Network Exchange (ONYX) standard and leverage its "model zoo" as a path towards DUNE's use of ML. However while this is a current push within the ML community in 2022 for the description of natural language processing, facial recognition and other major AI/ML enabled technologies, the scope and timing of the DUNE framework development would advise against a requirement that effectively names an implementation specification.

Instead, we require that the framework support configuration of AI/ML systems through deterministically defined configuration systems, and that this system provide the framework with provenance information, or provide a derivable method for obtaining provenance information, sufficient to fully describe the state of any external ML system and allow for reproducibility of that state at a later time with the same fidelity as originally described or configured.

In addition, the framework should be able to work with both the near detector (ND) and FD data on an equal footing and within the same job, which may require the simultaneous configuration, instantiation, management, and scheduling of multiple models or model instances based on the detector context.

In this respect, we require that the use of external libraries relating to machine learning, machine learning inference or other functions which are not part of the core framework or algorithms code stacks of the experiment be registered and treated by the framework proper in such a manner as to preserve the integrity and reproducibility of the data analysis.

- The framework should give special attention to machine learning, machine learning inference, and the use of external libraries and codebases in its [the framework's] design, both to allow simple exchanges of ML/external backends and to record the provenance of those backends, their states, and all necessary versioning information to permit full reproducibility of the scientific results.

4.3.7 Analysis

Based on the experience and recommendations of the HSF panel experts, the software framework used for large scale production may be different to that preferred for late stage data analysis. This difference may not necessarily be due to technological challenges or missing functionality in the primary production framework, but may arise due to sociological factors that can, and have in the context of other HEP experiments, played a large role. It was noted by the HSF panel that physicists may be willing to sacrifice performance, capability, and flexibility in favor of simplicity. In these cases, it has been observed that lighter-weight, less capable, less performant, less stable and less supported frameworks may grow organically from members of the collaboration, and even become preferred methods for addressing niche topics.

That being said, there is no clear dividing line separating the higher level analysis from the use cases served by the data processing framework, and the choice may vary depending on the use case or analysis group in question. In the following discussion, requirements more often associated with analysis use cases are discussed, but data processing framework developers should also consider them as many can, and should, be common to the analysis chains.

Analysis tasks include the extraction of oscillation parameters, comparisons between data and Monte Carlo, extraction of calibration and detector performance parameters, cross-section measurements, measurements of atmospheric, solar, and supernova neutrinos, and searches for non-standard phenomena. For these tasks, the event-by-event paradigm is not a good match in all cases. In contrast to the event-by-event model, ensemble distributions and representative spectra are the quantities of interest and while they can be generated by iteration or looping over event records, neutrino candidates, or data-atom style data organization, they often may undergo substantial processing in their own right, whether that be in the form of renormalizations and re-weighting, or different type of peak finding, filtering, de-convolutions and other similar ensemble operations. For example, the main work of an oscillation fit is the evaluation of many different combinations of oscillation and nuisance parameters against a given distribution of observed energies. Yet, for particle level analysis, while time or spatial slices provide sub-Trigger Record control flow, particle candidate control flow ignores the Trigger Record structure entirely. Indeed it is this contextual switching between the fundamental quantity of interest, and the ability to perform and propagate operations upon these contextually defined quantities that defines our analysis environment.

These analysis problems also tend to lend themselves to "columnar" analysis techniques and organizations of the data. While it is true that the manner by which we acquire the data is inherently "row-wise" given the way we generate our readout records in a temporal sequence, there is nothing that prevents the data from being effectively transposed into a columnar or table based structure at later points in the processing chain. These tabular organizations are almost universally more efficient in terms of ensemble operations, memory efficiency, data parallel operations, and other factors which influence the late state analysis process. As mentioned in previous sections, DUNE aims to support both paradigms for data organization as it has been shown to have benefits not only in our custom analysis codes, but in leveraging machine learning and other external products which expect these types of data frame oriented views of the information.

It should also be noted that analysis work will be undertaken by a large number of collaborators with varying levels of experience, in comparison to the mainline or centralized data processing efforts which

will be engaged in by smaller teams of individuals with more expertise in the data processing workflows. In these use cases, simplicity, rapid execution, feedback, and iteration are of paramount importance:

- The analysis framework should have a low entry-level in terms of software expertise.

Analysis files, of course, are derived from the data-processing framework files, and it must be possible to reconstruct this history. Due to the very large number of Trigger Records expected to be summarized in a single analysis file, the size requirements, and the fact that per-Trigger Record information remains available in the parent files, we require:

- Analysis files must record their parent framework files, but no Trigger Record provenance is required. The full provenance information need not be retained in analysis files as this could easily become larger than the data itself.

One common and insidious class of mistakes is errors in exposure accounting and normalization. This is also a problem that is entirely solvable at the technical level. Each individual Trigger Record (beam spill or other trigger) has exposure associated with it, whether POT or livetime or both. When filling a summary histogram from processing Trigger Records, the exposure should be calculated and stored as an integral part of the histogram, and operations between histograms should take correct notice of the exposure, e.g. ratio of one large exposure sample to a smaller exposure sample should produce a dimensionless ratio that has allowed for the differing exposures.

- The framework must have native support for exposure accounting (POT and livetime), so as to make errors of this sort difficult.

All but the simplest analyses require a treatment of systematic uncertainties. There are three main technical means by which these systematics can be introduced. The most common, and most convenient, is reweighting. For example, the effect of various cross-section and flux uncertainties may be encapsulated by applying weights to Trigger Records of certain categories, to increase or decrease their representation in the final spectra. Secondly, Trigger Record data may be shifted. For example, an energy scale uncertainty may be most conveniently represented by rewriting of Trigger Records to increase or decrease reconstructed energies by a certain amount. Finally, the least convenient method is alternate simulation samples. The profusion of files requiring processing and bookkeeping makes this a heavyweight option, but in the case of uncertainties early in the analysis chain with complex effects, it may be the only way to handle them accurately. The treatment of systematics is cross-cutting across all analyses, it is important it is handled correctly, and the framework is able to offer substantive technical assistance. In addition to being able to handle multiple input data streams:

- The analysis framework should provide some means of cross-referencing (labelling) multiple input streams to correlate them in order to facilitate evaluation of systematic uncertainties.

For oscillation analysis, it will be important to work with both ND and FD data. Whether in an explicit joint fit, or where extracting constraints from the ND to apply to the FD analysis, there must be a uniformity in the treatment of various systematics. In general, experience gained with the ND (where the majority of analysis work is likely to happen) should be transferable to the FD. This re-emphasizes

the importance of the framework making minimal assumptions about the Data Model.

- The analysis framework should be able to work with both ND and FD data on an equal footing, and within the same job.

Experience shows that oscillation fits accounting for large numbers of systematic uncertainties are resource intensive, while analysts will likely only have access to modest local resources for prototyping and development. Therefore the framework should make scaling and concurrency transparent both to the analyst and the developer as far as possible. The use of declarative analysis techniques should be strongly encouraged to support this even when co-processors (and low-level implementation) changes.

- The analysis framework should easily scale from local resources such as a laptop, up to multi-node compute at an HPC.

It is noted that MPI-enabled frameworks written in python already exist and would be a good match to the above requirements.

4.4 Development Plan and Effort Profiles

The production use cases for the second run of the ProtoDUNE detectors can be handled by the existing art/LArSoft based software framework and toolkits. We therefore base our development timelines on the need to develop our next generation framework for DUNE, assuming that the framework will need to be commissioned and ready for production prior to far detector installation. It will be validated against the late stage ProtoDUNE analyses, and also used for the first detector data for the far site systems, and for pre-cursor systems which will be need to be commissioned prior to the far detector installation. Proto-DAQ systems will produce readout data prior to the formal start of detector operations that must be processed through the offline computing system.

Currently the compute architecture landscape is evolving rapidly, a challenge being experienced by running experiments with large code bases. DUNE is engaging with the HSF to ensure that progress made elsewhere can be effectively utilized in its own software frameworks. The general strategy is to tackle DUNE-specific challenges, discussed in Section 4.3.3, as soon as possible, and then work on consolidating best practice on heterogeneity from the wider community into a core framework. Following the delivery of a core framework, algorithmic code will be migrated by the wider collaboration, similar to the development pattern used by previous HEP collaborations that was reported by the HSF panel [109] (see appendix).

The development plan starts with dedicated effort to accommodate the dynamic time window needs of the full DUNE physics program, detailed in Section 4.3.3. Given that this as-yet-unknown solution is both central – and critical – to the success of the framework, Fermilab has funded LDRD investigations in 2022/2023 to provide proof of principle and prototype data handling systems. Successful completion of these prototypes will significantly reduce risk associated with the extended readout events and allow us to reduce our contingencies relating to later phases of the development, and in particular to the supernova burst science case. The work will also investigate techniques for efficient minimization of runtime memory footprints and restructuring of simulation chains to accommodate the far detector

simulations.

The next major development work concerns the rapidly changing landscape of compute architectures. The shift towards heterogeneous and accelerated architectures by major platform manufacturers such as Intel, AMD and NVidia, are very relevant to DUNE due to the numerous compute problems that are well-matched to accelerators. These include a mixture of both highly accelerate-able signal processing and signal transform work that dominates the early stages of the data processing chain, and emphasis on AI/ML techniques which are more prominent in the later stages of the pattern recognition and tracking, event identification and higher level analysis computing chain. After the LDRD work, DUNE will incorporate best practice from HSF, following a clearer DOE direction for future facilities expected on a timescale of around 2024.

An estimate of the effort needed for core software framework development and to migrate algorithmic code uses the HSF panel report appendix, which provides approximate numbers for several experiments with diverse frameworks. In that report the HSF collected information from major HEP experiments which had either developed a new "green-field" framework to meet their experimental needs, or had migrated from a legacy framework to a newly developed framework which was capable of satisfying their scientific needs. The report included information from LHCb, CMS, Atlas, ALICE and NOvA along with details of the development and migrations which were undertaken. When the information that was provided was normalized by the size of the code stacks that were migrated or developed, a fairly consistent estimate was obtained across the experiments with some deviation based on simultaneous DAQ software development that was undertaken by ALICE. Our resulting estimate of 6 core FTE years was then obtained by scaling the current DUNE codebase and including both a moderate allowance for the expanded feature/requirement set that has been outlined here and modest contingency. Based on this estimation methodology, 6 FTE years is estimated for core framework developer effort from 2024-2027, and 10 FTE years is estimated to be needed for algorithmic code migration to deliver a fully functional framework in 2027. It should be noted that of this effort the initial core framework development requires highly specialized domain expertise in frameworks, workflows, and parallel computing techniques. This domain expertise profile is consistent with the highly technical computational science staffs that are available from the collaborating laboratories, and which have significant experience in similar projects within the HEP communities. The majority of the algorithm and physics code migration, in contrast, requires domain expertise aligned with the neutrino sciences and is well covered by the wider scientific expertise of the DUNE collaboration.

With these effort profiles we believe that a realistic development pathway to a full production framework for DUNE could be completed by mid to late 2027. The development effort timeline can be summarized as commencing with high risk R&D in 22/23 (covered by Fermilab LDRD), Core Framework effort from 24-27 (requiring funding for new expert effort), and algorithm migration in 26/27 (combining effort from the collaboration at large with support from Core Framework experts). Increased expert effort on core software framework development beyond the foundational LDRD-funded work will therefore be needed to deliver the core software framework for data processing, and this will form part of the next round of multi-institutional/national funding requests for the period 2024-2027. A core framework delivered by 2026 would allow two years for algorithmic code migration with full support from core framework experts.

Chapter 5

Databases

5.1 Introduction

In order to accommodate the large range of metadata that will be tracked by Deep Underground Neutrino Experiment (DUNE), the DUNE DB structure comprises several databases specific to the information, or metadata, that they contain. The subset of all DUNE metadata that is required, in the strict sense of being absolutely necessary, for data processing and analysis needs to be carefully identified and assessed. We refer to this subset of metadata as "conditions metadata", see section 5.1.1. It is critical that users be able to access conditions metadata throughout the full data processing and analysis chain with as little burden as possible. To achieve this, users will interact with a centralized high-level interface described in Section 5.2.

The DUNE experiment is expected to operate for 20-30 years and the DUNE databases need to be reliably maintained and operated for that entire period. In order to accommodate this requirement, the database system should not rely on implementation solutions that possess the risk of becoming unavailable during the operation period. Although no solution is without risk, the general DOE lab strategy has been to adopt open-source, non-proprietary solutions, and this is the strategy that DUNE will follow. Currently the databases housed at Fermi National Accelerator Laboratory (Fermilab) use the open-source PostgreSQL (Postgres) relational database management system. Postgres is supported by the scientific computing division (SCD). DUNE would also like to benefit from the close collaboration between DOE laboratories to improve service availability and mitigate single-point-of-failure risk by having secondary databases at other DOE labs. Brookhaven National Laboratory (BNL) is a good example of a lab that already provides database services to international experiments using the same Postgres technology.

It is expected that there will be reconstruction and analysis jobs distributed across large numbers of traditional grid-based and high-performance computing (HPC) systems and that database access will need to be able to scale appropriately. Additionally, it is important to ensure that users are able to work on analysis tasks when unable to access the database directly through a network connection.

Some of the database solutions outlined in this document have been deployed and tested during Run

I of the ProtoDUNE experiment. Experience coming from this will be briefly described in the sections below, when relevant. The ProtoDUNE-II experiment will provide a further testbed for the database systems proposed for DUNE.

5.1.1 Conditions Metadata

conditions metadata is defined as the information necessary to understand the context of physics data, e.g., beam data or calibrations. Metadata can be indexed by either time, run, or fraction of a run (subrun). An interval of validity (IOV) defines the period, indexed by any of the three options, for which a given metadata is valid. Alignment constants are an example of conditions metadata that will likely be valid for several runs. The DUNE conditions DB will have APIs that provide easy and transparent access to all conditions metadata independently of technical details.

Time-indexed metadata will, in some cases, be sampled at a rate higher than typically needed by offline users. In these instances, the metadata will be filtered down, or interpolated, to a lower rate for inclusion in the conditions DB. In general, metadata falls into two categories, interpolated and non-interpolated, where an example of the latter would be run-indexed values pertaining to run configuration. Interpolated metadata, e.g., values read back from the slow control system, can be interpolated through a rolling average or updated on changes to the values. The method used will depend on the natural variation of the value being recorded and the physics use case. The database group will provide interpolated values to match the physics requirements of the experiment.

Conditions metadata will in general be stored in appropriate databases but there will be some cases where it is more reasonable to include the metadata with the raw trigger records instead. An obvious example of this is metadata that changes at the individual trigger-record level, i.e., trigger-bit information.

The following table contains classes of conditions metadata:

Table 5.1: Example conditions metadata values and types, and the databases in which they are stored. (I) indicates interpolated metadata.

Conditions Metadata Type	Example(s)	Database
Run Configuration	Start Time, config file	Run Configuration
Detector Conditions (I)	TPC high voltages	Slow Control
Beam Conditions (I)	Horn polarities, beam current	IFBeam
Hardware Information	Component history	Hardware/QC
Calibration Constants	Channel gains	Calibration
Physics/Hardware Locations	Channel maps	Geometry
Data Quality	Good runs list	Data quality monitoring (DQM)

A DUNE metadata task force was assembled in 2020 and a resulting report discussing the interfaces between online and offline systems can be found in reference [111].

5.2 Conditions Database

The conditions DB is a high-level centralized database that provides an easy-to-use interface for users and reduces the number of database connections required by offline processes. This will ensure that jobs will be "lightweight" and processing time will not be extended due to database accesses. The granularity and content of the conditions DB will be well-matched to offline needs by design, which will allow the heavy use of caching technologies to optimize resource usage. The design of the interface between the data processing software framework (see chapter 4) and the conditions DB will play an important role if all of the access patterns needed by DUNE are to be well supported. Experience from other HEP experiments with distributed computing resources show that careful design of this interface is crucial to the success of the computing model[1]. Figure 5.1 shows the relationship between the conditions DB and the other DUNE databases. The "Master Metadata Store" is an intermediate database that allows for potentially heavy data interpolation tasks to run without creating extra load on the other, often critical, database services. It provides separation between the online and offline worlds at the cost of partial data duplication. A design utilizing the Master Metadata Store also allows for maximum flexibility without the need for a predetermined schema. The final database system design for DUNE will benefit from experience on ProtoDUNE with this design.

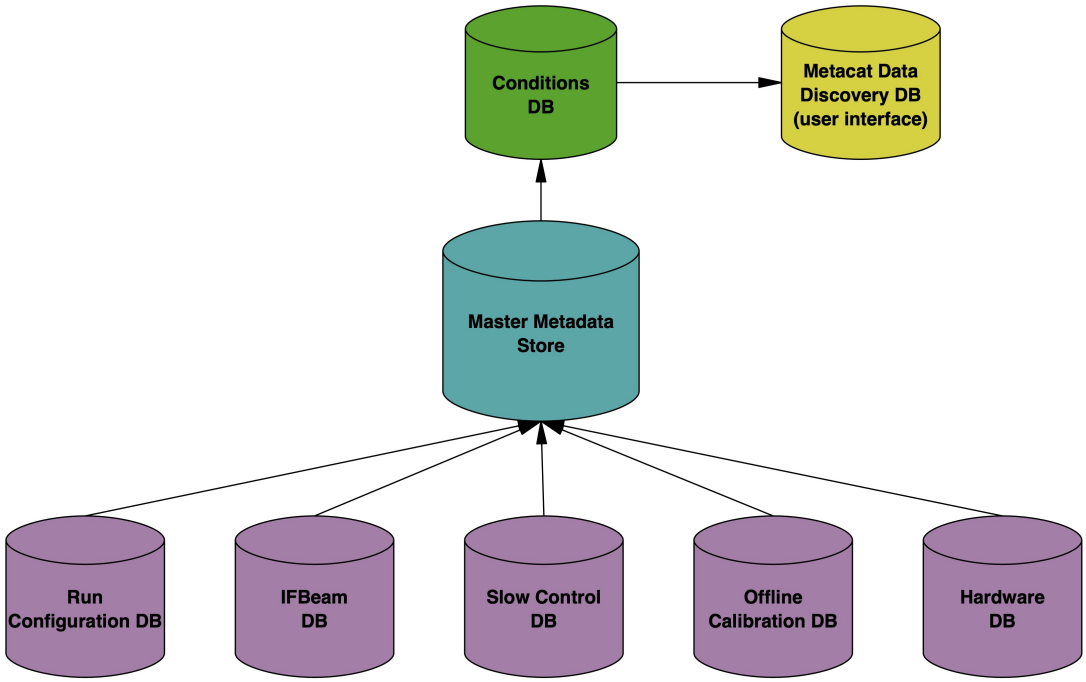


Figure 5.1: Map of DUNE databases showing the conditions DB, the Master Metadata Store and the lower-level DBs. The arrows illustrate the flow of metadata. Offline access to conditions metadata is made through the conditions DB, which contains a subset of information from the Master Metadata Store.

[1] Overlays (see section 3.7.1) are an example of a workflow that can cause issues for conditions DB access if this interface is not well designed.

5.2.1 Conditions Database for ProtoDUNE II

In order to provide a balance between the availability of the largest set of metadata possible and allowing schema evolution, the ProtoDUNE II conditions DB will employ an unstructured approach utilizing an unstructured database Unstructured Conditions Database (uconDB) [112]. It is expected that the schema for the conditions DB will evolve during the lifetime of the experiment and therefore it is advantageous to be able to avoid the constraint of committing to a schema in advance. The uconDB stores metadata from several specific databases and sources, like the data acquisition (DAQ) system, in "blobs" corresponding to temporal periods (run, time blocks, or IOVs). Each blob will contain a JavaScript Object Notation (JSON)-formatted record of metadata. Folders will be used to hold metadata corresponding to time and run keys. Tools will be provided to correlate between the two.

The following is an example fragment of DAQ metadata from the first run of ProtoDUNE. A typical file from this run was on the order of 10 MB.

```
{
Start of Record
Run Number: 5185
Packed on Oct 11 22:19 UTC
#####
boot.fcl:
#####
DAQ_setup_script: \
"/nfs/sw/work_dirs/dune-artdaq_artdaq_v3_03_00_beta/setupDUNEARTDAQ"
PMT_host: "localhost"
PMT_port: 5400
debug_level: 1
partition_number: 0
tcp_base_port: 15000
request_port: 3000
request_address: "227.128.12.25"
table_update_address: "227.129.1.128"
routing_base_port: 10010
zmq_fragment_connection_out: 17437
...
}
```

5.2.2 Conditions Database for DUNE

For DUNE, the interface between the conditions DB and the software framework will become particularly important. The consensus among several HEP experiments was reported in an HEP Software Foundation (HSF) whitepaper [113]. The use of a lightweight relational database schema allows a single point of entry, a "global tag", to configure conditions metadata access for the framework. Evolving the ProtoDUNE solution to benefit from that experience is foreseen by the database group. Further details on the development plan can be found in section 5.10.1.

5.3 Run Configuration Database

The run configuration database contains the intended configuration of the detectors and DAQ during data collection – physics or otherwise.

Metadata contained in the run configuration database includes hardware settings, run type, and run start and end times. Table 5.2 contains some examples of typical metadata that will be contained in the run configuration database.

Table 5.2: Example metadata values and types stored in the run configuration database.

Metadata Value	Type
Start of run	Time
Readout window size	Integer
Readout trigger type	Integer
Readout firmware version	Integer
Baseline start	Integer
Shifter comments	Text
Run end status	Integer

The majority of run configuration metadata comes from the configuration files used by the DAQ system during run execution. Some additional metadata collected at the end of the run, or shortly thereafter, may also be included. Examples are run completion status and comments made by the shifter during the run or in run-related checklists.

Parameters used to configure the run will be collected and packed into JSON-formatted blocks in a single blob corresponding to a DAQ run.

5.3.1 Run Configuration Database for ProtoDUNE II

Following the completion of a run, the run configuration parameters corresponding to the run are read from a MongoDB [2] set up by the DAQ group for immediate archiving of run configurations. The DAQs metadata is then packed into a single "blob" of key-value pairs in JSON format. Any additional information, such as end-of-run time, are added to the blob, which is then transferred to the uconDB at Fermilab. A typical metadata blob is on order 10 MB in and contains more information than most users will want to use. An additional step of reducing the metadata is performed to produce a subset of metadata needed by offline users. The reduced set of metadata is stored in a single table in a relational database referred to as the "run history database." An interface is provided to users, enabling them to retrieve run numbers and file locations based on queries of the history database. Figure 5.2 shows a diagram of the flow of metadata from ProtoDUNE DAQ to users.

[2] MongoDB©, `https://www.mongodb.com`

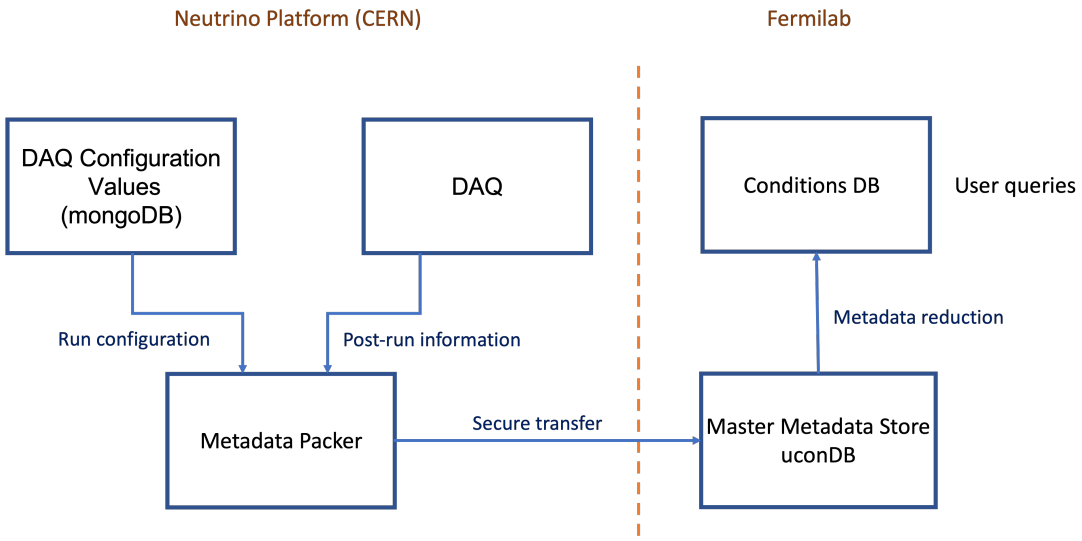


Figure 5.2: Flow of metadata from ProtoDUNE DAQ to user interface.

5.4 Data Quality and Monitoring Database

The Data Quality and Monitoring Database (DQMDB) contains monitoring histograms and metadata derived from data collected during operation of the DUNE detectors. The DQMDB is an online database and the histograms it stores to assess and monitor data quality are critical for the operations of DUNE, but not directly relevant for offline data processing and analysis. The derived data quality metadata, which include boolean flags indicating the results of the online data quality assessment algorithms, are relevant for offline analysis and this small subset of DQMDB data will require an interface with the conditions DB either directly or via an offline replica of the data quality database.

5.5 Offline Calibration Database

The calibration database contains calibration constants determined from collected data corresponding to IOVs. The conditions metadata from the calibration system will result from offline calculations using data collected from the DUNE detectors. There will generally be multiple versions of calibration constants corresponding to the same IOV, and the conditions DB will provide coherent access to the appropriate version of these calibrations via e.g., the global tag mechanism.

5.6 Slow Control Database

The slow control DB contains metadata specific to the state of detectors at the time those data were collected as well as before and after. Examples of slow control metadata are measurements of power supply voltages and currents, and temperatures. Each slow control quantity corresponds to a particular device. The slow control DB metadata is time-indexed and hence must be interpolated. Additionally, different devices will be sampled at different rates.

The slow control metadata is captured via a Supervisory Control and Data Acquisition (SCADA) system that is the responsibility of the Slow Control and Monitoring group. The SCADA system pushes values

to a back-end database, where the DB flavor is tied to the SCADA solution. The SCADA system can provide data reduction through filtering prior to insertion of metadata into the back-end DB, which reduces the workload on any API used to move the metadata to the conditions DB.

5.6.1 Slow Control Database for ProtoDUNE II

The ProtoDUNE experiment has been using an Oracle[3] back-end DB for the slow control system. As SCD does not support Oracle, the information from the Oracle database must be extracted and moved into a Postgres DB at Fermilab. Any filtering of the metadata not handled by the SCADA system when populating the Oracle database can be handled by the API that transfers the Oracle records to Postgres.

No data filtering was provided by the SCADA system for ProtoDUNE Run I but for Run II it is expected that the slow controls conditions metadata will be filtered based on the physics needs.

5.7 Beam Conditions Database - IFBeam

The beam conditions database, IFbeam[114], will contain metadata related to the condition extracted beam and corresponding diagnostics. The functional form of this database is essentially the same as that of the slow control database. A large number of devices are sampled into the IFbeam DB. The IFbeam metadata transferred to the conditions DB will be a coarser subset of the original set.

Quantities contained in the IFbeam DB include beam currents, horn currents and polarities, and beam monitoring instrument metadata.

Table 5.3: Example metadata values and types stored in the IFbeam database.

Metadata Value	Type
[0.5ex] Horn 1 Polarity	Integer
Horn 2 Polarity	Integer
Beam current	Float

5.8 Hardware Database

The principle purpose of the hardware database (HWDB) is to track the lineage of hardware components and record the results of their quality control (QC) tests. In this context a component can be a sub-detector module or any of the individual parts comprising it. For example, a readout board is a component as is a mezzanine daughter board or programmable logic chip mounted on the readout board. The lowest level component tracked within the HWDB will be unique to the corresponding hardware system.

A requirement of the HWDB is that any component, or part, stored in the database must have a unique identification number assigned to it, coordinated by the DUNE Integration group. The part numbers will

[3] Oracle©, `https://www.oracle.com`

be designated as shown in Table 5.4. The project field corresponds to DUNE detectors (D), integration (I), LBNF (L), and future project (P). The project identifier is allocated by the project management team while the other identifiers are left for the various hardware consortia to assign. There are additional fields not listed as they are not relevant to the HWDB. More details of the parts identification number can be found in [115].

Table 5.4: Unique parts identification number assigned to each component stored in the hardware database.

Project	System ID	Subsystem ID	Item Type ID	Dash	Item Number
D/I/L/P	01-99	001-999	00001-99999	-	00001-99999

Hardware DB metadata will reflect the complete lifetime of the detector component, including the following:

- Procurement
- Fabrication
- Quality control testing
- Shipping and storage
- Installation
- Maintenance

The relationships between components will be reflected in the HWDB. Metadata corresponding to multiple instances of events such as QC tests will be handled using time series within the database.

The database group will provide an interface to the HWDB and each hardware consortium will be required to ensure that their metadata is inserted into the database. Given the wide range of hardware consortia, the international nature of the experiment and the fact that individual consortia must manage their own construction projects, it is beyond the scope of the database group to dictate how the consortia will handle data entry workflows for the HWDB. Instead, the database group will consult with the consortia, and in particular their database group liaisons, developing simple APIs, providing documentation for the HWDB and advising on efficient methods for working with it. Sharing of tools will be very strongly encouraged to reduce the duplication of effort as far as possible.

5.9 Service and Maintenance

Most, if not all, of the DUNE databases will operate in advance of the full DUNE experiment coming online and these databases will need to be maintained and serviced once they are operational.

The second run of the ProtoDUNE experiment (ProtoDUNE-II) will employ a suite of databases that will be the precursors to the full database system that will be in place for DUNE. Each of these databases (run configuration, beam instrumentation, conditions, slow controls, and hardware) will require stable monitoring, maintenance, and service to address operational issues that will arise in the lead up to and during the running of the ProtoDUNE II experiment.

Monitoring will be achieved using automated web-based tools and responses to offline database issues will be made within an 8-hour period corresponding to a typical operation or production "shift", i.e. 5 days a week[4]. For ProtoDUNE-II databases will be located at both Fermilab and European Laboratory for Particle Physics (CERN), both of which have a long history of database support. Moving on to DUNE, the expectation is that another DOE lab like BNL would share database operations with Fermilab.

5.10 Development Plans

There are a number of database-related projects where R&D is needed or underway, utilizing effort and expertise from both DOE labs and universities. Recent effort (about 3 postdoc FTEs for 3 years) has been added through DUNE computing-focused DOE support. Coordination is provided by the Database group and attempts to balance the short-term needs of ProtoDUNE with the longer term needs of DUNE.

5.10.1 Conditions Database Development

The conditions DB is the primary interface to conditions metadata for all DUNE distributed computing resources and this task includes designing and developing caching strategies. The DUNE Database group will collaborate with the HSF to benefit from the experience of other HEP experiments, as well as the experience gained from ProtoDUNE, to develop a robust system while optimizing development effort. The development for the conditions DB is estimated to be 6 FTE years with that effort largely flat as it is spread over the following sub-tasks.

5.10.1.1 Conditions Database Core Design

The HSF conditions DBs group identifies the following points as key features of a good conditions DB design:

- Loose coupling between client and server using RESTful interfaces
- The ability to cache queries as well as payloads
- Separation of payload queries from metadata queries

These guiding design principles are likely to remain valid throughout the lifetime of DUNE, while implementation will need to evolve with technology.

5.10.1.2 Conditions Database and Software Framework

The interface between the conditions DB and the Software Framework is defined by the API of the conditions DB. The HSF Conditions Database group is in the process of defining a generic API, again using the experience of existing HEP experiments to understand best practice for supporting all read and write use cases. DUNE will collaborate with the HSF on this API definition and share experience on implementations. It is noted that, in line with the HSF, given the heterogeneity of compute hardware that DUNE needs to use, a common implementation is considered to be less important than a common API and sharing experience.

[4]The assumption is that offline database services will not require 7 days a week coverage.

5.10.1.3 CDB Service Robustness and Distributed Computing

In addition to the cache-friendly design of the central conditions DB service itself, a major issue for a critical (for offline) service like the conditions DB is resilience such that a robust and reliable service can be provided. DUNE plans to take advantage of synergies with other DOE labs, particularly at BNL which hosts conditions DB services for Belle II and other experiments. The backend database technology (Postgres) is common between Fermilab and BNL, and generally conditions DB services, under the umbrella of the HSF, are expected to evolve to look more and more similar. DUNE will therefore be well-positioned to benefit from progress in understanding multi-site database deployments assuming that these favorable circumstances encourage the necessary investigations. This could potentially include having a first class backup service that would support writing as well as reading functionality, in addition to the existing redundancy of read functionality via intelligent use of caches at sites.

5.10.1.4 Database Access for High-Performance Computing Facilities

As DUNE will utilize HPC facilities for some analysis tasks it is important that DB access is manageable when tens, or hundreds, of thousands of processes are distributed across an HPC cluster. Studies on how scaling on such systems can be handled without overwhelming the conditions DB when an enormous number of simultaneous queries are required. Again, DUNE plans to leverage the experience of other HEP experiments and contribute back to the common tools for using HPCs at scale.

5.10.2 Conditions Metadata Format and Serialization

Similarly to the discussion in chapter 8, the data format and serialization of conditions metadata deserves careful attention to avoid problems including technology lock-in. The need to interact with many different groups, essentially all providers of conditions (and other persistent) metadata, is the main driver of the amount of effort required for this task, estimated to be 4 FTE years with that effort concentrated in FY2024-FY2026.

5.10.3 Slow Control Database Development

The SCADA system chosen for Run I and II of the ProtoDUNE experiment was WinCC with Oracle as the back-end DB. This system is well tested at scale and it is relatively trivial to transfer information from the Oracle DB to a Postgres DB located at Fermilab. As discussed earlier, for DUNE it would be beneficial to use a solution that enables a Postgres back-end and recent developments allow WinCC to use Postgres (and other) backends. The SCADA system for DUNE, which at the time of writing has not been decided, may also bring new challenges Direct collaboration is envisaged here in order to prevent a disconnect between the SCADA system, which will generate massive amounts of data, and the subset needed for offline data processing. The development effort from the Database group is estimated to be 2 FTE years with that effort front-loaded to work with DUNE SCADA experts.

5.10.4 Hardware Database Development

The core functionality of the Hardware Database is described in section 5.8 and the first version is expected to be complete in FY22. Experience gained as the consortia use the database are expected to motivate feature requests, many of which will require urgent attention to support the construction phase of DUNE. Looking further ahead, some attention will be needed to ensure the maintenance of this crucial database for the long lifetime of the collaboration. Some information in the Hardware Database

will correspond to Conditions Metadata (calibrations), thus requiring the transfer of data while retaining provenance. The development for the Hardware DB is estimated to be 2 FTE years with that effort concentrated in FY2022-FY-2023.

5.10.5 Run Configuration Database Development

Transfer of Run Configuration information for offline use was a major problem for ProtoDUNE Run I, and significant effort will be used to improve the situation for Run II. The evolution of both the DUNE conditions DB and DAQ system could imply a similar amount of effort moving to DUNE. The development for the Run Configuration DB is estimated to be 1 FTE year with that effort concentrated in FY2022 and again in FY2026.

5.10.6 Other Database Development

The DUNE Calibration system is expected to generate very large datasets that will be processed to create conditions metadata that needs to be transferred to the conditions DB. Meanwhile the Beam conditions database stores far more data than needed for the data processing Conditions Metadata, and effort will be needed to find efficient solutions for data reduction. Finally, the Data Quality Monitoring Database is under the control of the DQM group, with some amount of data needing to be transferred to the conditions DB. All of these connections could imply additional development work which, if arising from offline-only constraints, should be supported by Database group effort. The development effort here is estimated to be 1 FTE year with that effort responding to needs as they appear and hence having a flat profile.

5.10.7 Database Access Tool Development and Documentation

Guided by use cases, tools will be developed for all of the DUNE Databases to enable direct access for studies. Most offline DB access will be made to the Conditions DB via the DUNE software framework, transparently to users, but there will be cases where expert users will need direct access to the various databases.

Documentation is another critical aspect of the database system. This includes descriptions of the DB system components and training materials, that will need to be updated as development work proceeds. The number of use cases, databases and groups involved drive the significant amount of effort required here. The effort for tool development and documentation is estimated to be 2 FTE years with that effort spread over time as needs arise.

5.10.8 Person Power Estimates

The personnel needs will be largely front-loaded as the database systems are researched, implemented, and tested. The database requiring the most effort will be the conditions DB, with significant effort required for the data format, other databases, and tools.

Part III

Global Computing Model

Chapter 6

Data and Processing Volume Estimates

6.1 Introduction

To understand the resources needed for Deep Underground Neutrino Experiment (DUNE) Computing and the development needed to utilize those resources, we start with an estimate of the data volumes and CPU needs from bottom-up estimates. In this chapter, we describe the assumptions that go into the estimates of data volumes and describe possible methods of reducing the total volumes while retaining physics capabilities. These assumptions have been coded into a python-based model and are updated frequently based upon changes to the far detector (FD) and near detector (ND) designs, and the physics requirements.

DUNE's detectors will produce information from a variety of technologies. We anticipate that raw data volumes will be dominated by the digitized waveforms from the liquid argon (LAr) detectors, and to a lesser extent from the photon detectors (PDs). Liquid argon time-projection chambers (LArTPCs) read out over long time windows, while the PD detectors read out only above threshold. We find that the LArTPC information dominates at the raw data level and drives the total data volume.

The data acquisition (DAQ) system can reduce these raw data volumes by several means, including:

- short readout windows tailored to one drift time;
- triggered readout of particular time slices;
- triggered readout of specific detector sub-regions;
- lossless compression;
- lossy zero suppression; and/or
- hardware pattern recognition.

Overall, we assume that the above methods can reduce data volumes from the hundreds of exabytes that would be produced by continuous readout to a manageable 30 PB/year. For beam and calibration events our assumption is that readouts of LArTPC detectors will generally be confined to a single drift time window. Supernova neutrino burst (SNB) readouts can last up to 100 s and would consist of up to

40,000 drift time units. The exact time segmentation for a SNB readout has not yet been determined but is limited by reasonable file sizes to much less than the full readout.

6.2 ProtoDUNE Experience

Our estimates are largely based on our experience with the ProtoDUNEs that ran at the European Laboratory for Particle Physics (CERN) in 2018 (ProtoDUNE-SP) and 2019 (ProtoDUNE-DP).

The ProtoDUNE-SP detector, which used single-phase (SP) horizontal drift technology (FD1-HD) technology, was read out by six anode plane assemblies (APAs) and a mix of PDs. The corresponding first FD module will have 150 anode plane assemblies and PDs based on the ARAPUCA technology. The second FD module will use SP vertical drift technology (FD2-VD) and will share some technology with ProtoDUNE-DP. Table 6.1 summarizes the parameters from the ProtoDUNE-SP experience that have been used to predict data volumes for the FD. The memory footprint per APA (charge-readout plane (CRP) for FD2-VD) was estimated from the algorithms and frameworks available for ProtoDUNE. Work is ongoing to process the raw data separately for each APA (CRP) so that the memory usage is not needed for all APA (CRPs) simultaneously, but instead the work can be serialized or parallelized.

6.2.1 ProtoDUNE Single Phase Experience

The ProtoDUNE-SP data have been processed through three reconstruction campaigns with the first operating just-in-time as data arrived, in "keep-up" mode. There have also been several ProtoDUNE-SP simulation campaigns. This work has resulted both in publications [14, 116, 17, 117] and in robust estimates of the computational characteristics of the data and processing. The characteristics of the ProtoDUNE-SP data are summarized in Table 6.1. For example, the uncompressed SP raw data for a single ProtoDUNE-SP trigger record were observed to be around 178 MB in size, which is the amount expected for the number of time projection chamber (TPC) channels read + a 20% overhead for other detectors and headers. Compressed SP raw data averages 70 MB, consistent with compression by a factor of 2.5.

6.2.2 ProtoDUNE Dual Phase Data

ProtoDUNE-DP recorded signals using two CRPs during the 2019 run. Observed data size without compression was 110 MB. In fall 2021, SP VD readout was tested in a smaller cold box and successfully reconstructed. Full ProtoDUNE tests of both HD and VD technologies with beam and cosmic rays are slated for 2022-24. The data volume estimates in Section 6.7 include data and simulation for this second ProtoDUNE test campaign.

6.3 Far Detector Data Volume Estimates

The raw data volume estimates presented here are based on the Spring 2022 FD and ND designs. This assumes that the first module will be FD1-HD, the second module vertical drift technology (FD2-VD), and a third and fourth module equivalent in terms of data volume to FD1-HD will be added in the mid 2030s. Estimates are provided for the Phase I and Phase II near detector (ND) designs. A separate discussion is provided for each early FD module type and the ND. The summary plots and table at the end include data and simulation for all detectors.

Table 6.1: Useful quantities for computing estimates for HD readout based on ProtoDUNE-SP experience. These numbers assume 12-bit readout. Hit reconstruction combines signal processing and hit finding.

Quantity	Value	Explanation
Number of APAs	6	
Number of channels/APA	2,560	
Readout time	3 ms	
# of time slices	6000	
Single APA readout	23 MB	Uncompressed estimate
Full detector readout	178 MB	Uncompressed real
Full detector readout	70 MB	Compressed real
Effective compression factor	2.5	
Beam rep. rate	4.5 Hz	Average
Hit reconstruction CPU time/APA	30 sec	from MC/ProtoDUNE
Pattern recognition CPU time/event	400 sec	from MC/ProtoDUNE
Simulation CPU time event	2,700 sec	from MC/ProtoDUNE
Memory footprint/APA	0.5-1GB	ProtoDUNE experience

6.3.1 Horizontal Drift

For the initial HD FD data volumes, we use our ProtoDUNE-SP experience and assume that raw data sizes and hit-finding CPU times scale with the number of APAs, while pattern recognition and simulation times scale with the number of interactions.

The DUNE Data Volume document [118] describes the expected event rates for various signatures in a FD module. These can be combined with the above numbers to provide the integrated data estimates shown in Table 6.3.

6.3.2 Far Detector Module with Vertical Drift Readout

Tables 6.4 and 6.5 summarize the expected data rates and volumes from physics signals of interest in a FD2-VD far detector module. The data volume corresponding to calibration events in the vertical drift module will likely be similar to the numbers shown for the horizontal drift in Table 6.3; a more detailed estimation is ongoing.

6.3.3 Far Detector Summary

Overall, bottom-up estimates yield data volumes of around 9.4 PB/year and 16 PB/year for each FD1-HD and FD2-VD module respectively. Lossless compression and restriction of the readout to geographical regions of interest should reduce this volume substantially. However, additional modules will increase these rates. A maximum rate of 30 PB/year of raw data across all modules and modes of operation has been specified and it is assumed that DAQ and detector configurations will change to meet this specification. We note that 30 PB/year is an average of 1.3 GB/sec, less than the rates already demonstrated for ProtoDUNE DAQ and storage. In principle, at 1 CPU-sec/MB of uncompressed input (from ProtoDUNE-SP experience), a few thousand cores (current FNAL - ∼11 HS06 [81] each) could keep

Table 6.2: Useful quantities for computing estimates for HD readout based on the DAQ requirements document of January 2022. CPU times are scaled from ProtoDUNE-SP assuming all detectors are used in data unpacking, signal processing and hit finding, but interactions are confined to a subsection of the detector not much larger than ProtoDUNE-SP.

Quantity	Value	Explanation
Far Detector Horizontal Drift		
APAs per module	150	DAQ spec.
TPC channels	384,000	DAQ spec.
TPC channel count per APA	2560	DAQ spec.
TPC ADC sampling time	512 ns	DAQ spec.
TPC ADC dynamic range	14 bits	DAQ spec.
FD module trigger record window	2.6 ms	DAQ spec.
Extended FD module trigger record window	100 s	DAQ spec.
Size of uncompressed trigger record	3.8 GB	DAQ spec.
Size of uncompressed extended trigger record	140 TB	DAQ spec.
Compression factor	TBD	
Beam rep. rate	0.83 Hz	Untriggered
Hit finding CPU time	4500 sec	from MC/ProtoDUNE
Pattern recognition CPU time	1500 sec	from MC/ProtoDUNE
Simulation time CPU time event	2700 sec	from MC/ProtoDUNE
Memory footprint/APA	0.5-1GB	ProtoDUNE experience

Table 6.3: Data sizes and rates for different processes in each horizontal drift detector module. Uncompressed data sizes are given. As readouts will be self-triggering, a 2.6 ms drift-window readout time is used instead of the 3 ms for the triggered ProtoDUNE-SP runs. We assume beam uptime of 50% and 100% uptime for non-beam science. These numbers are derived from References [119] and [118].

Process	Rate/module	size/instance	size/module/year
Beam event	41/day	3.8 GB	30 TB/year
Cosmic rays	4,500/day	3.8 GB	6.2 PB/year
Supernova trigger	1/month	140 TB	1.7 PB/year
Solar neutrinos	10,000/year	$\leq$3.8 GB	35 TB/year
Calibrations	2/year	750 TB	1.5 PB/year
Total			9.4 PB/year

Table 6.4: Useful quantities for computing estimates for FD2-VD readout based on the DAQ requirements document of January 2022. CPU times are scaled from ProtoDUNE-SP assuming all detectors are used in data unpacking, signal processing and hit finding, but interactions are confined to a subsection of the detector not much larger than ProtoDUNE-SP.

Quantity	Value	Explanation
Far Detector Vertical Drift		
CRPs per module	160	DAQ spec.
TPC channels	491,520	DAQ spec.
TPC channel count per CRP	3,072	DAQ spec.
TPC ADC sampling time	512 ns	DAQ spec.
TPC ADC dynamic range	14 bits	DAQ spec.
VD module trigger record window	4.25 ms	DAQ spec.
Extended FD module trigger record window	100 s	DAQ spec.
Size of uncompressed trigger record	8 GB	DAQ spec.
Size of uncompressed extended trigger record	180 TB	DAQ spec.
Compression factor	TBD	
Beam rep. rate	0.83 Hz	Untriggered
Hit finding CPU time	6,000 sec	from MC/ProtoDUNE
Pattern recognition CPU time per event	1,500 sec	from MC/ProtoDUNE
Simulation CPU time per event	2,700 sec	from MC/ProtoDUNE
Memory footprint/CRP	0.5-1GB	ProtoDUNE experience

Table 6.5: Data sizes and rates for different processes in a far detector module based on vertical drift technology. Uncompressed data sizes are given. As readouts will be self-triggering, an extended 4.25 ms readout window is used. We assume beam uptime of 50% and 100% uptime for non-beam science. The interaction/readout rates are derived from references [120] and [118].

Process	Rate/module	event size	size/module/year
Beam event	41/day	8 GB	63 TB/year
Cosmic rays	4,500/day	8 GB	12.5 PB/year
Supernova trigger	1/month	180 TB	2 PB/year
Solar neutrinos	10,000/year	46 TB/year	
Calibrations	2/year		1.5 PB/year
Total			16 PB/year

up with these data rates, but this throughput must be maintained over many years. In addition, SNB candidates may produce bursts requiring much higher DAQ and processing rates.

6.4 Near Detector Data Volumes

This section is based on the estimates provided in the near detector (ND) CDR [5]. Abbreviated descriptions of the ND components are given in this document in Section 1.4, along with the detector parameters that drive the annual data estimates. Summaries of the simulation and reconstruction workflows for the ND components are given in Sections 3.5 and 3.9, respectively.

The expected annual data sizes from the ND are summarized in Table 6.6. Due to the much higher data density in the ND as compared with the FD, CPU times/beam spill are expected to be much higher and are estimated to be 300 CPU/sec/spill using current processors and the 1.2 MW beam; i.e., a total of 1.5×10^7 spills/year. Simulated data samples will need to be an order of magnitude larger and thus require at least 10 times the CPU power. This leads to a rough estimate of CPU needed for ND reconstruction and simulation of approximately 3,000 core-years/year. Replacement of The Muon Spectrometer (TMS) with ND-GAr in Phase II, along with an increase in the detector occupancy for all three detectors, is expected to increase the required CPU per spill. Table 6.7 accounts for the change in the detector configuration from Phase I to Phase II but not the effect of occupancy, which can be mitigated by improvements to reconstruction algorithms.

6.5 Data Retention Assumptions

The Computing Consortium has made the decision to require all data located on DUNE storage elements to be registered within the Data Management system and have a data retention policy associated with each file. The goal is to assure that there is no "dark" data occupying storage capacity and that these data retention policies will help minimize total resource needs and our ability to recycle/reuse storage media. A short summary of the currently planned retention policy is listed in Table 6.8.

6.6 Data Tiers and Flow

6.6.1 Data Tiers

As the data from the DUNE detectors and simulation systems are processed, they evolve through a series of data tiers. The data tier concept was first adopted by the D0 and CDF collaborations two decades ago and is an important component in the definition of datasets in the data catalog. The algorithms which are run against data, transform it from one data tier to another in what is effectively a directed graph. This allows deterministic workflows to be constructed in a modular fashion and for dependencies in the data representation to be identified. This also allows for different branches and paths to be taken logically in the data lifecycle and for the overall computing model to manage these transforms that advance the state of the data in much the same manner as one would expect from a simple state machine.

Most importantly, this concept of data tiers allows for different information and informational representations to be present in each tier. This allows easier management and storage of data, (as down

Table 6.6: Annual DUNE ND raw data volume estimates. No compression is assumed. TMS is assumed to be installed in DUNE Phase I while ND-GAr is assumed to be installed in DUNE Phase II.

Type	Volume/year
ND-LAr	
In-spill data	144 TB
Out-of-spill cosmics	16 TB
Calibration	16 TB
Total	176 TB
TMS (DUNE Phase I)	
In-spill data	2 TB
Out-of-spill cosmics	8 TB
Calibration	1 TB
Total	11 TB
ND-GAr (DUNE Phase II)	
In-spill data	52 TB
Out-of-spill cosmics	10 TB
Calibration	6 TB
Total	68 TB
System for on-Axis Neutrino Detection (SAND)	
In-spill data	40 TB
Out-of-spill cosmics	8 TB
Calibration	1 TB
Total	49 TB
Total ND (Phase I)	**236 TB**
Total ND (Phase II)	**293 TB**

Table 6.7: Preliminary CPU estimates per event for the DUNE near detector components, in seconds.

Type	time/event
LArTPC	
Monte Carlo gen+sim	100 s
Reconstruction	60 s
TMS (DUNE Phase I)	
Monte Carlo gen+sim	50 s
Reconstruction	5 s
ND-GAr (DUNE Phase II)	
Monte Carlo gen+sim	100 s
Reconstruction	10 s
SAND	
Monte Carlo gen+sim	100 s
Reconstruction	10 s

Table 6.8: Retention policies by data tier

Tier	Description	Tape copies	Lifetime	Disk Copies	Disk Lifetime
Raw	Physics data	2	indefinitely	1	1 year
Test	test and commissioning	1	6 months	1	6 months
Waveforms	processed waveforms	1	10 years	1	1 month
Reco	pattern recognition	1	10 years	2	2 years

selections can be applied at different tiers to reduce overall storage footprints) while still permitting specialized applications to be developed that require specific data products or representations be present in the data. This approach also allows for parallel treatments of simulation and real detector data, which is key to the analysis process.

Table 6.9: Example data tiers. We show the number of copies for each version (1 version/year) and the proposed lifetimes for each data type on tape and disk. The total volumes of data/year are discussed in Section 6.7.

Data Tier	Produced by	Copies	Disk Lifetime	Tape Lifetime
PDSP raw data	DAQ	2	few months	> 20years
PDSP full-reconstructed (data)	pattern recognition	1	2 years	5-10 years
PDSP detector-simulated (MC)	geant4+detsim	1	transient	5-10 years
PDSP hit-reconstructed (MC)	hit finding	1	transient	5-10 years
PDSP full-reconstructed (MC)	pattern recognition	1	2 years	5-10 years
PDSP root-tuple	data reduction	1	2-3 years	5-10 years
FD raw data	DAQ	2	few months	> 20years
FD hit-signalproc (data)	signalproc	1	–transient	–
FD full-reconstructed (data)	pattern recognition	1	2 years	5-10 years
FD hit-reconstructed (MC)	pattern-recognition	2	few months	> 20years
FD full-reconstructed (MC)	pattern recognition	1	2 years	5-10 years

This diversity in tiers can be seen by examining the projected estimates for the DUNE data and retention policies for each tier. Tables 6.9 and 6.10 summarize some of the more common formats for DUNE, along with their typical trigger sizes and expected retention schedules. Full volume estimates over the course of the experiment are given in Section 6.7. For more details on the processing flow for data tiers, see Figure 3.1. Examples of common data tiers are:

raw for raw detector data.

simulated simulated data from event generators with Geant4 simulation only. The geant4 step is computationally expensive so these data may be kept long-term.

detector-simulated geant4 simulation with additional detector response simulation. Different variants of, for example space charge effects, overlays and photon libraries are applied here.

hit-signalproc future tier for data that has had basic data preparation signal processing up through the deconvolution step performed. This stage of processing is well suited to HPC's so this data-tier

may be produces as an intermediate step.

hit-reconstructed for data that has had initial reconstruction of regions of interest. This step is currently not stored but may play in important role in future if hit finding and reconstruction are done on a different architecture than the full reconstruction algorithms.

full-reconstructed for both data and simulation. This tier has advanced reconstruction to identify higher-level physics objects and interaction regions.

root-tuple Small common root-tuples summarizing the outputs of the full-reconstructed tier. These would be inputs for final analysis, including columnar analysis.

reco-dst An intermediate stage between full-reconstructed and analysis tuples. As we are still in the algorithm development phase, we have not yet converged on this format which likely will be $<$10% of the reconstructed stage.

Table 6.10 shows the estimated event sizes and rates for the different data tiers. These are used to generate the much more detailed data volume predictions below.

Table 6.10: Summary of data sizes (in MB) and numbers, when running, for the 3 majors DUNE efforts. The Near Detector (ND) and ProtoDUNE need full detector simulation while the Far Detectors (FD) only need simulation in the region ($\sim 15\%$) around the interaction.

Data Tier	ProtoDUNE (MB)	DUNE FD (MB)	DUNE ND(MB)
Raw	70–140	3750–8000	10
Simulated Tiers	200-300		20-50
Reconstructed Data	35	175	20
Number of raw M-Events/year	10	2.2	25
Number of sim M-Events/year	10	10	100
Peak dates	2018-2025	2028-2040	2028-2040

6.6.2 Far Detector data flow

A possible FD workflow motivated by the ProtoDUNE-SP experience is described below. Data may be stored temporarily at intermediate steps, for example if the optimal site for processing moves from an HPC to a more typical batch processing center.

Raw data 3750–8000 MB for a single readout of a full module, is processed to produce:

hit-signalproc waveforms that may be stored temporarily and then be processed further to produce:

hit-reconstructed hit regions of interest. These are likely 100-200 MB/readout, both because of data reduction within a single waveform and the ability to exclude large regions of the detector far away from the interaction. Those hits are then used for calibration and higher level pattern recognition producing:

full-reconstructed which contains the hits and higher level tracking information. Those data are then used to produced much smaller data summaries for later analysis.

Figure 3.1 in Chapter 3.1 illustrates this process.

As the bulk of the far detector data will be calibration samples, we can anticipate a need to process and temporarily store the early steps in reconstruction multiple times soon after it is taken but the permanent store of reconstructed data for physics analysis will be orders of magnitude smaller. However, at this point we make a very conservative estimate that all raw data will be reconstructed and the results stored.

We assume that we will be performing a full reprocessing of all raw data once/year with limited new simulation campaigns on a similar cadence. Generally the 2-year retention policy for reconstructed samples assures that the most recent two versions are readily available while older versions would need to be staged from tape and only available for special use-cases.

The processed waveforms are of considerable interest for machine learning algorithms but constitute a very large data volume. Given limited disk resources, we will likely need to prioritize disk access to the full reconstructed data by physics analyzers, possibly with a subset of the waveform information. Full access to the waveforms will likely require reprocessing of samples from archival storage.

6.7 Model Studies for Data and CPU Needs

Given the above estimates we can estimate total disk and CPU needs every year. The July 2022 version of these numbers is documented in [119]. Parameters are entered via a JavaScript Object Notation (JSON) file and results generated using python scripts. Figure 6.1 and 6.2 shows the assumed number of events/year for each detector type.

Planned simulation event volumes (shown in Figure 6.2 are a factor of 2-4 times the raw data rate. We assume that the rate of triggers that are not useful is at least 50% even for ProtoDUNE and the ND, and, in the case of the FD, far larger. This implies a ratio of 4 to 8 between simulation and data. Discussions of strategies to optimally use the simulation data volume are given in Sections 3.3.1 and 3.7.

For downstream CPU and disk calculations, data and simulation are shown together and simulation tends to equal or dominate the resource needs due to a combination of event size, CPU time and number of events.

CPU and size/readout are drawn from the above estimates.

Accessing data from tape imposes large lead times due to competition for resources within and outside of DUNE. Our strategy is to archive all raw and reconstructed data and simulation on tape but to also retain raw data on disk for short periods (six months) for calibrations and reconstruction and to place several recent versions of reconstructed data and simulation on disk in both the US and Europe to optimize access times. This motivates the following retention strategy and the data placement strategy discussed in 9.

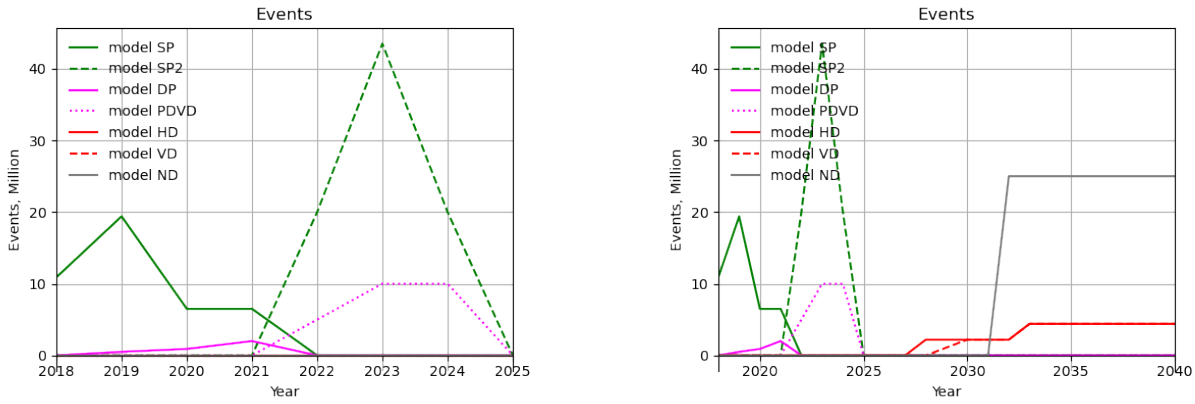


Figure 6.1: Numbers of raw data events from the detectors per year used in the model data volume estimates. Left is through 2025, right is the same through 2040.

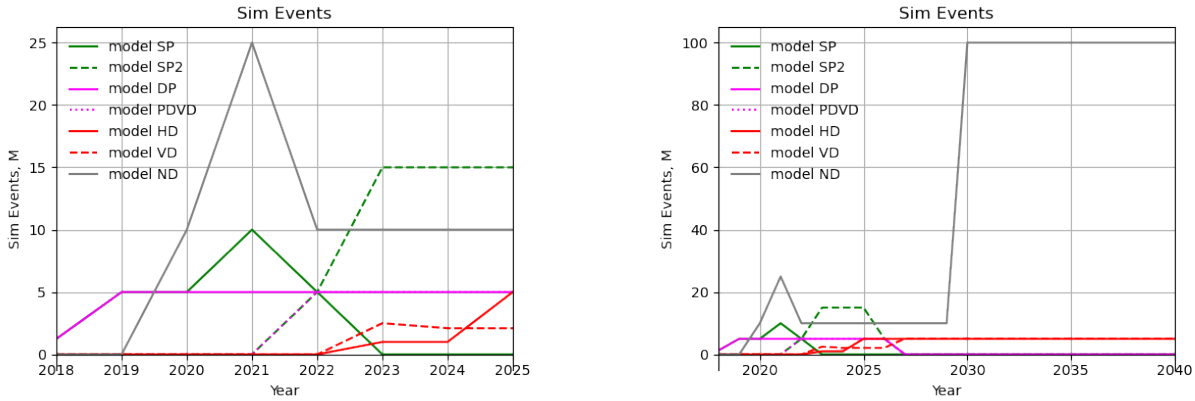


Figure 6.2: Numbers of simulated data events produced per year used in the model data volume estimates. Left is through 2025, right is the same through 2040.

- Two copies of raw data are retained indefinitely.
- DAQ commissioning data is marked "test" and one copy is retained on disk (and tape) for six months.
- Detector commissioning data is marked "study" and one copy is retained on disk for six months and on tape for up to five years, depending on physics needs.
- Reprocessing (likely just pattern recognition on processed hits) is performed on the full data sample once/year and two copies are retained on disk for two years. This ensures that all data have been processed with the most recent version of the reconstruction and that the two most recent versions are readily available on disk.
- Analysis-related CPU estimates include calibration. Current experience indicates that analysis efforts are equivalent in CPU utilization to reconstruction and simulation but produces smaller outputs.

Data lifetimes on tape are likely to be longer than the retention policy states but expired data may be overwritten or will not be migrated to new tape technologies.

Figures 6.3, 6.4,6.5,6.6, 6.7 and 6.8 illustrate the estimated storage and CPU needs through 2025 and 2040. In the early years, PD and ND prototype tests dominate while commissioning and operation of the first (and second) far detector modules and the ND become important after 2025. We include actual numbers for 2021. CPU and disk utilization were lower in 2021 due to not performing a fifth reconstruction pass that year and delays in distributing a second copy of reconstruction output to remote sites.

The estimates to 2040 are also shown compared to estimates from the CMS experiment[121] through the HL-LHC era. DUNE is estimated to use a few % of CMS's CPU needs, around 10% of CMS's disk and up to 20% of CMS's tape by the end of the 2030's.

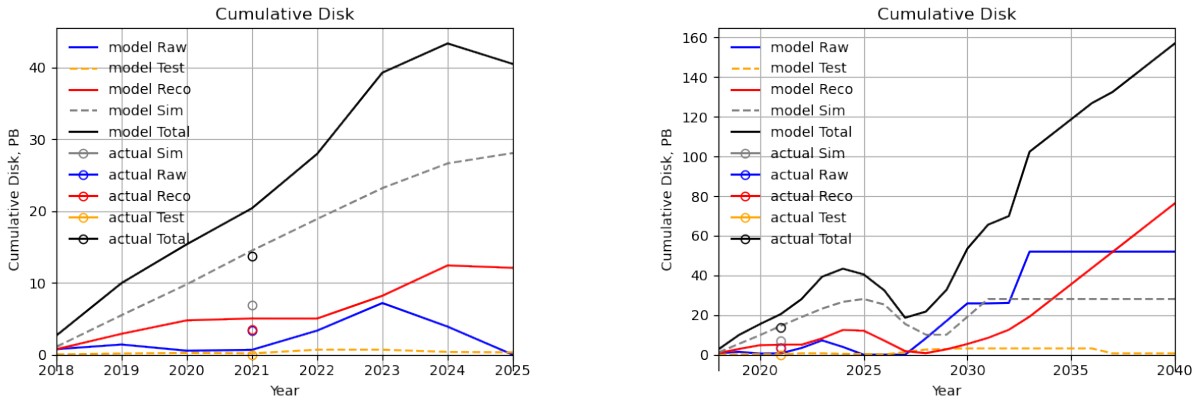


Figure 6.3: Estimated size of various disk samples in PB. This estimate includes retention policies and multiple copies. Left is through 2025, right is the same through 2040. The points show actual use in 2021 which was lower than planned due to delays in distributing second copies of samples to remote sites.

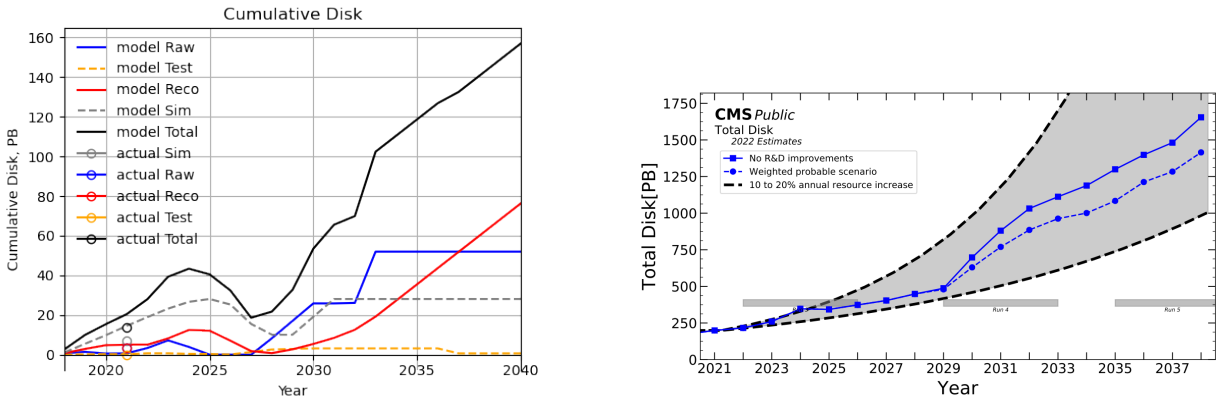


Figure 6.4: Estimated size of various disk samples in PB for DUNE and CMS at the HL-LHC for comparison. This estimate includes retention policies and multiple copies. The points show actual use in 2021 which was lower than planned due to delays in distributing second copies of samples to remote sites.

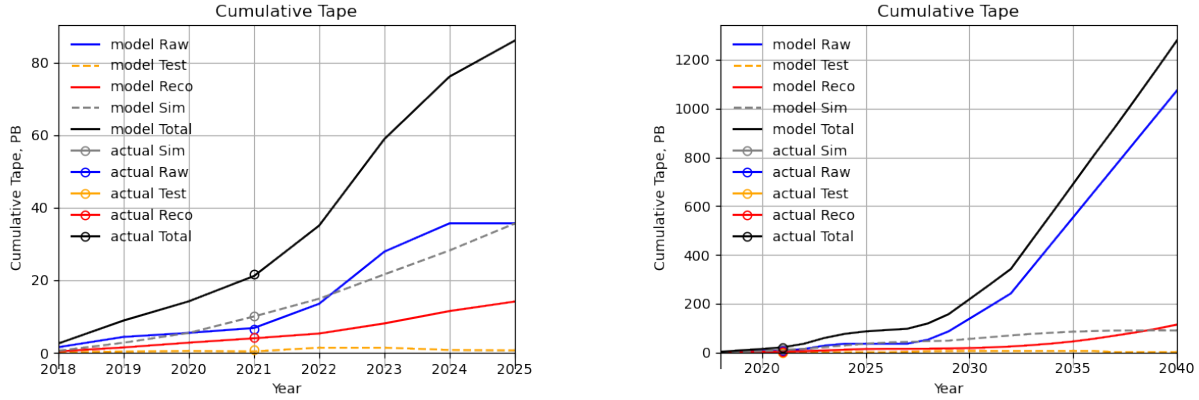


Figure 6.5: Estimated size of various DUNE tape samples in PB. This estimate includes retention policies and multiple copies. Left is through 2025, right is the same through 2040.

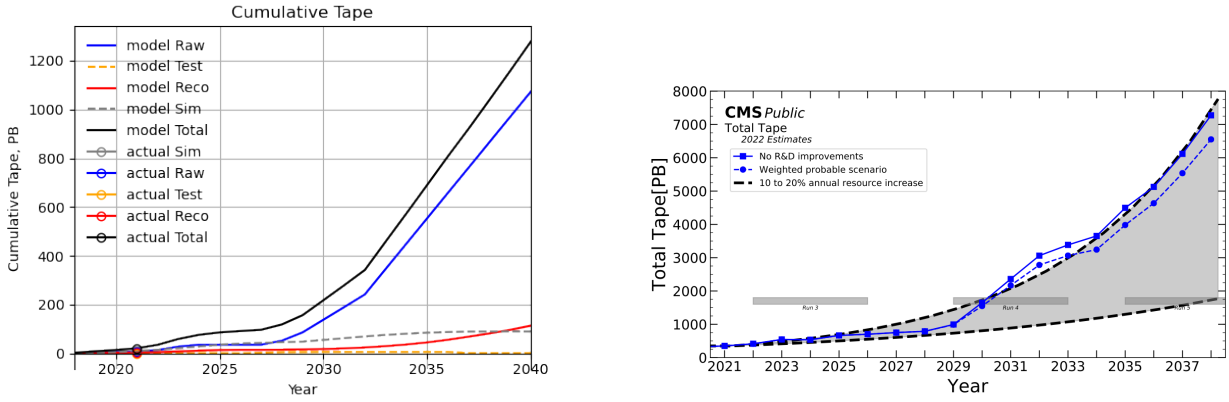


Figure 6.6: Estimated size of DUNE tape volumes compared to CMS HL-LHC estimates. This estimate includes retention policies and multiple copies.

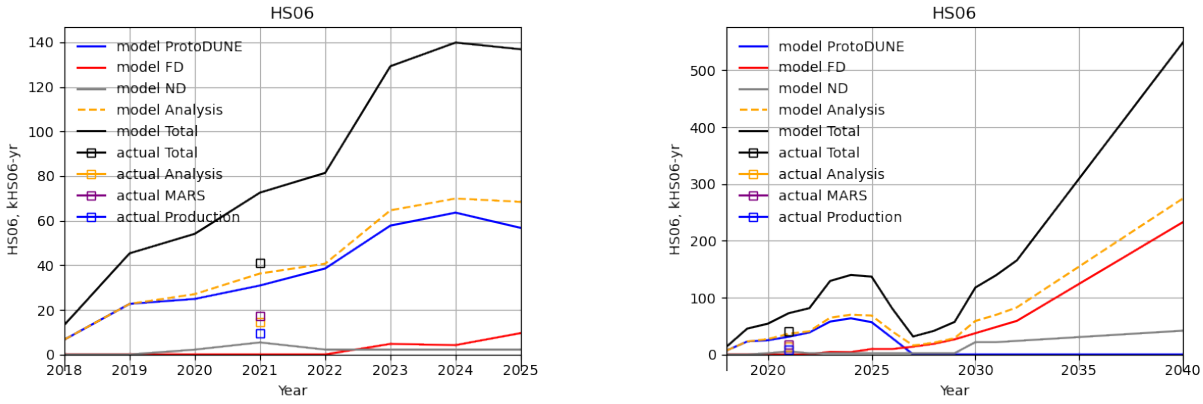


Figure 6.7: Estimated CPU needs for various samples. The units are kHS06-years (2020 vintage CPUs are ∼11 HS06 [81]) assuming 70% efficiency. Left is through 2025, right is the same through 2040. CPU utilization in 2021 was lower than the model due to the absence of a yearly reconstruction pass.

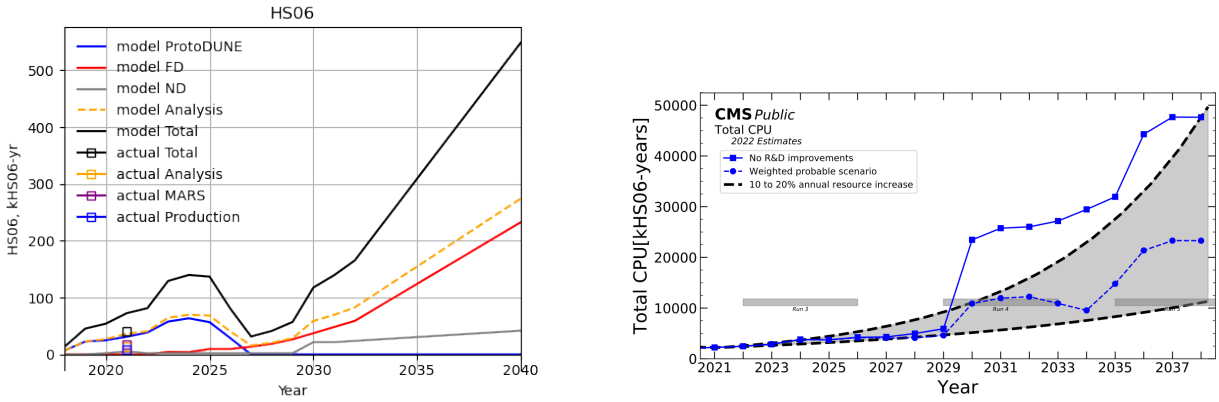


Figure 6.8: Estimated CPU needs for DUNE compared to CMS. The units are kHS06-years (2020 vintage CPU's are ∼11 HS06 [81] per core) assuming 70% efficiency. CPU utilization in 2021 was lower than the model due to the absence of a yearly reconstruction pass.

Chapter 7

Overview of the Computing Model

7.1 Introduction

The current DUNE global computing model is an organic extension of the Fermi National Accelerator Laboratory (Fermilab) Fabric for Intensity Frontier Experiments (FIFE) [122] computing model, used for smaller Intensity Frontier experiments, to the full global DUNE collaboration. This effort has relied heavily on global infrastructure such as Open Science Grid (OSG) and the Worldwide LHC Computing Grid (WLCG) and was successful for the first small-scale ProtoDUNE tests. However, it needs substantial enhancement to cope with anticipated data and processing volumes. This chapter describes the current situation and proposals for future improvements.

7.1.1 Global Resources

DUNE is a global collaboration with contributions from institutions worldwide. The long-term strategy for computing resources is for primary raw data storage to reside at the large host labs (the European Laboratory for Particle Physics (CERN) and Fermilab) and other national class facilities, with computing resources such as CPU and storage largely contributed by collaborating institutions. CPU contributions are provided by a large number of sites worldwide as shown in Figure 7.1, while storage is concentrated at a few larger sites. Table 7.1 shows the distribution of disk space pledges for 2021 and 2022, while Tables 7.2 and 7.3 list the sites contributing CPU resources.

Requests for resource pledges are made early in the calendar year, with pledges made before April for a computing year based on the UK fiscal year of April 1-March 31. This process is described in more detail in Section 2.3. The actual need can change over time, especially due to delays in experimental running and validation of new codes for production. Our current experience is that Central Processing Unit (CPU) needs have been met but storage pledges fall below the estimated need. For storage, shortfalls in pledges relative to requests are addressed by reducing the lifetime of second copies of some samples. For CPU, most usage is currently opportunistic and is running below the pledged levels. Chapter 6 gives a comparison of actual use in 2021 to the model used to make these requests.

Computational resources are currently dominated by conventional UNIX batch systems accessible via

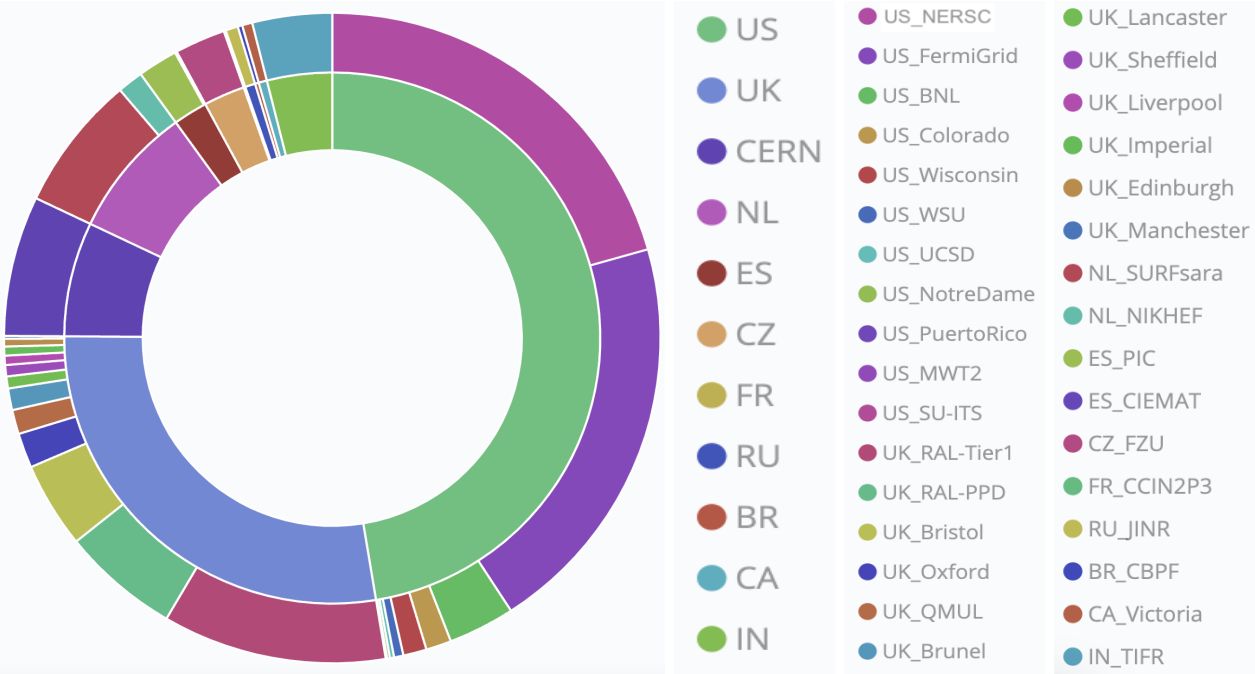


Figure 7.1: Distribution of wall time for DUNE production jobs, October 2021 to March 2022. Inner ring: country. Outer ring: site. Over 50% of wall hours have come from outside the United States in Fiscal Year 2022.

Table 7.1: Disk pledges in PB for 2021 and 2022.

Country/Lab	Name	2021	2022
Fermilab	Fermilab	2.2	7.6
CERN	CERN	2.2	3.0
Brookhaven National Laboratory (BNL)	BNL	0.5	0.5
United Kingdom	GridPP	4.0	4.0
France	CC-IN2P3	0.5	0.5
Spain	PIC Tier-1	0.5	0.72
Netherlands	NL/LHC Tier-1	1.9	1.8
Czechia	CZ-Prague-T2	0.3	1.0
India	TIFR	0.75	0.75
Russian Federation	JINR	-	0.5
Total pledge		12.85	18.97
Total request		20.4	27.3

Table 7.2: List of non-US international DUNE compute sites as of December 2021. Sites with substantial Rucio controlled disk are noted.

Site name	RC Site	Disk	Country
BR_CBPF	BR_CBPF		Brazil
BR_UNICAMP	BR_UNICAMP		Brazil
CA_Victoria	CA_Victoria		Canada
CERN	CERN-PROD	Yes	Switzerland
CH_UNIBE-LHEP	UNIBE-LHEP		Switzerland
CZ_FZU	FZU	Yes	Czechia
ES_CIEMAT	CIEMAT-LCG2		Spain
ES_PIC	pic	Yes	Spain
FR_CCIN2P3	IN2P3-CC	Yes	France
IN_TIFR	IN_TIFR	Yes	India
NL_NIKHEF	NIKHEF-ELPROD		Netherlands
NL_SURFsara	SURFsara	Yes	Netherlands
RU_JINR	JINR_CONDOR_CE	Yes	Russian Federation
UK_Bristol	UKI-SOUTHGRID-BRIS-HEP		United Kingdom
UK_Brunel	UKI-LT2-Brunel		United Kingdom
UK_Edinburgh	UKI-SCOTGRID-ECDF		United Kingdom
UK_Imperial	UKI-LT2-IC-HEP		United Kingdom
UK_Lancaster	UKI-NORTHGRID-LANCS-HEP	Yes	United Kingdom
UK_Liverpool	UKI-NORTHGRID-LIV-HEP		United Kingdom
UK_Manchester	UKI-NORTHGRID-MAN-HEP	Yes	United Kingdom
UK_Oxford	UKI-SOUTHGRID-OX-HEP		United Kingdom
UK_QMUL	UKI-LT2-QMUL	Yes	United Kingdom
UK_RAL-PPD	UKI-SOUTHGRID-RALPP		United Kingdom
UK_RAL-Tier1	RAL-LCG2	Yes	United Kingdom
UK_Sheffield	UKI-NORTHGRID-SHEF-HEP		United Kingdom

Table 7.3: List of US DUNE compute sites as of December 2021. Sites with substantial rucio controlled disk are noted.

US_UConn-HPC	UConn-HPC	Disk
US_BNL	BNL-SDCC-CE01	Yes
US_Caltech	CIT_CMS_T2	
US_Clemson	Clemson-Palmetto	
US_Colorado	UColorado_HEP	
US_Florida	UFlorida-HPC	
US_FNAL	GPGrid	Yes
US_KSU	BEOCAT-SLATE	
US_Lincoln	Rhino	
US_Michigan	AGLT2	
US_MIT	MIT_CMS	
US_MWT2	MWT2	
US_Nebraska	Nebraska	
US_NERSC	NERSC	
US_NMSU-DISCOVERY	SLATE_US_NMSU_DISCOVERY	
US_NotreDame	NWICG_NDCMS	
US_Omaha	Crane	
US_PuertoRico	UPRM-CMS	
US_SU-ITS	SU-ITS-CE2	
US_UChicago	MWT2	
US_UCSD	UCSDT2	
US_Wisconsin	GLOW	
US_WSU	WSU - GRID_ce2	

the OSG and WLCG but HPC resource such as NERSC and commercial resources such as GPUs from Google Cloud[80] are also being tested and used when needed and available. These systems are believed to be well-suited to many DUNE workflows, but are very diverse in the hardware offered and require significant effort to access, see Section 7.6.3 for further discussion.

Deep Underground Neutrino Experiment (DUNE)'s model is flatter and more network oriented than the original model for LHC experiments. As a result, it relies mainly on well connected data centers for storage and CPU provision. To date, we have found that most users prefer to do their work through the large national facilities available to them rather than building and using small local clusters at their home institutions.

7.2 Current Performance

DUNE has performed multiple simulation and reconstruction passes on the ProtoDUNE-SP data and is running significant simulation campaigns for the far detector (FD) and near detector (ND) design and physics studies. The Production Operations Management System (POMS) and sequential access via metadata (SAM) described in Chapters 11 and 13, are highly instrumented and allow assessments of the performance of the global computing system in near-real time. There are four major data/CPU access patterns:

- Simulation requires little input (mainly beam flux files and photon libraries), has a large memory footprint, uses significant CPU resources and writes back a few large files.
- Hit reconstruction and pattern recognition both read in large files, have an intermediate memory footprint and use $\sim$10 sec/MB of input data.
- Data reduction reads the reconstructed data and produces small tuple outputs for further analysis. Reduction uses ~ 0.1 sec/MB of input data and is generally I/O limited.
- Data analysis consists of repeated access to smaller tuple outputs for calibration and parameter estimation.

Each of these use cases is best suited to a different combination of data/CPU resources and the global compute model should be able to allocate resources appropriately. Currently the default configuration for all High Throughput Computing (HTC) jobs is for data delivery to be handled through xrootd streaming. To understand the impact of this choice on efficiency, we have used the SAM instrumentation to measure the xrootd streaming performance for disk/CPU location combinations.

7.2.1 Implications for Data and Processing Placement

The current study indicates that for CPU-dominated applications, notably reconstruction of raw data and simulation, the relative location of CPU resources is not critical. Wall time/MB is similar regardless of location. For IO-dominated applications, proximity to the data is important. Here there is a trade-off between the availability of resources and the efficiency with which they can be used. Intra-US processing, especially for locations near the national laboratories, is highly efficient, as is processing within the UK. However, efficiency falls off as the CPU and disks become more separated.

There are additional constraints imposed on our use of each site at scales greater than that of a single

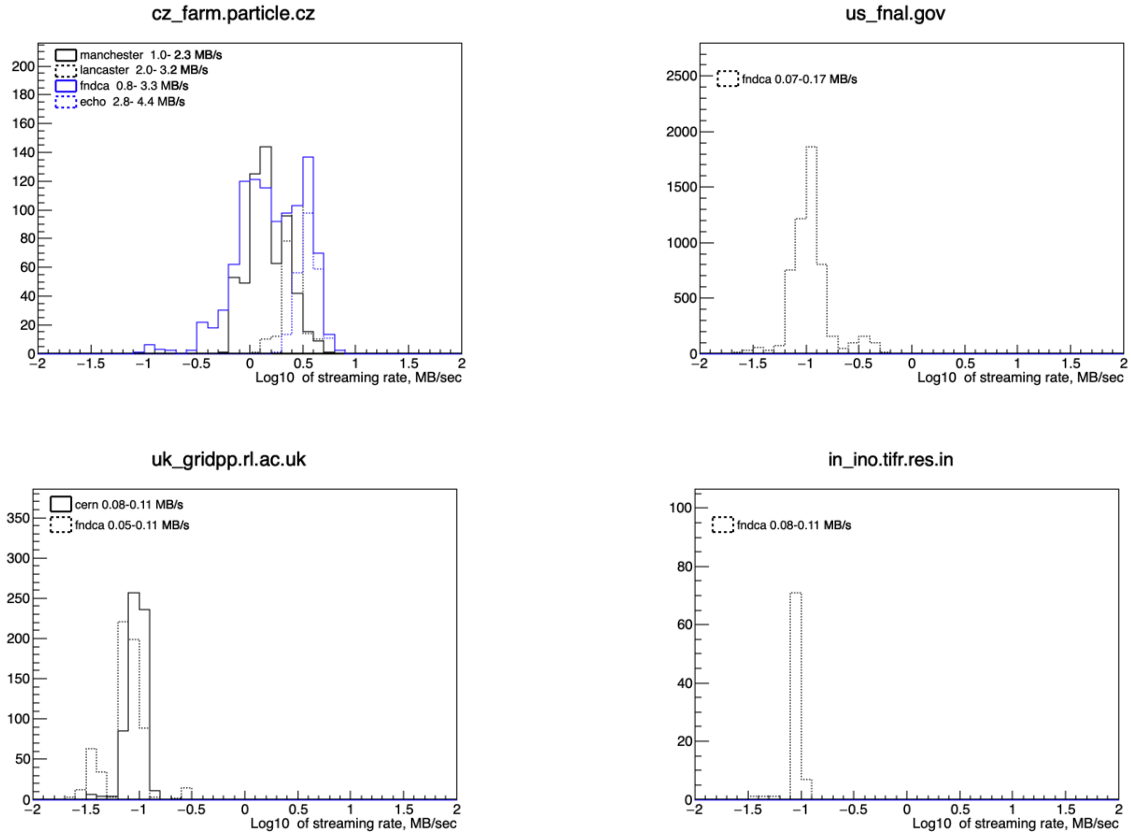


Figure 7.2: Streaming speeds for reconstruction jobs running at different locations. Raw data are stored at CERN and Fermilab. The histograms show the $\log_{10}$ of the inferred streaming rate (wall time/file size) for reconstruction jobs running at selected sites with different data sources.

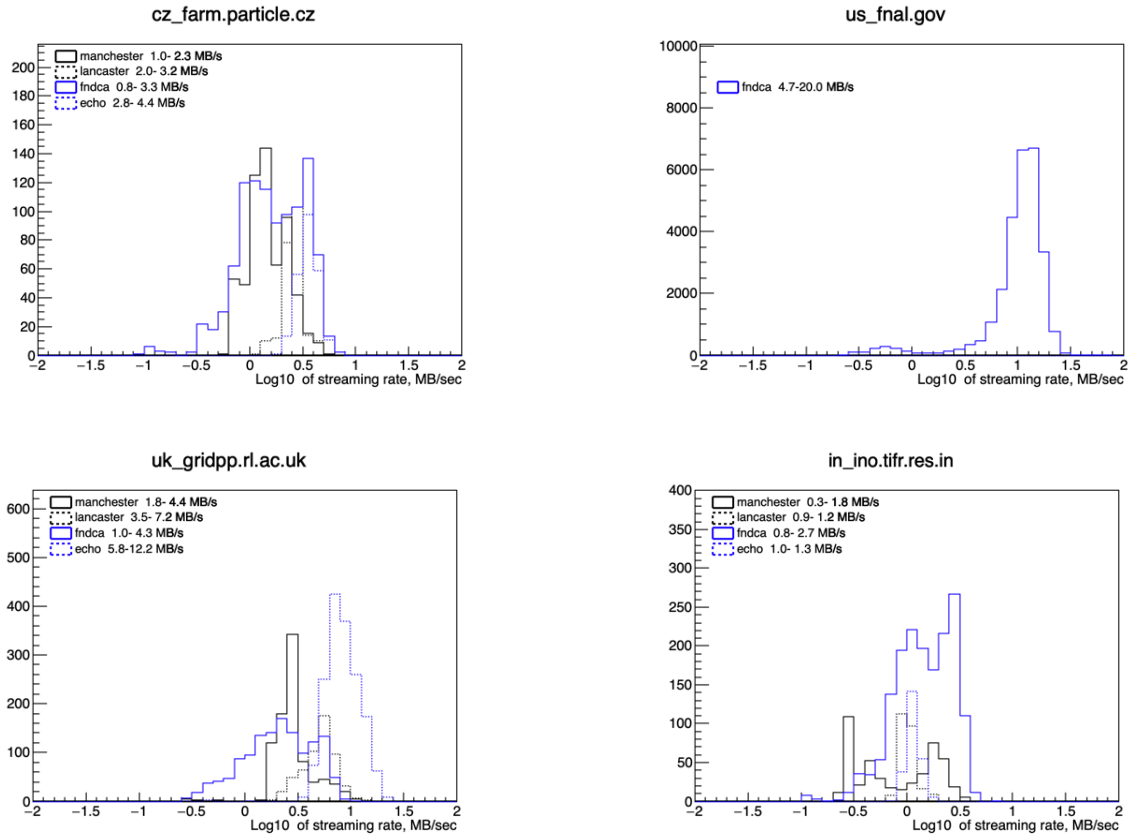


Figure 7.3: Streaming speeds for tuple creation jobs running at selected locations during a test in early January 2022. Reconstructed simulation files were located at sites in the UK and at Fermilab. The histograms show the $\log_{10}$ of the inferred streaming rate (wall time/file size) for tuple creation jobs running at selected sites with different data sources.

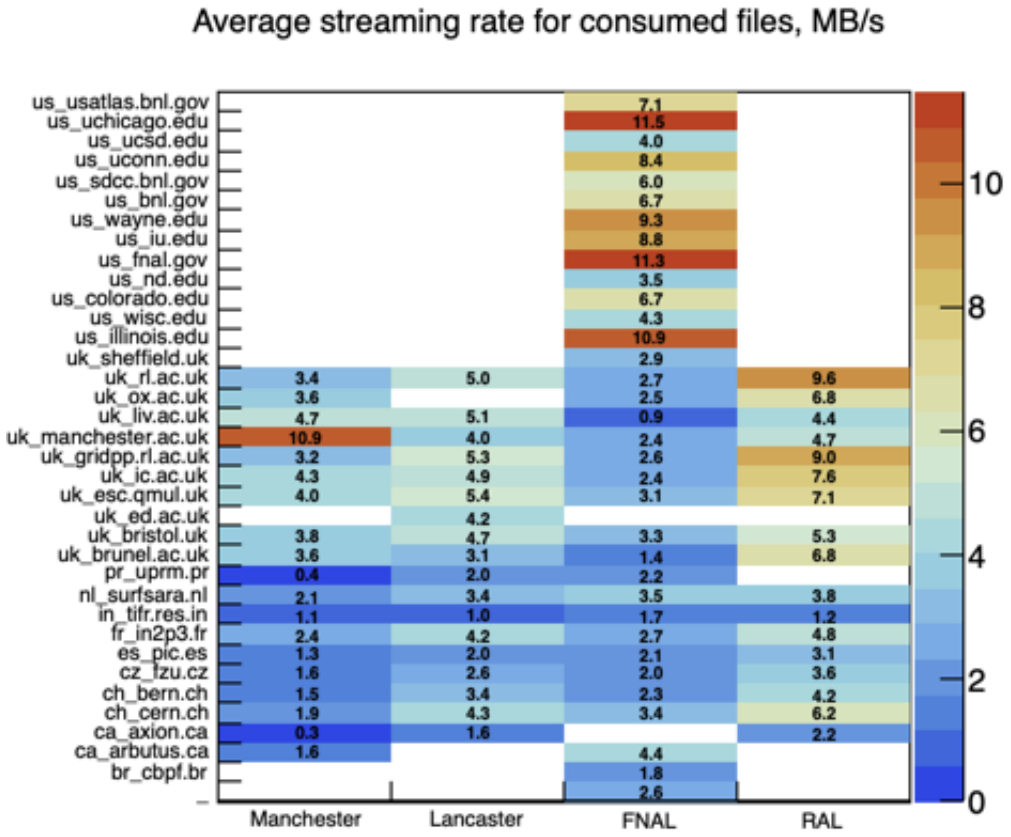


Figure 7.4: Streaming speeds for tuple creation jobs running at multiple locations during a test in early January 2022. Reconstructed simulation was stored at sites in the UK and at Fermilab. The average estimated streaming rates are plotted as a function of disk location (x-axis) and compute site (y-axis). Jobs in the US were required to use Fermilab disk but international sites were tested with multiple samples.

job. Due to local networking limitations, the number of jobs of a particular type we can run may need to be limited to avoid saturating the site's inbound or outbound capacity.

7.3 Design Philosophy

The large LHC experiments have historically relied on a tiered structure, with national Tier-1 centers and regional Tier-2 and Tier-3 centers. The DUNE model builds on the emergence of faster networks to move to a service-oriented model, where sites provide services – disk, CPU, real memory/core and archival tape – and projects are distributed to them based on their capabilities and available networking. For example, a site with large CPU and memory/core but slower networking would be ideal for simulation while small memory/core and fast access to large local disk stores would be ideal for high-level data analysis. In the long run, this model will require a high-level view of data locations and job placement with continual monitoring for bottlenecks, but allows new sites to contribute in an optimal way. The intent is to lower the bar for contributions without overburdening the core computing operations of the experiment. To date, we have successfully and quickly reconstructed and simulated the ProtoDUNE data using this flatter model and were able to improve I/O bound data analysis processing speeds at European sites by a factor of two by tuning the SAM data locality tables based on the measurements shown in Section 7.2.

We have the site information and monitoring tools to provide a workflow management system with the inputs it needs to do optimal placement of data and jobs to maximize efficiency. However our existing data management and workflow tools do not currently have a top-level request management overlay and are not able to take full advantage of system-wide information. This motivates development of an improved workflow model that scales to DUNE's world-wide site model, in addition to replacement of the existing SAM functionality for simpler use cases.

In Chapters 11 and 13, we describe the present and future designs for data and workflow management. We are pursuing two complementary designs: (1) a Data Dispatcher replacement for the existing SAM file delivery system which retains the loose coupling between the job submission system and data management (Section 11.7) and (2) a new, more tightly coupled, workflow system (Chapter 13) which uses a global view of CPU and disk locality to further optimize efficient processing. The existing SAM system is used by multiple Intensity Frontier (IF) experiments for interactive and batch use, mainly within the FNAL/OSG computing ecosystem, and those experiments will also benefit from the Data Dispatcher SAM replacement. The new workflow system is being designed by DUNE-UK based on LHC experience and is intended to optimize processing efficiency based upon data location across the much wider DUNE computing landscape.

7.4 Sites and Services

This section sets out our "Sites and Services" model for using sites for DUNE computing tasks, including sites that participate in OSG or WLCG more generally. Since our requirements are not the same as the LHC experiments, this requires implementing both a distinct naming scheme in WLCG and mapping our scheme onto the WLCG tier model.

At this stage, DUNE has chosen to express its requirements in terms of services provided by sites. Each

site provides networking plus one or more approved DUNE services, which satisfy DUNE's minimum requirements for the service in terms of capacity, quality, and interfaces. This model does not rely on assumptions about how sites and federations of sites will be organized in the future, as the community evolves away from the strict WLCG tiers model towards data organization, management, and access (DOMA) and concepts such as data lakes.

DUNE expects to be able to access services using broadly the same set of APIs as WLCG (e.g., HTCondor-CE and xrootd) and by using common cloud APIs (e.g., OpenStack and S3). For this reason, sites may be operated using conventional grid technologies, on-premises cloud systems, or commercial cloud services (where cost-effective). Nevertheless, sites do appear in the DUNE computing model, as the atomic unit for operations activity. For example, staff at a site can receive and process tickets, and may be required to have a representative at an operations meeting with the technical knowledge to comment on issues as they arise.

In terms of workflows and data management, DUNE does not impose or require any hierarchy or grouping of sites, and assumes that, in general, data may flow between services at any two sites. That said, DUNE expects to use network proximity and bandwidth information to guide the efficient transfers of data between services. The details of the different services within the DUNE Computing Model are described in detail in Section 7.6.

7.5 Sites, Federations, and Countries

As well as sites, there are two more administrative concepts: federations and nations. Federations are borrowed from WLCG and represent one or more sites that together pledge a particular amount of capacity to DUNE and enter their pledges into a system such as Computing Resource Information System (CRIC). Sites may choose to organize themselves this way as it allows more flexibility in how pledges are met against a background of planned upgrades at sites, unplanned outages, etc.

Nations are represented directly or indirectly at the DUNE Computing Contributions Board (CCB), and consist of one or more federations. Broadly, nations map to funding bodies and are the entity reviewed when evaluating the level of contribution to computing capacity relative to their number of DUNE members.

7.6 Types of Service

We have identified eight classes of service on which we will put requirements and request capacity:

1. Network – Section 7.6.1
2. DUNE Computing Element (conventional) – Section 7.6.2
3. DUNE Computing Element (HPC) – Section 7.6.3
4. Data Cache – Section 7.6.4

5. DUNE Storage Element – Section 7.6.5
6. DUNE Data Archive – Section 7.6.6
7. Interactive Analysis and Build Facility – Section 7.6.7
8. DUNE Analysis Facility – Section 7.6.8

Note that each class of service may have varying levels of resources provided within a service, and the details of those levels are discussed in the following subsections.

7.6.1 Network

Networking is needed at all sites, with basic requirements including IPv4 and IPv6. All sites should be connected to the wide area network, typically via their National Research and Education Network (NREN), with sufficient capacity to handle the data I/O commensurate with their fraction of the workload. In practice this means at least 40 Gb/s for major sites with large amounts of storage (with 100 Gb/s becoming the normal expectation for a shared site such as a WLCG Tier-1) and at least 20 Gb/s for smaller CPU-only sites (but with 40-100 Gb/s becoming the norm in shared sites). Sites with less network capability than this will be restricted to CPU-limited activities such as simulation. Other DUNE-approved services may impose further requirements in terms of network capacity.

7.6.2 DUNE Computing Element (conventional)

A DUNE Computing Element is a service that provides access to conventional CPU resources.

We envisage three subclasses within the computing element services aimed at centrally managed data processing, at user or working group data analysis, and at detector simulation, with appropriate minimum standards for each case, involving the following criteria:

- the support level, in terms of whether tickets will be acted 24/7 or only during working hours; note that our data model envisions dual copies of most user-facing data samples and a distributed processing system that is reasonably robust against single node or site failure. Services such as databases need to be at sites with 24/7 support.
- the total number of logical processors across the service;
- the interface used to submit jobs or create virtual machines;
- the operating system version for grid capacity;
- memory and scratch disk space per processor (Note that DUNE computing, due to the large data objects, often requires a large memory footprint. Provision of high-memory resources is very valuable.);
- incoming and outgoing network capacity per processor; and
- for, I/O limited activities, access to DUNE Storage Elements or Data Caches (defined below), which allows data-intensive jobs to execute without an unacceptably low CPU efficiency.

7.6.3 High-Performance Computing Elements (HPCs)

High-Performance Computing facilities form a special class of compute elements. These are large multi-purpose national-scale systems, such as NERSC, that give access to large amounts of specialized, interconnected back-end hardware such as Graphical Processing Units (GPUs) or field programmable gate arrays (FPGAs). There are several hurdles that stand in the way of DUNE taking full advantage of these resources. An initial challenge is that the HPCs available to DUNE are generally not controlled by DUNE, or even HEP, with access subject to annual applications and restrictions. In addition to a mismatch between HEP experimental time scales and the center's annual allocation models, current high-performance computing (HPC) centers have limited WAN connections at the node level making it difficult to move 100's of TB of data into, or out of processes. To address some of these challenges, DUNE is in active dialog with NERSC, as a first step, to gain access to larger temporary storage and bandwidth to support large-scale DUNE processing in the late 2020's. Other large US and European centers, as well as commercial systems, are also being explored but, for the moment, each requires custom negotiations and code adaptions. In addition, DUNE is taking an active role in the design and alignment discussions as part of the Integrated Research Infrastructure Architecture Blueprint Activity [1] that will define the landscape for DOE HPC in the next decades.

Given the large compute resources available at these centers, and the match between DUNE's image-like data and HPC hardware, we anticipate making substantial use of these very diverse resources. To date, DUNE collaborators have made significant progress in exploring algorithms that utilize the capabilities of GPUs, but those research and development efforts have not yet been incorporated into the production offline workflows. It should be noted that DUNE analyzers have individually been taking advantage of opportunistic access to GPU resources on the OSG, local GPU computing clusters (e.g. Wilson Cluster at Fermilab), university resources, and individual allocations at some HPC sites. With the exception of the Google Cloud studies described in Section 3.13, DUNE has not made use of the full capability of HPC systems in production (e.g. via MPI or multi-node calculations), but we consider incorporating these algorithms an important part of the framework and workflow development in the future. The integration of GPU and accelerator-based algorithms, and matching those with the available resources as part our production workflows, will take substantial administrative and technical effort to achieve. Given that at this time GPU utilization is on an individual basis with myriad algorithms and is not part of a centrally-managed workflow, DUNE has not been making predictions within the current computing model about the amount of GPU resources that will be needed in the future. It is a near-term goal to begin estimating the GPU resource needs and the effort needed to integrate those resources as production algorithms and workflows start to incorporate GPU-based processing. At this time, DUNE has been able to meet the development and analysis needs of the collaboration with opportunistic access, limited NERSC allocations, and university-based hardware.

7.6.4 Data Cache

A Data Cache is local storage used temporarily by DUNE jobs to optimizes data movement into and out of compute and storage sites. A data cache service provides transient storage that is not managed by DUNE; it improves the access speed for data located on remote storage that is accessed multiple times. Technologies such as XCache, StashCache, and "Data Lake" proposals may be able to fulfill this role.

[1] An overview presented to the Open Science Grid Council can be found here https://indico.fnal.gov/event/52594/

A data cache is generally associated with CPU resources and requires:

- suitable networking,
- sufficient resiliency against transient problems in order to prevent jobs from failing and losing the outputs from the work they have already done.

DUNE may be able to achieve similar functionality using some of the storage that it does manage to create temporary copies of data files, but this will require further development of the Data Management system. Support for data caches in the DUNE computing model may still be beneficial, by bringing in resources from sites that are willing to operate a cache service but do not wish to support long-term managed storage for DUNE.

7.6.5 DUNE Storage Element

A DUNE storage element is a large storage element dedicated to DUNE and managed via Rucio. The concept of a DUNE Storage Element mirrors that of a DUNE Compute Element. It must be of sufficient capacity, measured in hundreds of TB or in PB, for the operational overheard to be worthwhile. It must have a suitable support level, depending on the degree to which its services are replicated elsewhere. There must be enough inbound and outbound networking capacity for global data placement operations, and for jobs to write data there or to consume the data already present. In particular, for I/O intensive applications, there must be a minimum amount of DUNE Compute Element capacity available nearby, on which DUNE jobs can access appropriate storage services without unacceptably low CPU efficiency. For some jobs, for example full reconstruction and simulation, almost all storage services are appropriate while for analysis jobs, the only appropriate storage may be co-located at the same site.

This formulation allows conventional grid sites to place CPU and disk storage in adjacent racks, and it also supports novel regional architectures such as data lakes with sufficient network capacity to link CPU and storage at different locations. At this stage of the project, DUNE does not want to prejudge what will be available at the start of FD data taking, and does not want to discourage the exploration of new and more efficient ways of providing resources.

7.6.6 DUNE Data Archive

Data archive is designed to be archival storage of DUNE raw data, simulation, and any data derived from those files that must be permanently stored. Traditionally and as foreseen with DUNE, this service has been provided by tape storage facilities. There are currently at least five institutions providing tape storage facilities with the largest and most important providers being Fermilab and CERN. The volume and access patterns for these services is an ongoing area of research and development to both keep up with modern technology and to understand how it impacts DUNE's ability to accomplish physics. A central part of our model is prepositioning the samples most likely to be accessed on DUNE managed disk for extended periods, preferably in two locations, with tape access reserved for production processing of raw data.

7.6.7 Interactive Analysis and Build Facility

Both Fermilab and CERN provide centrally managed interactive computing facilities for users to perform small-scale data analysis and algorithm development. Fermilab hosts 15 4-core DUNE General Purpose Virtual Machines (dunegpvms) dedicated to DUNE that run Scientific Linux 7 with ∼2-3 GB of memory/processor. CERN hosts the Linux Public Login User Service) (LXPLUS) facility with machines with similar characteristics running CENTOS variants. The CERN machines are shared across CERN experiments. These systems have access to local network attached disk and to the Pseudo Network File System (PNFS)(Fermilab) and EOS (EOS)/CERN Advanced STORage manager (CASTOR)/CERN Tape Archive (CTA)(CERN) storage systems respectively. The laboratory interactive systems and many institutional clusters access the DUNE computing environment through the CERN VM File System (CVMFS) and can access data via xrootd once authenticated to the DUNE virtual organization (VO).

The Fermilab system has around 700 DUNE user accounts of which about 300 have been active over the past six months. It is harder to get statistics on interactive DUNE activity at CERN; it is substantially smaller but non-zero.

Fermilab also provides two 16-core virtual machines dedicated to fast code builds.

7.6.8 Dedicated Analysis Facilities

Most user data analysis is currently being done using serial reads of Liquid Argon Software (LArSoft) outputs or small root ntuples on the generic interactive systems described in 7.6.7. DUNE is actively exploring using and provisioning dedicated user analysis facilities capable of columnar access to data and more sophisticated analysis techniques in addition to notebook based analysis.

7.6.8.1 Compute Canada Prototype

An interactive analysis facility has been prototyped on a cloud allocation provided by Compute Canada's OpenStack platform on the Arbutus Cloud (https://docs.computecanada.ca/wiki/Cloud_resources). This experimental allocation is dedicated to DUNE and currently includes 300 virtual CPUs, 2.2 TB of shared memory, 2 TB of disk storage, and 1 TB of CephFS storage that uses a shared storage protocol. More storage has been requested for the next resource allocation year (2022).

Several computing nodes, each with eight CPUs and 90 GB of memory, have been formed out of this allocation. The facility relies on Jupyterhub to implement a web-based interactive experience and uses containerization technologies to maintain a dedicated workspace for each authenticated user. Kubernetes is the industry-standard platform for this use case and can be deployed at scale for up to 5000 nodes[123]. A user authenticates through CILogon and a single-user JupyterLab server on a Scientific Linux 7 Docker image is quickly deployed thereafter.

The facility provides a few features to facilitate analysis. LArSoft and DUNE-specific libraries are enabled through CVMFS. Users building event loop-based analysis routines will have the identical experience to the bash-based virtual machines. Data files can be streamed with xrootd once users authenticate with the VOMS server.

7.6.8.2 Fermilab Elastic Analysis Facility

We are testing similar functionality at Fermilab on the Fermilab Elastic Analysis Facility. Among other interfaces, this facility hosts the Fermilab Analytics Hub, which provides access to DUNE code and data to Jupyter notebooks via CVMFS and xrootd. This system has already been used to test reconstruction algorithms and validate data from the ProtoDUNE cold boxes. For the future we are exploring modern analysis frameworks such as Columnar Object Framework For Effective Analysis (COFFEA).

7.6.8.3 Coffea

Coffea[124] is a new analysis framework that could enable analyses on the fly with a quick turnaround time and a low barrier for a new analyzer. The framework was originally developed by LHC collaborators to enable columnar-based analyses of large volumes of LHC data in flat ROOT format using Python. Coffea employs uproot for file I/O and uses a NanoEvent class to parse TTree branches into user-defined physics objects for columnar computation. Users have the option to distribute their computations across different CPUs using a Dask[125] cluster.

A prototype NanoEvent for the ProtoDUNE PDSPAnalyzer flat analysis format is already available with the goal of demonstrating a full-chain analysis (event selection, background tuning, systematics). Small-scale tests of data streaming from Fermilab using xrootd has also been successful.

Longer-term goals include token-based authentication to seamless bridge user login and grid authorization, understanding network limitations and developing means to circumvent them.

7.6.8.4 Analysis Facility Issues

One of the major issues for a dedicated analysis facility is the ability to access and maintain data samples. We expect that, with many different analysis users, many disparate data samples will need to be available at any given facility but they must also be on local storage to maintain optimal efficiency. We are designing user Rucio-controlled samples into our new systems, but do not yet have experience with them in production. The analysis-user-facing Rucio services will be need to be designed, developed, and integrated into the future analysis facilities. The expectation is that the majority of the effort will be the development of DUNE specific modules, scripts, and services within the analysis facility and framework, and that there will be limited development in Rucio to meet the needs of analysis workflows. DUNE expects to draw requirements and end-user analysis workflow examples from ProtoDUNE-HD.

Chapter 8

Data Formats

8.1 Data Format Overview

DUNE data are stored and processed in a variety of forms and representations which are tuned to the analysis that is performed on them. In general, the data are characterized and classified by their data tier (described in 6.6.1) and their data format within that tier. The data tier corresponds to the stage of the data in its life cycle and its progress from the original detector acquisition data and simulations, towards its final analysis stage. The data format refers to the low level representation and organization of the data as it takes on in either a persistent or transient form during the analysis workflows. This classification of data by tier and format allows Deep Underground Neutrino Experiment (DUNE) to develop, catalog, and maintain its data ecosystem in the context of the larger computing model and computing resources.

The specific data format of each data tier can have a large impact on the performance of processing and accessing that data tier. A review of data formats and their serialization performance with respect to LHC data, was performed based on the state of major data formats in 2017[126]. While the actual results have since been superseded by improvements in the formats they examined, their study highlighted the importance of the data format and access libraries to analysis turn-around times in HEP.

The low-level representation of the DUNE data is keyed to the data tier(s) at which they exist and are accessed. It is common for these representations to change over the data lifecycle to reflect organizational needs of the algorithms that are run on the data. In particular in HEP, it is common for raw data coming from the detector subsystems and DAQ systems to have representations that are closely tied to the electronics and readout (and is often customized due to optimizations that are needed for high speed readouts), while later stages of processing use more generic formats and representations more amenable to use in algorithms and transforms. In the HEP and neutrino community, there have been a number of different "high level" data representation and I/O systems that have been used. Starting in the late 1990s the ROOT I/O system became one of the dominant systems for HEP data, and has continued to feature prominently for experiments which make heavy use of C++ in their software stacks.

In the case of DUNE, we have historically represented our data through a combination of custom and ROOT I/O based data formats. In our initial ProtoDUNE run 1 data processing, and in our current simulation chains, this has been our strategy and is fully supported throughout our processing chain. ROOT I/O provides data objects and methods that have been optimized for certain types of HEP applications and access methodologies. The Liquid Argon Software (LArSoft) framework, which is discussed in section 4.2, is compatible with the ROOT I/O system and other data formats.

Many Machine Learning (ML) toolkits, which have been developed by the computational sciences and computational industries, do not use the ROOT I/O data formats and systems. These general toolkits have been successfully tested in DUNE for event classification and particle identification. Written in Python, these ML workflows read and write data in formats commonly used in other scientific communities, such as Hierarchical Data Format (HDF5) [127]. Not all workflows work in this manner. Some export image data from LArTPCs as image data in two-dimensional ROOT histograms, and even comma-separated value (CSV) format has been used as a communication format between *art*/LArSoft jobs and external ML tools. In the case of ROOT histograms, a DUNE-supplied read-in layer is provided.

DUNE is planning to use HDF5 storage format for the upcoming raw data from ProtoDUNE-HD, ProtoDUNE-VD and ICEBERG. Dedicated I/O modules have been written to read the HDF5-formatted data into an art/LArSoft job, and as of this writing, they are in use reading HD and VD coldbox data. The DUNE software stack also contains modules that export data from an art/LArSoft DUNE job in HDF5 format, so that they can be read by external AI/ML software.

Not all data used for AI/ML applications is in the HDF5 format. For example, section 3.13 describes a use of a convolutional neural network (CNN) in a LArSoft job in which data are exchanged via TCP/IP network communication between compute elements in a format dedicated for the purpose.

A common solution to the compatibility issue in HEP experiments is to use an interface layer that can mitigate the choice of persistent data format with some performance cost. The scikit-hep project[128] provides a python ecosystem to read ROOT persistent data formats and convert them to native data science formats.

The HDF5 format is designed to save data as organized tables of fundamental data types along with links between tables which then allow for complex record structures similar to modern databases. The strength of HDF5 is that its tabular structure allows efficient "columnar" analysis of data (i.e. analysis of one or more variables from across a dataset). The format also has native support for parallel data access, and in particular for parallel reading and writing of compressed data. This support extends to highly parallel file systems, such as those found in leadership computing facilities, and has been tuned to scale extremely well using the MPI protocol in these environments. For the large DUNE event data, which can be highly compressed, this support for parallel I/O is advantageous, especially in the near real-time data acquisition environment. As a result of these features, and the use of the format in other segments of DUNE, we expect to support the HDF5 format as part of our computing model.

In contrast to the HDF5 format, the ROOT I/O format has support for the recording and retrieval of complex data structures in the C++ language, and can be used to serialize and deserialize C++ structures and classes. It is naturally integrated with the ROOT data analysis framework which, while domain specific to HEP, is nonetheless one of the largest and most robust toolkits for HEP data analysis. We also expect that support for parallel I/O within the ROOT I/O systems will be expanded by the

time of the DUNE era, and that the parallel reading and writing of compressed data will be supported. Similarly we expect that in the DUNE era, new organization and representations of tree'd data structures (e.g. n-tuple or TTree like structures) will be available through the RNtuple system[106, 107]. It is expected that RNtuple will provide columnar data access and performance as good, and potentially better, than HDF5. Indeed, the LHC experiments are expected to migrate to the RNtuple format on the timescale of DUNE. As a result of these features of the format, and the use of the format across HEP, we expect to support the ROOT format as part of our computing model.

One concern that arose during our format evaluation was the availability of a streaming option for HDF5. Our large-scale offline processing is most efficient with a mix of both streaming and direct copies of input data. Streaming also allows any DUNE site to access centrally stored data transparently. For this reason HDF5 streaming is very desirable. We have successfully adapted the DUNE LArSoft framework to read local HDF5 input and very recently have demonstrated streaming via xrootd in test mode. However, due to increased latency, efficient streaming has to include caching strategies which have not yet been investigated and making HDF5 streaming work in production will require further, crucial developments.

In addition to the ROOT and HDF5 data formats, we expect that the DUNE data representations will need to be flexible and adapt to changing technologies and evolutions in the data science and physics communities. For this reason we expect to allow for the addition of other data representations and have placed requirements on the software frameworks used by DUNE, as discussed in chapter 4, to support multiple data formats and I/O layers, as well as requirements on the data management and cataloging systems to support multiple data formats.

Given the vast variation in scale and data access patterns that DUNE computing needs to support, it is likely that most if not all of these options, and additional as-yet-unknown options, will have a role in creating the most effective suite of solutions to support the physics program. Detailed studies of the ProtoDUNE I & II datasets will be needed to make cost-benefit analyses of the various options and guide the analysis model. A final consideration, and by no means the least important, will be the ease of use of the analysis model for physicists to maximize their analysis output.

Chapter 9

Data Placement

9.1 Current Status

DUNE relies on a multi-level data placement strategy that has grown out of the Fermilab fixed-target program with elements from CERN fixed target experiments. Figure 9.1 illustrates the storage available to users. Reference [129] provides a more detailed description. Access methods and catalogs are described in detail in Chapter 11.

There are two general use cases considered for storage systems in Deep Underground Neutrino Experiment (DUNE): production operations and analysis by end users. Production operations are in general focused on utilization of large-scale storage elements. End users, however, utilize a wide variety of storage systems and typically log into unix systems, at Fermilab, CERN, or home institutions, which have disk mounts of varying size and level of backup.

9.1.1 Fermilab Storage systems

At present, most users use the interactive resources at Fermilab and/or submit their grid jobs via the Fermilab systems so those systems are described in enhanced detail.

Local disk - There are small volumes on local physical disks, directly mounted on the machine hosting the virtual machine (VM) with direct links to the /dev/ location. These are mainly intended as temporary storage for infrastructure services (e.g., /var, /tmp). These areas must be used to store secure items such as user authentication and authorization credentials to avoid exposing them over the network. Such secure items are saved to local disk automatically with owner-read permission and other permissions disabled. These local areas are usually very small and should not be used for data file storage or for code development.

Network Attached Storage

Users have access to POSIX compliant network attached storage areas with varying sizes and levels of backup. Network attached disks are not safe for storage of certificates and tickets.

- Users are provided with a network mounted home area with $\sim$2 GB of quota. A valid Kerberos ticket is needed to access files in the home area. Periodic snapshots are taken and available to users in case there is a need to recover deleted files.
- A network attached storage (NAS) Application volume is intended for code development and is limited to 100 GB/user. The area has periodic snapshots to allow for recovery of unintentional deletion of files.
- Additional NAS data volumes provide storage for fast analysis and code testing interactively. The expectation is approximately 1 TB of storage per user will be sufficient. Any larger samples should be cataloged and managed by Rucio or by physics groups.

The NAS disks are not accessible to grid worker nodes.

dCache - Several PB of dCache[130] storage are available for multiple purposes. dCache storage is not fully POSIX compliant but network file system (NFS) mounts with limited functionality are available on Fermilab interactive machines. The dCache storage is readable and writable from grid machines via xrootd and transfer mechanisms such as Intensity Frontier Data Handling (ifdh). It is expected that this is the main storage element for the output of both production and analysis distributed computing jobs.

- The dCache scratch area is used mainly for files returning from grid jobs. The total volume of the dCache scratch volume is (2022) greater than 5 PB. It is currently shared with other experiments at FNAL. Files are automatically removed based upon a LFA policy with typical lifetimes of one 1 month.
- The dCache persistent volume is a ≈ 800 TB area for persistent storage of user files. Files must be removed by their owner. Half of this volume is currently controlled with physics group based quotas, while the other half will soon be transitioned to quota-controlled usage.
- The dCache tape-backed volume is a ≈ 3.2 PB disk cache sitting in front of the Enstore tape system. Files in this area are cataloged and tracked by the sequential access via metadata (SAM) system and, in future, by Rucio. Files that have been flushed from the tape-backed cache need to be prestaged before they can be used. At the moment, individual users can do this but we anticipate, once most samples have been moved to DUNE controlled disk, that users will need to request prestage to avoid drive contention.

Remote users also have similar access to local disk at European Laboratory for Particle Physics (CERN) and their own facilities and to collaboration samples located on the large DUNE Rucio Storage Elements (RSEs). Much of the dCache storage at Fermilab and at other DUNE sites is migrating to management via the Rucio data management system that is described in more detail in Chapter 11. A small fraction will remain managed by physics groups or directly available to general users.

Tape - At Fermilab, DUNE shares the Public instance (separate from a CMS instance) of the tape storage infrastructure. The physical system is currently (March 2022) composed of two IBM TS4500 tape libraries with LTO8 drives (72 total) with over 100 PB capacity per library. In addition there is a legacy Oracle SL8500 library being replaced with a 150 PB capacity library (vendor selection and purchase pending). Current DUNE utilization of the tape complex is 21 PB

of the total (non-CMS) active utilization of 168 PB. The tape storage software used is Enstore. Fermilab is actively pursuing a transition to using the CERN Tape Archive (CTA) software, and plans a multi-year transition.

Tape storage is also available at CERN through the CERN Advanced STORage manager (CASTOR) system. The CERN system is mainly used to archive the raw data from ProtoDUNE. We are also in the process of integrating tape archives in the UK and France for processed samples.

Distributed caching systems - DUNE makes use of the CERN VM File System (CVMFS) caching system to distribute code, flux and shower libraries and user grid executables. CVMFS mounts are available on DUNE grid nodes.

- `/cvmfs/` mounts of DUNE-specific code, shared executables such as Liquid Argon Software (LArSoft), and general utilities are distributed via CVMFS. Version control is provided by the Fermilab UNIX product support (UPS) system.
- StashCache a.k.a. XCache is used to deliver larger payloads such as flux files and shower libraries to grid jobs.
- A CVMFS-based `dropbox` is also available to transfer user executables to grid jobs.

9.1.2 Storage systems at CERN

Data at CERN are stored in the EOS (EOS), CASTOR, and CTA systems. Some of this storage is under the control of the DUNE Rucio system, including a full copy of the ProtoDUNE raw data in CASTOR/CTA. Transfers out of CERN to storage elements elsewhere are made using FTS3 and standard transfer protocols. DUNE is working to avoid Globus transfers in order to reduce cost and support effort. During ProtoDUNE runs, CERN provides a vital step in fast data transfer and the raw data on disk at CERN is very important for fast turnaround analysis. Along with archival storage, there is also substantial space for generic use by CERN-based users. The CVMFS file systems are mounted at CERN and local users have access to the CERN batch resources through the normal DUNE batch system which requires membership in the DUNE virtual organization (VO) and through normal CERN submission systems.

To date, most general users only use the CERN CPU and disk resources as they would use any other Worldwide LHC Computing Grid (WLCG) site. The expectation is that access to files on storage elements at CERN will be through xrootd transfers or streaming.

9.1.3 Collaboration disk stores

DUNE collaborating institutions also contribute very substantial storage resources. These resources are populated and managed via Rucio and the SAM catalog. As with the Fermilab dCache stores they are available via streaming (xRootD) and grid-copy mechanisms (non-Globus) but are not mounted on interactive nodes. These sites have come online over the past year (June 2021-2022) and contain copies of the most recent reconstructed and simulated samples. Figure 9.2 shows the distribution of disk storage across DUNE institutions as of July 2022. The status of storage elements from the DUNE Data Management side will be monitored using Rucio tools, while the local monitoring by sites will be the responsibility of site administration.

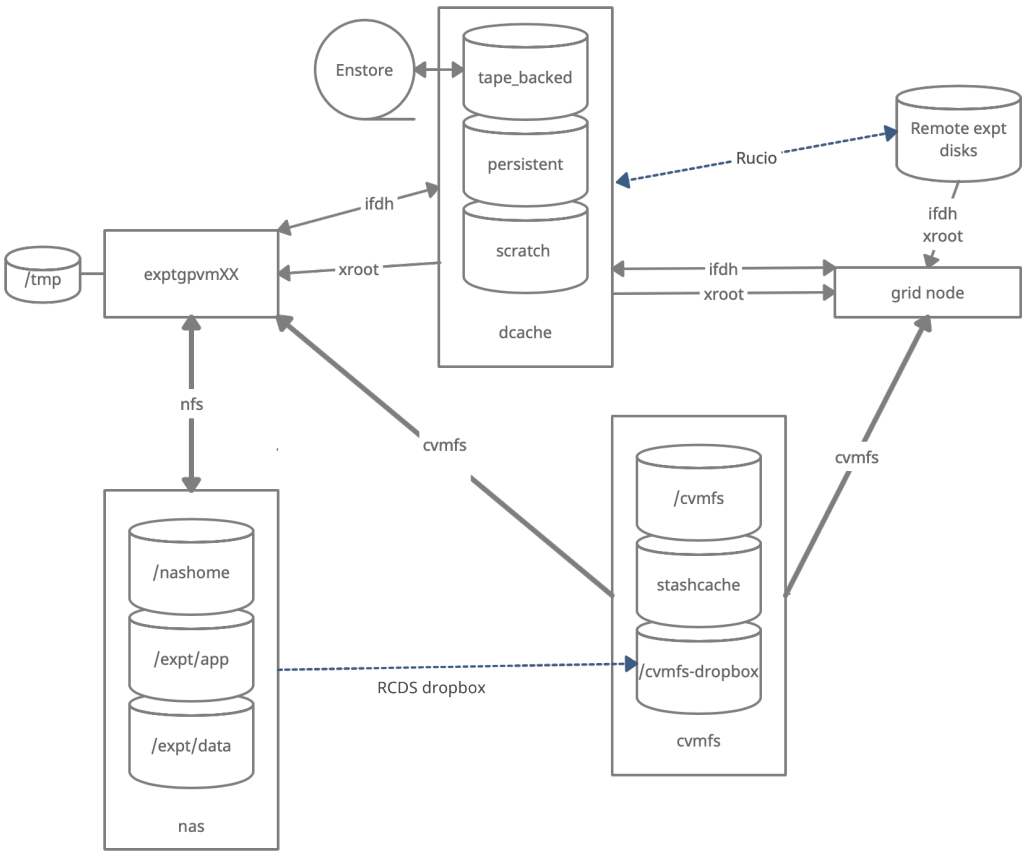


Figure 9.1: Storage systems for Fermi National Accelerator Laboratory (Fermilab)-based DUNE computing. Thick lines denote mounts while thinner lines denote transfer methods such as ifdc and xroot streaming. CERN-based computing shares access to the dCache and CVMFS systems but uses lxplus and eos for local interactive computing.

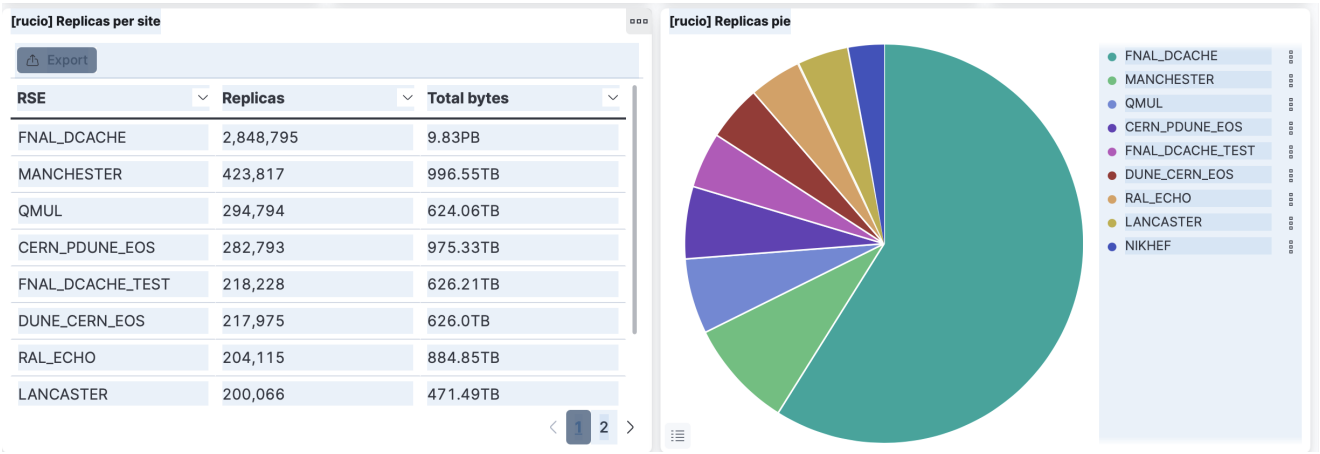


Figure 9.2: Summary of Rucio RSE's as of July 2022. Tape systems such as CASTOR are not shown.

9.2 Data Placement Strategy

DUNE data has a large number of data access patterns based on the size, CPU/byte and frequency of access. Major examples are:

Raw data - Raw data from the TPC detectors comes in 4-8 GB files which are generally run through reconstruction algorithms that take of order 5-10 sec/MB to process. Raw data is generally accessed a few times for calibration and reconstruction as part of an organized production effort.

Reconstructed data - Reconstructed data from ProtoDUNE-SP is around three times smaller than the raw data once the raw TPC waveforms are dropped. It is accessed many times by end users for algorithm development, calibration and production of physics analysis samples. Observed data access rates range from 1 to 40 MB/sec depending on the amount of reprocessing done to data.

Simulated data - Simulated data are around twice as large as real data due to the large amount of simulation information that is kept. Even after raw waveforms are dropped, records are still significantly larger. The access pattern for simulated data is very similar to that of reconstructed data.

Analysis samples - End-user analysis samples are significantly smaller. For example the 1 GeV ProtoDUNE simulation sample consists of $\sim$80,000 files each 4 GB in size while the reduced analysis sample is around 8 GB total. These samples are expected to be access multiple times a day for several months during the development and finalization of analyses.

Flux files - Simulation and cross section extraction require access to large libraries of particle fluxes. These are currently delivered via StashCache. These files are accessed with regularity based upon the production schedule of simulated samples. This access pattern lends itself to caching for increased efficiency to distributed grid jobs.

As described in the data volumes section 6, raw data is kept in multiple tape copies but only kept on disk for a short period until calibration and reconstruction are complete. If reprocessing is needed, the raw data must be prestaged to cache and work on optimizing this process is underway. Recent reconstructed data and simulation copies for users are kept on Rucio/SAM controlled disk, with one copy in the US and one in Europe if possible. Smaller final analysis samples are kept on user or group controlled persistent dcache or NAS areas. Rucio will perform replication and distribution of raw data and production datasets. All datasets will have a well-defined data lifetime within the Rucio system.

Chapter 10

Data Lifetimes and Preservation:

10.1 Formal Data Management Policy

This section describes policies and plans governing the lifecycle of scientific data from the DUNE family of detectors and derived data products from such.

The current authoritative version of the DUNE Data Management plan[131] forms the basis of the data management plans for institutions and sites within the collaboration. The plan encompasses Raw, Analysis, and Scientific Results data tiers. That version is the authoritative source and will be updated as needed. This chapter summarizes the lifecycle elements.

DUNE and Fermilab data management policies are consistent with U.S. Department of Energy (DOE) policy "DOE Policy for Digital Research Data Management"[1]. The host lab portions of the data management plan are consistent with the documented Fermilab "Data Management Practices and Policies for Fermilab Experiments" [2].

The DOE policy references "DOE Policy for Digital Research Data Management: Suggested Elements for a Data Management Plan"[3]. Following these suggested elements, the DUNE data management plan addresses the following data elements.

10.1.1 Data Types and Sources

The data lifecycle policies are applicable to the Raw instrument data from the DUNE detector elements as noted within the DUNE Data Management Plan. The policies will also be applicable to certain Simulation, Analysis and Scientific Results data as deemed appropriate by the collaboration. Policies are also applicable to the metadata needed to understand and catalog the above data types, along with configuration and calibration information. In addition, to interpret and understand the data it is

[1] https://www.energy.gov/datamanagement/doe-policy-digital-research-data-management

[2] https://computing.fnal.gov/atwork/data-management-practices-and-policies-for-fermilab-experiments/

[3] https://www.energy.gov/datamanagement/doe-policy-digital-research-data-management-suggested-elements-data-management-plan

necessary to store and document software code bases used in reconstruction and analysis.

Test, commissioning, simulation, reconstruction and other generated data is expected to have a finite period of usefulness. As such a data lifecycle plan must be followed to include lifetime information in the corresponding metadata at time of creation. At the end of the lifetime a review process is instantiated to determine if the lifetime should be extended or the data may be deleted.

10.1.2 Content and Format

The scientific data contain raw values from the various detector elements and derived values from analysis of the raw information. The data are stored in standard HEP formats (ROOT and HDF5) and various databases (Postgres), all of which may evolve over time. Software and documentation will be stored in industry standard repositories (GitHub).

10.1.3 Data Sharing

All scientific data are available to members of the DUNE collaboration and may be disseminated to collaborating institutions. The data are globally accessible via the network to authenticated and authorized users. The host labs, FNAL and CERN, are the principle location for data, but data may be copied to other collaborating sites for expediency of access and redundancy. Derived data that are to be directly linked to publications (for example tables) shall be granted the same access as the publication. In the future meaningful and accessible data sets will be made publicly available.

10.1.4 Data Preservation

Fermilab, as the host laboratory, will maintain accessible copies of all raw data for a period dictated by lab and DOE policies, generally at least five years past the end of data taking. A longer retention period may be negotiated. Processed and derived data is retained for a lifetime deemed useful by the collaboration, which for ultimate processed results is also minimally five years past the end of data taking. Published and other publicly accessible data sets will be retained as long as technically viable. The data will reside on commercially available archive systems, requiring periodic migration to new technologies.

10.1.5 Protection

Raw and other valuable derived data will have a copy on tape at the host lab as the current solution for reliable storage. The tape systems include periodic integrity tests. Deleted data is only marked for removal, with a final review before tapes are destroyed or recycled. Raw data and other highly valuable derived data will also have an additional redundant copy on external collaboration resources. Tape and disk resident data is protected by storage system authentication and authorization controls.

10.1.6 Rationale

DUNE raw data and derived data products are unique and represent the return on an international investment in neutrino science.

10.2 Policy Implementation

Table 6.8 in Chapter 6 describes the current retention policies, with simulated and reconstructed data samples retained on tape for 10 years.

One of the major concerns for long-term preservation of data is the evolution of storage, operating systems and data formats. Long-lifetime binary data may need to be migrated between storage technologies multiple times. Our current experience is that data written in the ROOT format is readable for at least a decade.

As operating systems and compilers evolve, old code may cease to work or yield consistent numeric results. We will not have the resources to perform continuous integration tests on all code versions, so it is likely that reviving old codes will require substantial effort. The priority will be ensuring that the raw data remain accessible.

Long term, we anticipate that the smaller reduced samples, not the PB of reconstructed raw data or simulation, will form the legacy samples from the experiment. Those formats have not yet stabilized, but will need to be carefully documented and duplicated for use past the formal end of the experiment.

10.3 Data Releases

Final public data releases are approved by the DUNE collaboration in conjunction with publications. Unfortunately, formal data portals, such as those provided by most astrophysics experiments are not currently available for the Intensity Frontier experiments. It would be desirable to work with the host laboratories to set up a shared portal across the suite of neutrino experiments, and possibly the broader Intensity Frontier, with uniform data access methods that build on the experience of experiments such as the Sloan Digital and Dark Energy Sky Surveys in providing public access. At present, we are relying on the supplemental materials features on the arXiv and zenodo to make derived samples available while we negotiate a better technical solution.

Chapter 11

Data Management

11.1 Introduction

The Data Management subsystem brings data from the detectors to the archival storage facility and then distributes it to storage elements around the world. The components include a data ingest manager to receive data from the detectors, a replica manager that knows the location of files and manages transfers between storage elements, a metadata catalog that keeps track of the types and provenance of data, and interfaces to deliver the appropriate files to workflow management systems and to interactive users.

The DUNE Data Management group is currently designing and deploying several new components in the Data Management system to replace the legacy sequential access via metadata (SAM) [132] system, which combined the functions of replica manager, metadata server, and file delivery. Those functions are being separated with well-defined interfaces. The goal is to have the new systems in place before beam operation begins on ProtoDUNE-SP-II. As of the summer of 2022 all the new components of the data management system have been written and have been tested successfully at scale, but are not all in production as yet. Figure 11.1 illustrates the old and proposed data management architectures with the legacy SAM system replaced by a new catalog and the Rucio storage management systems.

The existing DUNE data catalog, SAM, was originally designed for the D0 and CDF high energy physics experiments at Fermi National Accelerator Laboratory (Fermilab). It is now used by most of the Intensity Frontier experiments at Fermilab.

The most important objects cataloged in SAM are individual files and collections of files called SAM-datasets. Data files themselves are not stored in SAM, their metadata is, and that metadata allows users to both identify and locate the physical files

SAM was designed to ensure that large-scale data processing be done accurately and completely, which led to high standards of reproducibility and documentation in data analysis. For example, at the time of the original design, the main storage medium was 8 mm tapes using consumer-grade drives. Drive and tape failure rates were $> 1\%$. Several SAM design concepts, notably luminosity blocks and parentage tracking, were introduced to allow accurate tracking of files and their associated normalization and

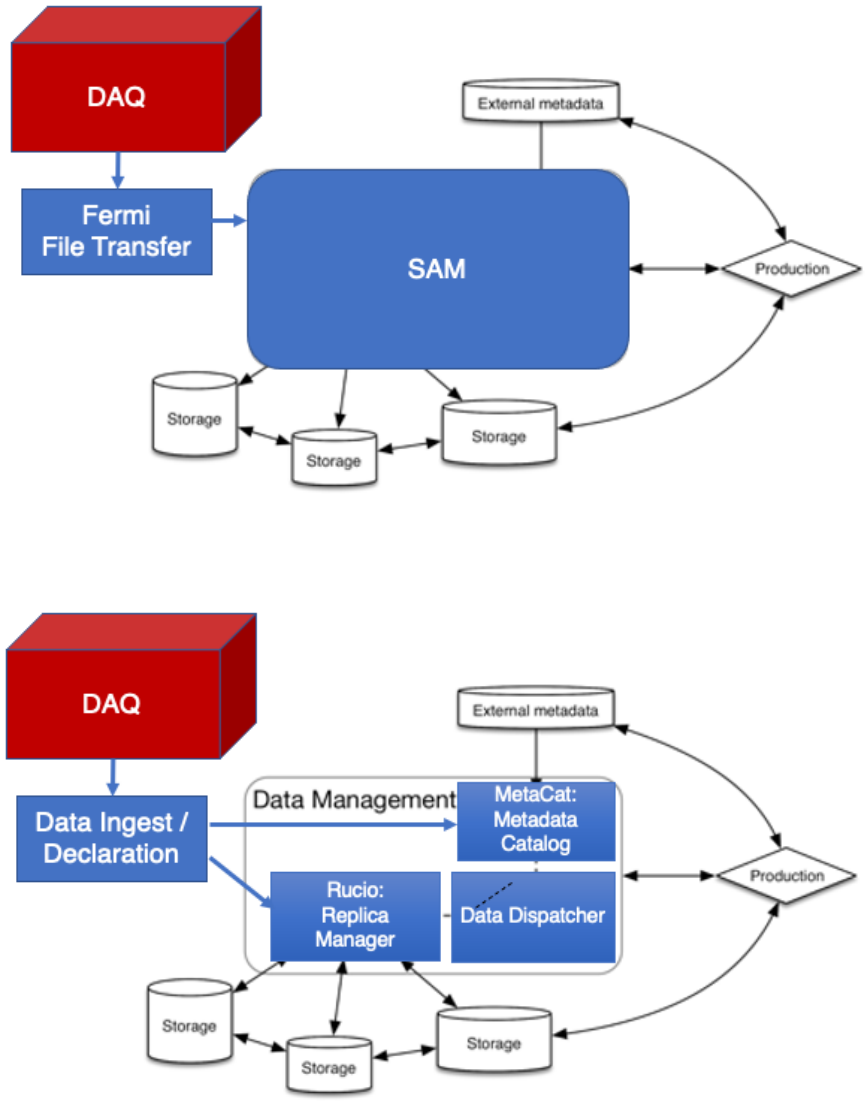


Figure 11.1: Top: Legacy SAM data management architecture. The SAM system provides both catalog and data location information to processing nodes. Bottom: Proposed data management architecture diagram. The SAM functions are divided between the MetaCat (MetaCat) catalog, the Rucio data movement system and a Data Dispatcher that interfaces to processing and monitoring systems.

provenance in a high error-rate environment.

Unfortunately SAM is almost completely file-based and often duplicates run-level information for convenience. The system has served the DUNE Collaboration through the first ProtoDUNE runs but going forward, a replacement is needed. In addition, replacement is also needed to provide better integration with external databases such as the conditions database, and to have data movement capacity for a distributed worldwide storage system.

11.2 Requirements for Replacing SAM Functionality

The replacement for SAM should support use of run or trigger record level information, in addition to file information, as appropriate. For example, DUNE trigger records may span multiple files. The new model needs to generalize from "events" and files to data objects and collections across multiple scales.

The functions of the existing SAM system that we wish to retain and extend are listed here. The first three functions that we need to retain relate to content and characteristics while the last four relate to data storage and processing tools. We propose to separate these functions where appropriate.

1. Describe the contents of individual files in a searchable manner to allow users to select well-defined data and simulation samples.

2. Create and document data collections "datasets" to allow later retrieval based on data characteristics.

3. Track object and collection parentage and describe processing transformations to document the full provenance of any data object and ensure accurate normalization.

The next four functions of SAM relate to the physical location and delivery of files.

1. Store the physical location of objects.

2. Track the processing of collections to allow reprocessing on failure and avoid double processing.

3. Provide methods for delivering and tracking collections in multi-process jobs.

4. Preserve data about processing/storage operations for debugging/reporting.

We propose to break the existing system up into functional sub-units specializing in cataloging, storage and delivery.

MetaCat The catalog function will be replaced by a combination of a file metadata catalog MetaCat and a run configuration and conditions database that stores run level information.

Rucio The file location function is replaced by Rucio [133], the file tracking system originally developed by the ATLAS collaboration. Rucio provides a location catalog for files and rule-based transfers between sites.

Data Dispatcher The existing SAM system supports a station and SAM-project ecosystem where SAM-datasets are submitted for processing as SAM-projects. The SAM-project is a server process that contains a list of files based on a SAM-dataset description and, when asked, delivers location information for the next file in the list. Request, delivery and processing are tracked and recorded. This allows resubmission on failure. A website allows users to see the details of file delivery.

In the following subsections we describe the existing features that we wish to retain in more detail with the proposed replacement technologies in later sections.

11.2.1 Existing SAM Features

11.2.1.1 Data Description Values and Parameters

SAM supports several types of data description fields. SAM Generic "values" such as data_tier, run_type and file_size are common to almost all HEP experiments and are optimized for efficient queries. These values are a limited set and it is possible to request a list of the values (of the "values") already in use.

SAM also allows definition of free-form "parameters" as they are needed within each experiment's instance. This allows the schema to be modified easily as needs arise. Unfortunately, a major problem is that it is not possible to request a list of the values for a given parameter and there is little protection against typographical errors in parameter names or their values. This has, over time, led to considerable chaos. The new MetaCat extends these concepts to make them easier to search and maintain.

11.2.2 SAM Datasets and Projects

11.2.2.1 Datasets and Snapshots

In addition to the files themselves, SAM allows definition of SAM-datasets. A SAM-dataset is not a fixed list of files but a query against the SAM database. An example query would be "run_type protodune-sp and data_tier full-reconstructed and run_number 5141 and version v09_41_00" which would be all files from protodune single phase run 5141 that are reconstructed data produced by version v09_41_00. The SAM-dataset is dynamic; if after this dataset is created, an additional file from run 5141 is reconstructed with v09_41_00, and satisfies the criteria, the SAM-dataset will grow to include it. There are also "SAM-snapshots" that are derived from SAM-datasets and capture the exact files in the SAM-dataset at the time the SAM-snapshot was made. Both have their uses: SAM-snapshots are always reproducible while SAM-datasets are very valuable when processing data from a running experiment or simulation as they can incorporate new data without user intervention.

11.2.2.2 SAM Projects

SAM also supports access delivery and tracking mechanisms called SAM-projects and SAM-consumers.

A SAM-project is effectively a processing campaign across a SAM-dataset that is owned by the SAM system. At launch, a snapshot is generated after which the files in the snapshot are delivered to a set of SAM-consumers. The SAM-project maintains an internal record of the status of the files and SAM-consumers. Compute processes instantiate SAM-consumers attached to the SAM-project. Those

SAM-consumers then request "files" from the SAM-project and, when done processing, tell the SAM-project of their status.

The original SAM implementation actually delivered the files to local hard drives. Modern SAM delivers the location information and expects the SAM-consumer to find the optimal delivery method. This is a pull model, where the consuming process requests the next file rather than having the file assigned to it. This makes the system more robust on distributed systems. There is also a web interface[1] that allows users to view the status of running SAM-projects. This functionality will move to Rucio, the data dispatcher and the workflow systems.

11.2.3 Data provenance and tracking

Because SAM stores the ancestry of files, both children and parents, complex queries such as "give me all the raw data files of type X that do not have children processed and stored by code type Y versions Z-AA" are possible. Complex queries such as this can be used to pick up missing files in reprocessing without creating duplicates.

An additional protection against duplicate files is requiring constrained filenames for output files based solely on the inputs and code characteristics. If variable fields such as timestamps are put into filenames, SAM will happily catalog duplicate instances of the same processing and the provenance information is then necessary to check for and remove duplicates.

11.3 Future Components

We propose to replace the SAM system with the following set of components:

- MetaCat Metadata Catalog: stores file characteristics but not locations - described in section 11.4;
- Data Ingest Manager: takes in data from detector sites, declares it to the catalog and transfers to the managed storage system - described in section 11.5;
- A Rucio-based Replica Manager: stores the physical location of files and provides tools to move them between storage elements – described in section 11.6; and
- Data Dispatcher: interacts with user and production processing systems to provide file location information from the Replica Manager to clients and can track file processing – described in section 11.7.

The Rucio system is already being used for file placement. We are testing use of MetaCat/Rucio in hopes of putting them into production for ProtoDUNE-SP-II runs late in 2022.

These components, collectively, will provide almost all of the existing SAM functionality while providing significant enhancements in implementation and features.

[1] http://samweb.fnal.gov:8480/station_monitor/dune/stations/dune/projects

11.4 MetaCat Metadata Catalog

The current SAM data catalog combines file description with location and delivery. The new MetaCat catalog provides the permanent file description information, with location and delivery handled by Rucio and a data dispatch system. A prototype version of MetaCat has been produced and is being tested as part of our 2022 data challenges. MetaCat is described in Reference [134].

DUNE file metadata includes mandatory and optional fields and can be represented to the API by a Python dictionary or a JavaScript Object Notation (JSON) file. As with SAM, the metadata includes a description of the file itself and data about how it was created, including enough information to follow the full processing chain.

The DUNE data management system operates on a collection of "physical" copies of files (or objects), moving them between storage elements and making them available to the data processing and analysis. A single "logical" file can have multiple "physical" replicas across multiple storage elements. The MetaCat package replaces the SAM file description function and generalizes it to objects. MetaCat stores metadata about logical files and objects and makes it possible to select "interesting" logical files based on criteria expressed in terms of metadata parameters.

The MetaCat design builds on SAM experience, preserving the useful concepts while introducing additional flexibility and constraints where needed. For example:

- To enforce reproducibility, free-form datasets no longer exist and must be explicitly modified.
- Metadata values are more flexible.
- Namespaces are introduced to separate different functions (e.g., user and production files)
- Access to external databases (e.g., for run information) is available where needed. In SAM, run characteristics were often duplicated in every associated file.
- Modification permissions are more granular.
- An Event Catalog is provided, SAM currently does not contain a means of easily determining which file(s) contains a given event. If a data acquisition (DAQ) system is writing multiple streams, an event from a given subrun could be in any stream. Existing neutrino experiments are low enough rate that this has not been an issue but ProtoDUNE already needs this feature. The SAM replacement will require the development of a true two-way map between objects (events) and collections.

Although the main target user of the project is ProtoDUNE/DUNE, MetaCat is generic enough to be usable by other experiments, as well.

11.4.1 MetaCat requirements

MetaCat has to satisfy the following general requirements:

- The Metadata representation must be powerful, flexible and abstract enough to accommodate a wide range of metadata data types and possibly complex metadata structures.
- It needs to scale to several 100 million objects, based on Fermilab fixed-target experience, and

likely to billions of objects by the end of Deep Underground Neutrino Experiment (DUNE) running.
- The unit of operation of the catalog should be an abstract "file" or "object" with a minimal but sufficient set of predefined attributes and the ability to add user-defined attributes in a flexible and convenient way.
- The file selection mechanism must be powerful and at the same time simple enough to be able to express a wide range of metadata selection criteria.
- The MetaCat should not need knowledge of physical file replicas, aside from a checking mechanism to make certain that all physical objects have logical representations in the catalog.

Based on our experience with SAM, an important improvement is to provide a mechanism to use data from external metadata sources, such as conditions or runs databases, as part of metadata queries without copying the external data into or making it appear as part of the metadata database.

11.4.2 MetaCat implementation

In SAM, files were identified by a unique immutable name within a single namespace. To support multiple use cases (for example user and production spaces) MetaCat's objects can be identified by names within namespaces. The name of an object is unique within its namespace. A namespace/name pair uniquely identifies an object in MetaCat. In addition to namespace/name, files are assigned unique text identifiers. These identifiers are primarily for internal use within the MetaCat database, but are available to the user. At the time of the file declaration, the user can specify a file ID, which must be unique, otherwise MetaCat will generate a unique file ID. After a file is declared to MetaCat it can be renamed. Both namespace and name can be changed, provided the new namespace/name pair is unique. However, file ID can not be changed.

As in SAM, MetaCat stores metadata associated with files and datasets. A MetaCat-dataset is a collection of files. A MetaCat-dataset can have child MetaCat-dataset. Unlike the active SAM-datasets defined in SAM, files must be explicitly added to and removed from a MetaCat-dataset. A file can belong to zero or more MetaCat-datasets.

As in SAM there can be a many-to-many provenance relationship between files. A file can have zero or more derived (child) files and zero or more parent files. The system makes sure the provenance relationship is not circular.

Files do not necessarily have to have physical locations. Some use cases, for example those involving production and then merging of output files, may include a "virtual" file in the parentage relation between the large parent file and the merged child file.

Metadata attributes are name-value pairs. Any file or MetaCat-dataset can have zero or more metadata attributes. Attribute names are alphanumeric words optionally combined with dots. Any JSON structure can be an attribute value. Therefore, a file or a MetaCat-dataset attribute set is a JSON dictionary.

Here is an example MetaCat entry for a processed data file from ProtoDUNE-SP.

```
checksums :    {'md5': 'ea8d1a009f23accf9582f9e2bd7f58fd',
                'adler32': 'c689390f', 'enstore': '3896981774'}
```

```
children  :    ['52593351']
created_timestamp:  1616688097.41347
creator   :    dunepro
fid       :    52593202
name:    np04_raw_run005141_0006_dl1_reco1_42477998_0_20210324T231631Z.root
namespace :    pdsp_det_reco
parents.  :    ['6606706']
size      :    3599190934
metadata. :   {
    'DUNE.campaign': 'PDSPProd4',
    'DUNE_data.DAQConfigName',
    'np04_WibsReal_Ssps_BeamTrig_00021',
    'DUNE_data.acCouple': 0,
    'DUNE_data.calibpulsemode': 0,
    'DUNE_data.detector_config':  ......
    'DUNE_data.detector_config.object': ['cob2_rce01', 'cob2_rce02', .....
    'DUNE_data.feshapingtime': 2,
    'DUNE_data.inconsistent_hw_config': 0,
    'DUNE_data.is_fake_data': 0,
    'beam.momentum': 7,
    'data_quality.online_good_run_list': 1,
    'detector.hv_value': 180,
    ...... many other items not shown  .....
}
```

11.4.3 Ownership and Permissions

A MetaCat user is identified by a unique username. A user can be a member of zero or more roles (groups). Namespaces and attribute categories have owners. A namespace or category owner can be either an individual user or a role. If the namespace or the category is owned by the role, that means it is automatically owned by all role members. The namespace owner automatically owns all the datasets and files in that namespace (either directly or via the role membership). Ownership does not automatically propagate along the dataset parent/child or file provenance relationships. Only the owner of a dataset can add or remove files from the dataset. Only the owner of a dataset can change its metadata attributes. The same is true for files in that only the owner of a file can change its metadata attributes.

11.4.4 Queries

One of the most important parts of the MetaCat functionality is the ability to query the database for "interesting" files. Essentially, the query is a logical expression in terms of file and/or dataset metadata attributes, file provenance relationship, and dataset parent/child relationship. There are two types of queries in MetaCat: file queries and dataset queries. File queries return a set of files (a list of file IDs) whereas dataset queries return a list of datasets. The file and dataset lists are not guaranteed

to have a consistent order, so they are in fact "sets" rather than "lists." A query is merely a formula specifying the selection criteria. MetaCat does not save the results of the query, so re-running a query can produce different results as files are added/removed from the system, to/from the datasets or their metadata attributes change. However, there is an option to save results of the file query as a new MetaCat-dataset or add selected files to an existing MetaCat-dataset. A query can be saved into the database under a name within a namespace. This function can be used to publish complicated queries and make them reusable by other users.

11.4.5 MetaCat Query Language (MQL)

The original SAM query language was produced in the late 1990s to provide a restricted command set and avoid having users execute full SQL queries.

In MetaCat the query is written in a specialized query language, MetaCat Query Language (MQL). MQL allows the user to specify file/dataset metadata attribute criteria, use MetaCat-dataset parent/child relationships and file provenance relationships to select a set of file or a MetaCat-dataset. Further, simple queries can be combined into more complicated ones using logical operations like union, join, subtraction. Named queries can be referred to from within the MQL expression by their name.

11.4.6 External data sources

MetaCat functionality includes the ability to access external metadata sources, such as conditions databases, and use the data stored there to filter file selection results. In order to make an external metadata source available to MetaCat instance, a Python plug-in module with a standard interface must be provided. Once the module is plugged into a MetaCat instance, it can be referred to in the MQL query as a named filter and used to filter results of an intermediate query or queries within the MQL expression or even "inject" metadata from the external source into the query and make it available as a selection criterion.

11.4.7 Architecture and Interfaces

MetaCat is a typical web-services database application. The underlying database is not exposed to end users. It can be accessed via a Python API, but the primary method of interacting with the system is through a web service or web GUI. Use of this REST-based web service improves the scalability and cacheability of the system and completely unties the server side from the client implementation. Any standard HTTP/HTTPS client can interact with the system either directly or through a standard HTTP proxy or cache.

The system publishes the following interfaces:

- a direct database access Python API,
- a Web services REST interface,
- a python client side API that communicates with the server via HTTP, and
- a command line interface with a basic set of commands to enter (modify) data into the database and to query the database.

11.4.8 File naming conventions

In principle, a data catalog such as MetaCat means that file names could be completely generic. In practice random bitstrings makes it very difficult for humans to interpret and check what they are doing. DUNE does not yet have a general file naming convention, with different subgroups defining their own. A general specification would be useful and will be produced. Similarly, datasets can be identified based on their characteristics via queries but should also have a naming convention. Previous experiments have generally encoded the data format, data tier, run/subrun number and processing campaign in the file name for easy comprehension.

11.4.9 Current Status

MetaCat has been successfully tested in the Data Challenge at full operational scale. We are currently operating it in parallel with the SAM catalog. A plan for switching from SAM/Rucio is being finalized with full deployment either during ProtoDUNE-SP-II or for subsequent processing steps in 2022-2023.

11.5 Data Ingest Manager

The Data Ingest Manager will run at detector sites. It consists of two components: the Ingest Daemon and the Declaration Daemon. The Ingest Daemon will run at each detector location, at CERN EHN1, SURF surface and Fermilab. Each instance will detect new files in the data store, extract the metadata and add new metadata fields if necessary, calculate the checksum of the files, initiate and monitor the transfers to the first managed storage element, and send a signal back to the DAQ when the file has been transferred, properly stored, and can be deleted from the data store. Files will be transferred to a drop box on the nearest managed storage element, using FTS3 as the transport mechanism.

The Declaration Daemon will run on or near a storage element that is managed by the replica manager. It will detect new files that have been added to the drop box, and declare them to the metadata catalog and the replica manager. It will then instruct the replica manager to send these files to permanent tape-backed storage and monitor the transfer to be sure that this has been done, and then send a signal to remove them from the temporary drop box.

These daemons will be performing most of the functions of the current data transport system. They only require modifications to interface with the new replica manager and metadata catalog rather than the current ones. These daemons were both successfully tested at full scale in the recent Data Challenge 4. We expect to use them during the upcoming ProtoDUNE-HD running.

11.6 Rucio Replica Manager

The Rucio[135] replica manager system is now used to track the physical location of data files and ensure that they are moved from point to point as needed. This system was originally developed by the ATLAS experiment and has now been deployed to a number of other HEP and astronomy experiments. DUNE has had a Rucio server since 2019. All raw data that was taken in the fall 2018 ProtoDUNE-SP (NP04) run and since that time, as well as all raw data taken by the ProtoDUNE-DP (NP02) run, are now tracked by Rucio. All output of Monte Carlo (MC) simulation and reconstruction are also tracked by Rucio. DUNE plans to take full advantage of the data lifetime features of Rucio.

Rucio is a rule-based system. Files are sent to remote servers via a system of rules that are created by the experiment data managers. Rucio provides tools to implement those rules. Data can be accessed interactively or in batch jobs and Rucio has a built-in system of delivering the URI of the closest replica to the running job for access via streaming.

DUNE has contributed several features to the Rucio base software. These include a new method to map from the `scope:filename` logical file name to the path-based directory structure that we use on tape sites. We also have added a DUNE-specific hook that requires any new files declared to Rucio already be declared to the MetaCat system.

We have implemented object store support, modularized the virtual organization (VO) customization code and added policy packages specific to the DUNE VOs, started to develop a lightweight client, and factorized external dependencies in order to make integration into DUNE and other non-ATLAS VOs easier. These contributions have been done in close cooperation with the CERN Rucio team and have been fed back to the main Rucio project code base.

Rucio has an internal metadata functionality but our evaluation indicates that it doesn't provide the full functionality required by DUNE physics and offline processing, and thus DUNE has proceeded to develop its own metadata catalog.

The development work needed on Rucio in the next few years includes completing the lightweight client, integration of this client into all DUNE workflows, adding quality of service support and proximity mapping and fully integrating the MetaCat. In the longer term Rucio will constantly need development as issues are uncovered and new features are needed. The main work that remains to be done in implementing the Rucio server for production is back-loading all relevant data from the legacy SAM system into Rucio. Currently about 50% of the data by volume is known to Rucio. In the Data Challenge 4 we used Rucio for all transport of data between sites and successfully tested it at the full expected data rate of ProtoDUNE-HD, which is also similar to the eventual data rate of the far detector.

11.7 Data Dispatcher

The Data Dispatcher replaces the project management and file delivery functions that were previously done by the SAM system. The Data Dispatcher functions include:

- Creating projects to process collections of data,
- Delivering file handles of the files to consumers,
- Keeping track of the project progress, including consumer status and files consumed,
- Keeping record of projects, consumers, and file consumption for a specified amount of time,
- Providing project monitoring and control to a user or a client, and
- Organizing and coordinating data processing among data consumer processes.

A Data Dispatcher has been written and is now being interfaced to the general use Fermilab job submission system and tested for processing large quantities of data. The exact boundary between the Data Dispatcher and the enhanced Workflow System described in Chapter 13 has not been finalized. It is likely that other Fermilab experiments will need the Data Dispatcher functionality if SAM is deprecated.

11.8 Tools and Integration

The data management system relies on several sets of underlying tools. The Rucio replica manager is dependent on the CERN File Transfer Service FTS3 [136] to actually move files from one storage element to another, although it can also be configured to use other protocols as needed. The FTS3 service in turn is dependent on the GFAL2 (Grid File Access Library) set of utilities.

We have developed a suite of data transfer monitoring tools and dashboards to track data placement and Rucio transfers. We have tested and integrated the CERN CTA Tape service into DUNE data management and are now doing the same for the RAL CTA based Antares Tape service. As new sites come along each must be integrated as a Rucio Storage Element. An unforeseen development was the need to deploy a StashCache instance in the UK to solve a low job efficiency problem due to slow data access.

Chapter 12

Networking

With the DUNE experiment located at two geographically distant sites and having computing resources distributed around the globe, the design and operation of networking will be critical to the success of the DUNE experiment. The network requirements can in general be divided between three main areas: far detector (FD), near detector (ND) and distributed computing. Assuring that interconnection and bandwidth requirements between each are met with appropriate uptime is necessary for operations and successful physics results. The requirements for each system are based upon current estimates of data rates for both the FD and ND data acquisition (DAQ), technical networks, and slow control networks.

12.1 SURF to Fermilab

The FD networking encompasses the Local Area Network (LAN) at Sanford Underground Research Facility (SURF), the connection to the wide area network, and the network path back to storage services at Fermi National Accelerator Laboratory (Fermilab) in Batavia, IL. This network path between the FD at SURF and the Fermilab campus is the most important network path for the DUNE raw data. This path will be used both for transfers of raw data, and also for connectivity between SURF and operations centers at Fermilab and elsewhere. While operations traffic is expected to use this connection, it should be noted that safety systems will not rely upon the network connection.

The requirements for the network connection between SURF and Fermilab were explored extensively in the High Energy Physics Network Requirements Review [137] . The primary bandwidth requirement is determined from both the steady-state yearly transfers from the DAQ, and by burst transfer rates during calibration and extended-readout trigger records. The Consortium Interface document between the DUNE DAQ Consortium and the DUNE Computing Consortium and the DAQ specification [138] states that no more than 30 PB/year of data will be output from the DAQ for permanent storage. This translates into 8 Gb/s of steady-state data transfer from SURF to Fermilab, and includes raw data from beam triggers, supernova candidates, cosmic triggers, and calibration runs.

The most extreme burst requirement considered in the ESNet review was the handling of extended-time trigger records for supernova neutrino burst (SNB) candidates. These trigger records, in contrast to the steady beam and cosmic ray rate, are expected to occur one to two times/month and involve

data volumes of up to 160 TB/module of uncompressed FD data. The ability to produce a reasonably fast pointing signal would be extremely valuable to optical astronomers doing follow-up, especially if the supernova were in a region where dust masks the primary optical signal. The need to be alert to SNBs and to quickly transfer and process these data imposes stringent requirements on triggering, data transfer, and reconstruction beyond those imposed by the more regular beam-based oscillation physics. For example, an uncompressed SNBs readout of the first two FD modules will be on the order of 320 TB in size and take a minimum of seven hours to transfer over a 100 Gb/s network, and then take on the order of 90,000 CPU-hrs for signal processing and pattern recognition at present speeds (See Chapter 3 for details). If processing takes the same time as transfer, a peak of 10-20,000 cores would be needed. In order to have initial analysis results of a SNBs trigger record within one day, the network between the FD at SURF and the Fermilab campus is being designed for a capacity of 100 Gb/s.

As the network will be used for both raw data transfer and for remote operations, there is a requirement to have both a primary and secondary network path between the FD and Fermilab. Working in conjunction with ESNet, the Core Computing Division at Fermilab has developed a deployment plan for two geographically separate network paths. The map of the proposed paths is shown in Figure 12.1 with the two paths merging between St. Cloud, MN and the Fermilab campus. Having a single contracted path from MN to the Fermilab campus is not considered a significant risk given the number of alternate paths between the two locations that are available. The primary path connects network switches at the top of the Ross Shaft (at SURF in Lead, SD) to Fermilab via ESNet infrastructure. As of February 2022, the primary path is available from Lead, SD to Fermilab at limited bandwidth of 10 Gb/s. In order to complete this path, a vendor will be secured to provide infrastructure between Lead and the Ross Shaft. After completion, the primary path will provide 100 Gb/s guaranteed bandwidth. The secondary path will provide 10 Gb/s of network bandwidth and should be available in 2022 or 2023, and discussions are underway to secure dedicated 100 Gb/s of network bandwidth once detector operations have begun in 2028. [1] The secondary path will be from the Yates Shaft (also at SURF) through Rapid City and Sioux Falls, SD to Fermilab via ESNet infrastructure. Additionally, a tertiary path that utilizes vLAN infrastructure from the South Dakota Higher Education Network (REED) and a southern path through Colorado and Kansas City, KS will provide 10 Gb/s of bandwidth. During construction, the tertiary path will serve as the primary link to the DUNE FD and then revert to a tertiary role once the dedicated primary path is complete. The networking is expected to be completed in stages with both paths capable of the full design bandwidth by the time that FD physics operations commences. A detailed schedule for networking is shown in Figure 12.2.

The network uptime is defined by the capability of the FD DAQ system to stage raw data locally in case all connectivity is lost. This DAQ is designed to have capacity to store one week of raw data from the detector on local storage at SURF. Given the nominal rate of 7.98 Gb/s from the DAQ, and assuming that the network interface cards (NICs) present in the local storage have effectively infinite bandwidth, the additional 92 Gb/s of bandwidth available on the primary path means that were connectivity to be lost for an entire week, the backlog in raw data could be cleared within one day of reconnection. Given the presence of three geographically separate paths from SURF to Fermilab, if all three networks have an uptime of $> 90\%$, then the expected networking live time (99.9%) is more performant than the DAQ required live time (98%) by more than a factor of 10. If both the primary and secondary network paths were interrupted but the tertiary path operating, there would still be limited impact on primary physics data taking but operational adjustments would be necessary. As mentioned earlier in this paragraph,

[1] This information was updated at the beginning of 2022 from early estimates that had the secondary path at 10 Gb/s.

the annual data volume from the FD requires only 7.98 Gb/s of continuous bandwidth, and therefore could be actively transmitted via the tertiary path (10 Gb/s) without need for caching at SURF. Any additional, data-intensive detector operations (e.g. calibrations or streaming of Supernova readouts) would require either local caching at SURF or be postponed until one of the primary or secondary paths are restored. Given the location of SURF and the potential for natural or man-made disasters, there is a real risk of weeks- or months-long repair times following physical damage to a network path (e.g. forest fire). The ability to continue normal physics operations via the tertiary path, and plans to adjust FD operations during such a period, is an important part of the operational planning of Deep Underground Neutrino Experiment (DUNE) Computing.

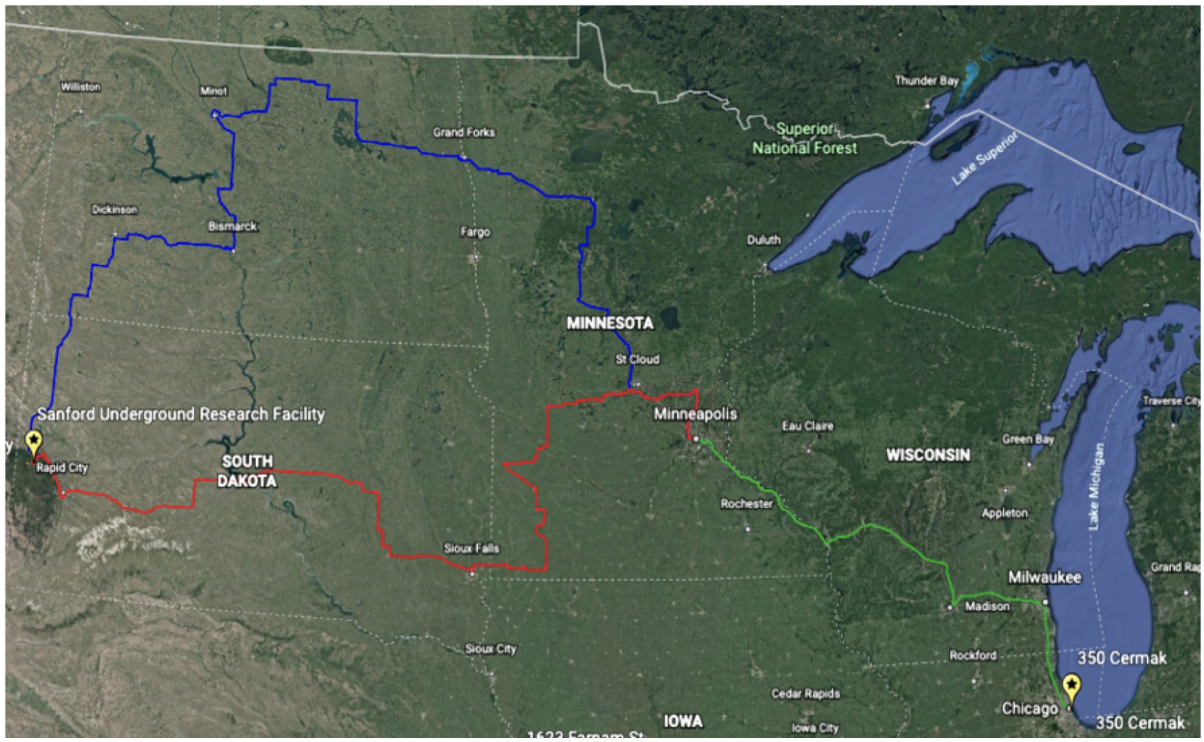


Figure 12.1: A map of the geographically separate primary (blue) and secondary (red) proposed network paths between SURF and Fermilab.

12.2 Far and Near Site Local Area Networks

The LAN for the Far and Near Sites will be designed, installed, and commissioned as a collaboration between the DUNE DAQ consortium and the Fermilab Core Computing Division (CCD). The DAQ has provided requirements to CCD and preliminary designs of the LAN have begun. The installation of networking infrastructure at SURF will be paid for by the LBNF/DUNE-US Project and is scheduled to start in 2026. Once network installation has begun, there is a joint responsibility between SURF and CCD network experts to configure and maintain the interface with external high-capacity networking for data transfers to storage at Fermilab. The installation of the LAN at the Near Site is planned to begin in 2029, and follows a similar set of responsibilities as at SURF. The DAQ group will provide requirements and CCD providing design, commissioning, and operational support for the Near Site LAN.

DUNE FD WAN Bandwidth Timeline Projections:

Date	*Stage of the experiment*	*Primary Path*	*Secondary Path*	*Tertiary Path*
Now	Cavern excavation	10GE	< 1GE via SURF	none
2025	Detector construction	10GE	< 1GE via SURF	none
2027	Computing/DAQ deployment	100GE	10GE	< 1GE via SURF
2028	Cryo deployment completed	100GE	10Gb/s+	10GE
2029	Start of science	100GE	10Gb/s+	10GE

vLAN service provided by REED/GPN (shared)
Dedicated circuit Ross Dry Bldg. to Chicago
Dedicated circuit Yates Complex to Denver (10GE or 100GE)

Figure 12.2: The current timeline for the implementation of networking connection between SURF and Fermilab. Note that while the secondary path bandwidth is listed as 10+ Gb/s, as of Summer 2022 it is anticipated that there will be 100 Gb/s of capacity once DUNE FD is operational.

12.3 Global Connectivity

Connection to European sites is accomplished via the Energy Sciences Network (ESnet), the pan-European research network (GÉANT) and National Research and Education Networks (NRENs). The current aggregate transatlantic bandwidth is of the order of a Tbit/s, of which 400 Gb/s is from ESnet (Boston, New York and Washington DC) to GÉANT (London, Amsterdam and Geneva (European Laboratory for Particle Physics (CERN))). GÉANT in turn peers with the NREN in each participating country, where details pertaining to each DUNE site vary. As an example, in the UK the NREN is JISC-JANET, which at the time of writing has a 400 Gb/s core and connects to the GridPP-RAL site redundantly at 100 Gb/s. Similarly the CC-IN2P3 site in France is connected to RENATER at 100 Gb/s, and the FZU site in the Czech Republic is connected to CESNET at 100 Gb/s. DUNE expects all participating countries to ensure that as part of their pledges of CPU and storage capacity, their sites have commensurate network connections. No systematic issues are expected to arise, and none have done so in the WLCG/LHC context. DUNE participates in the worldwide HEP Network coordination body which meets twice per year (the LHCOPN/LHCONE Group).

Layer-3 Virtual Routing and Forwarding (VRF) provision is now very prevalent in HEP. VRFs provide a logical routing overlay that can allow for traffic engineering to utilize high-capacity paths where needed. The LHC community uses a VRF called LHCONE, and this has also been used for DUNE traffic along with other non-LHC experiments such as Belle II. At present DUNE is agnostic regarding the use of LHCONE, and since Fermilab is connected to LHCONE it can easily accommodate sites with or without such provision. Investigations are currently underway to determine the technical requirements for the creation of a separate DUNEONE VRF were it to ever be required. It is, however, not currently foreseen, and does not form part of our baseline planning.

Chapter 13

Workflow Management

13.1 Introduction

Efficiently matching CPU and data is a long-standing problem in HEP computing. DUNE proposes to use a relatively non-hierarchical system that uses improved knowledge of the computing properties of applications (I/O rate, memory needs, data size) and the network connections between Rucio Storage Elements (RSEs) and Central Processing Units (CPUs) to optimally match processing and data. The sequential access via metadata (SAM)/JobSub system described previously and in the first section 13.2 below, has served Fermilab experiments well for 2 decades but the complex international nature of Deep Underground Neutrino Experiment (DUNE)'s computing resources motivates the development of a new Workflow system that uses more detailed information to define and optimize jobs running across DUNE global computing resources. This chapter describes the existing generic Fermilab production system which is being used to process data from ProtoDUNE, and lays out the requirements for the future system and the prototype of it that has been produced.

13.2 Existing Production Submission Infrastructure

13.2.1 Production Operations Management System

Production and large-scale analysis jobs currently use Fermi National Accelerator Laboratory (Fermilab)'s Production Operations Management System (POMS) [139] for submission. POMS is used by a large number of experiments and provides both GUI and command-line options for job launches (both immediate and scheduled), recovery project setup, and integrated monitoring with Fermilab's Landscape project. Job submission is typically performed via Fermilab's JobSub tool [122]. JobSub in turn interfaces with the GlideinWMS workflow management system [140] for resource provisioning and matchmaking to slots at Fermilab, on dedicated DUNE resources at other sites, or opportunistic cycles on the Open Science Grid (OSG) and Worldwide LHC Computing Grid (WLCG). Figure 13.1 illustrates the entire chain, including interaction with storage elements within the job. The architecture can also provision resources on high-performance computing (HPC) resources, such as Cori at the NERSC, within the HEPCloud [141] infrastructure. The submission mechanism is unchanged whether the jobs are High Throughput Computing (HTC) or HPC; this seamless transition is key to efficiently utilizing available

resources and also saves the job submitter significant effort by not requiring customized submission infrastructure for different resource types.

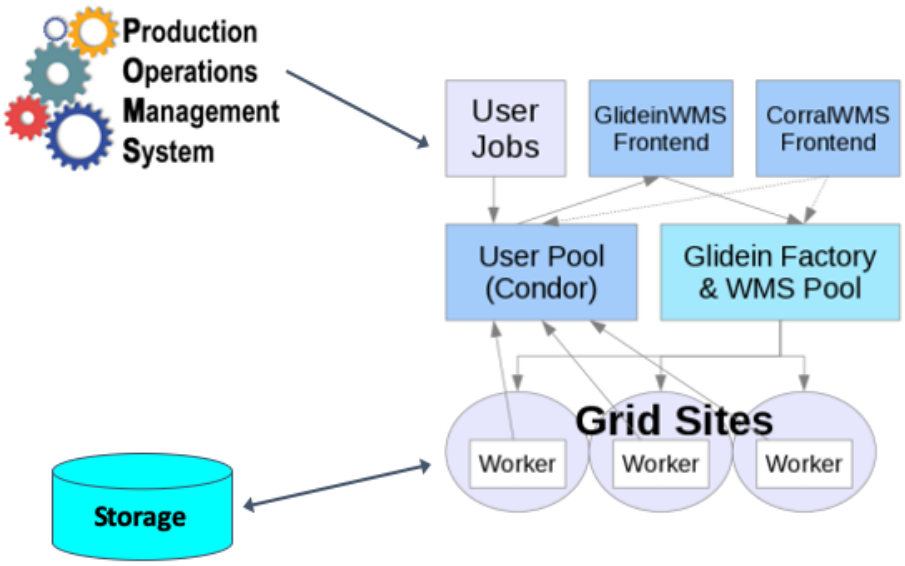


Figure 13.1: Overview of the current DUNE Production workflow setup used also by the ProtoDUNE detectors for data reconstruction and simulation. Production group members interact with POMS to submit jobs, which uses the JobSub tool to submit jobs to a HTCondor scheduler. GlideinWMS provisions worker node resources and jobs match to the available worker node slots. DUNE jobs interact with storage elements at Fermilab and other sites both for input copy (or streaming for most production workflows) and output copyback.

Choosing a Glidein-based system at this stage of the experiment had several advantages. DUNE was able to quickly leverage the existing FIFE [142] toolset, including POMS and JobSub, negating the need for significant effort from the experiment in getting jobs running quickly. Since the system is in use by other neutrino experiments at Fermilab, it is easy for new DUNE collaboration members coming from these experiments to begin submitting jobs quickly as they are working with a familiar system. The DUNE production workflows were also able to leverage the existing infrastructure support teams in place to serve other collaborations and consortia such as CMS and OSG. Finally, as GlideinWMS is widely used in HEP, setting up new sites becomes extremely straightforward, especially if the site is already supporting another experiment that uses GlideinWMS. Our integration times for new DUNE sites are typically less than one week and successful production jobs immediately after opening up the site are now the rule rather than the exception. This ease of setup has been a key enabler of DUNE's international expansion. International sites routinely deliver more than 50% of DUNE's CPU resources as illustrated by the total DUNE Production wall hours for August 2021 shown in Figure 13.2. International sites regularly run the full suite of DUNE jobs, including ProtoDUNE data reconstruction and user analysis.

13.2.2 Software, Input Data Distribution, and Output File Handling

DUNE builds its software suite for Scientific Linux. Since October 2019 DUNE jobs have been automatically run inside a Singularity[143] container at supported sites via a GlideinWMS mechanism that

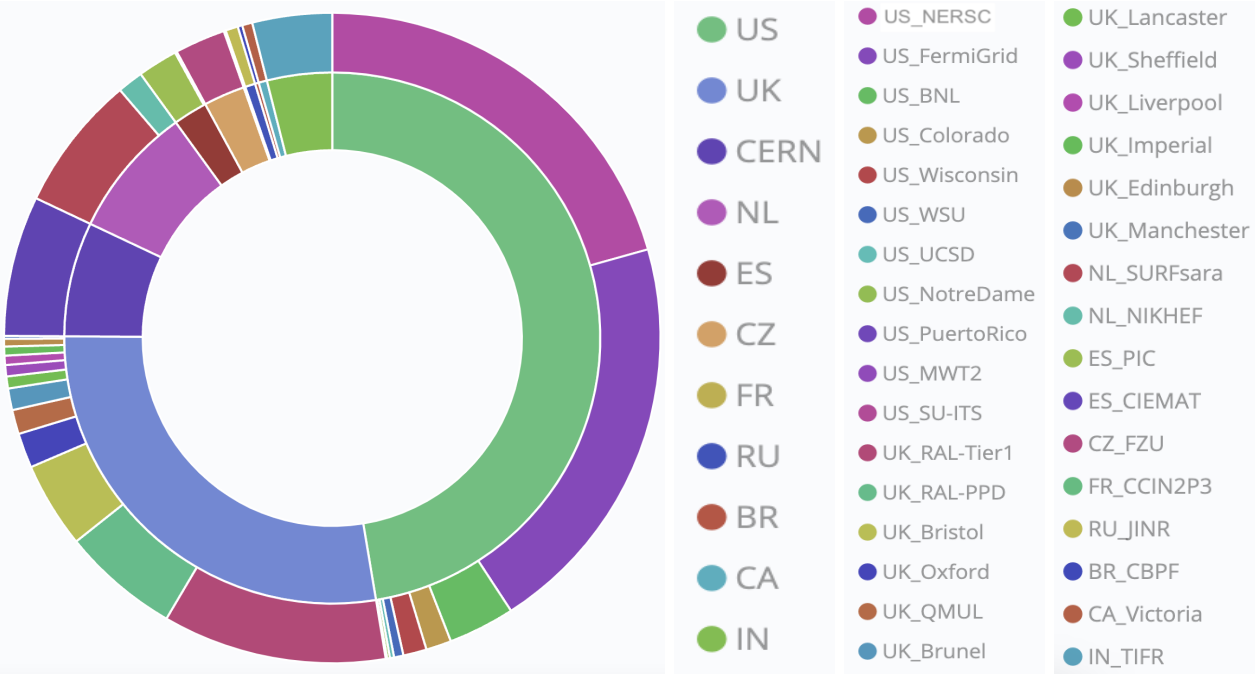


Figure 13.2: Distribution of wall time for DUNE production jobs, October 2021 to March 2022. Inner ring: country. Outer ring: site. Over 50% of wall hours have come from outside the United States in Fiscal Year 2022.

requires no user knowledge of Singularity other than specifying the desired image. This reduces the possibility of errors and guarantees a homogeneous environment across all sites.

As described in Chapter 11, Rucio [133] now provides file location information to processes. Production jobs typically stream input data files via xrootd, though in some cases they will copy a file to the worker node and directly read the local copy. The data source can be any storage system reachable from the worker node and to which DUNE has access. This is frequently Fermilab dCache, but EOS (EOS) at the European Laboratory for Particle Physics (CERN) and other storage elements in Europe are also used (jobs run at CERN would get their inputs from CERN EOS, for example, while those run in the UK often use replicas sited in the UK).

We have recently implemented minimal match optimization between RSEs and compute sites in SAM as many data samples used for user analysis have now been duplicated at European RSEs. SAM now directs data from European sites to jobs running in Europe when possible. This is done via a reasonably simple look-up table that provides a prioritized list of data locations for each compute site. Even this simple system substantially raised the efficiency for high I/O applications at many sites. Section 7.2 describes studies of RSE-to-CPU data rates that were used to implement the priority tables. The new workflow systems described below will support much more sophisticated matching algorithms.

Several DUNE workflows require one or more auxiliary input files such as calibration files or neutrino flux information, and other inputs necessary for Monte Carlo (MC) generation. Some simulation workflows randomly choose several such input files for each job from a much larger set, so the file overlap between jobs is small. Additionally the files are typically tens of MB in size. These two attributes make these files poor candidates for placement in a standard CERN VM File System (CVMFS) repository. For such

files we store them in a StashCache [32] repository, accessible in a POSIX-like fashion though a CVMFS overlay. With this method there is still some level of shared caching on a worker node, but as files need to be copied in from the source (Fermilab dCache), it happens in a transparent way, meaning the user can simply access the files via a CVMFS path in the `dune.osgstorage.org` repository.

For output file handling nearly all workflows currently copy their outputs to Fermilab dCache, with a small minority (ProtoDUNE-DP jobs) also copying to a storage element at IN2P3 in France. We use Rucio to manage file replication to other sites. In the future DUNE will likely move to a more distributed copyback model: at sites with local storage, a job can simply copy its output to a local location, and then we can use Rucio for replication as is already done, rather than requiring everything first go through Fermilab.

There is an exception for workflows run at NERSC; we read input auxiliary files from and copy output files back to Cori's global scratch filesystem. For job outputs, a separate process performs bulk transfer back to Fermilab from one of NERSC's dedicated data transfer nodes. We expect that workflows on other future HPC platforms will follow a similar approach, especially at places without external connectivity on the worker nodes.

13.3 Workflow System Requirements for Replacing SAM/JobSub Functionality

As well as the data management functionality of SAM described earlier, it will be necessary to provide a replacement for SAM's role in directing files to jobs and keeping track of what work has been done. We propose to take this opportunity to extend the SAM model further, by allowing the replacement Workflow system to determine which requests for work should be processed at a particular place and time, as well as determining which file to process next.

Discussions with partner projects reveal a variety of constraints imposed by the circumstances of different sites. Some sites have abundant network capacity and can readily support jobs that stream input data from elsewhere. Others have relatively limited networking compared to the volume of computing power they make available, and expect experiments to only send jobs that are either minimally I/O intensive or will process onsite data. A third class of sites has access to metropolitan or regional networks that can accommodate streaming from nearby sites only.

This landscape has led the computing consortium to look at models where systems in which jobs can be accurately matched to sites where their input data is present, to deal with the most constrained class of sites. However, the system must also allow for a larger fraction of data access over the network for jobs running at less constrained sites.

As noted above, SAM already has some region level matching capabilities, which have proven to be very useful, but significant effort will be needed to fully implement an adaptable application-aware matching system with site-level granularity.

To accommodate these constraints, we have designed and prototyped a generalization of the SAM/JobSub model with site-aware matching and an additional high level request interface. This Workflow System gathers and tracks requests. Using our Sites and Services model described in Section 7.4, generic jobs

arrive at grid worker nodes on a Computing Element at a site and ask a central Workflow Allocator service what they are to work on and which file to process within that activity. This matching is based on the physical characteristics of the worker node job slot (such as memory, processors, and lifetime), the application characteristics and what data are unprocessed and suitable to be accessed from local or remote Storage Elements.

13.4 Request Lifecycle

The central concept of the proposed workflow system is a request that describes how some data processing activity is to be carried out. Requests are submitted by users (which may include members of a central production team) to the Workflow Database described below, where it progresses through several states, for example, draft $\rightarrow$ submitted $\rightarrow$ approved $\rightarrow$ running $\rightarrow$ paused $\rightarrow$ running $\rightarrow$ checking $\rightarrow$ completed $\rightarrow$ archived. Human intervention is needed for some transitions, e.g., from submitted to approved. Requests also have types and priorities. For example, simulation or user analysis, and high or low, respectively.

As part of its definition, a request may include one or more stages, each of which can apply a sequence of processing steps to the input or output files. Each stage specifies a bootstrap script used by generic jobs to run the relevant applications. The script specifies the requirements on the worker nodes (for example memory) and the maximum number of input files to be issued to the job executing that stage.

The request definition includes a MetaCat MQL query to generate a list of files to be processed in the first stage. This list of files is cached in the central Workflow Database, associated with the first stage of that request. All these files are set to the unallocated state.

The request definition is an input to the Data Management placement agent, which transfers replicas of files to suitable sites as necessary. The location of the replicas of the files is also included in the database, cached from Rucio.

Once the various agents have finished building the request, it can move to the running state and the bootstrap script associated with the first stage will begin execution.

13.5 Grid Workload Systems

The workflow system makes use of the existing global grid infrastructures to deliver jobs for execution on worker nodes at sites. This allows DUNE's jobs to operate alongside those from other experiments in the WLCG without placing additional requirements on sites.

Fermilab operates a global HTCondor pool for DUNE which makes use of the existing OSG Pilot Factory service and HEPCloud to provision execution slots at sites. By using these existing systems we are able to use computing capacity presented with ARC CE or HTCondor CE grid interfaces, or on cloud and HTC services supported by HEPCloud.

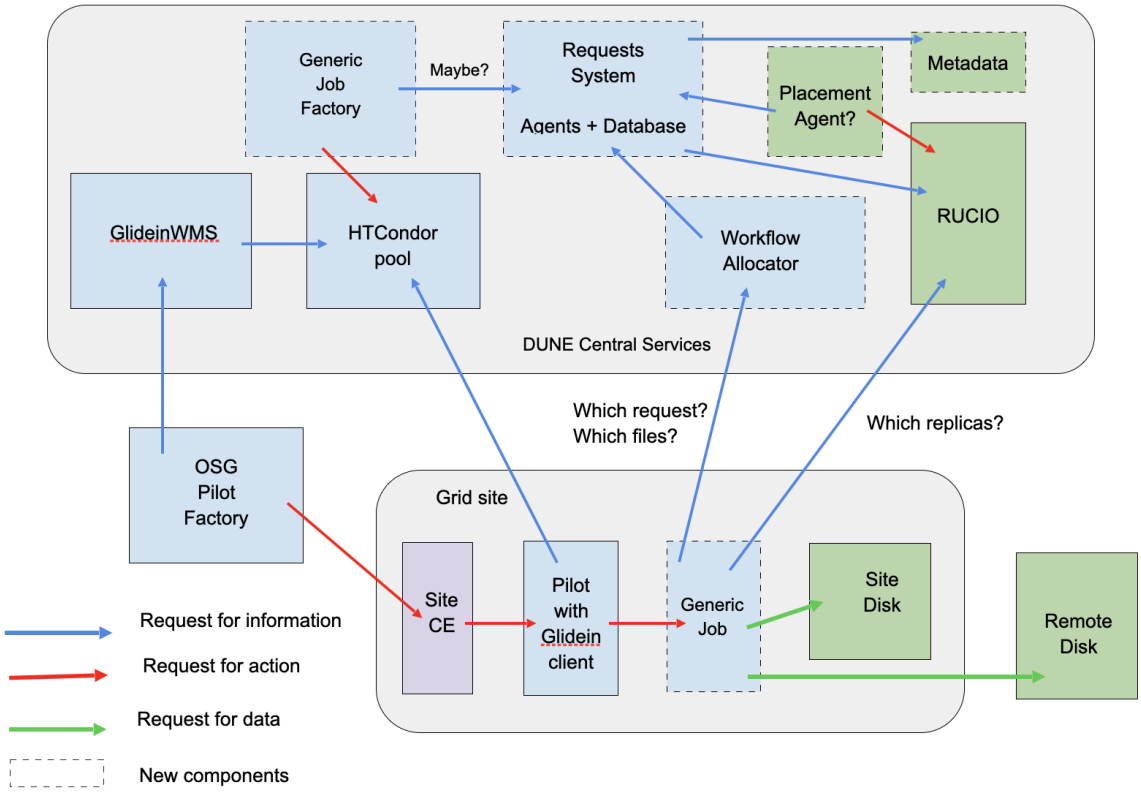


Figure 13.3: Workflow and data management architecture diagram.

13.6 Generic Job Factory

The Generic Job Factory agent creates and submits HTCondor jobs, which are each assigned to a specific execution site. It uses a mixture of matching successes, site limits from DUNE Computing Resource Information System (CRIC), and prioritization of sites to determine how many generic jobs to submit and have waiting at each site. It may also use inspection of the central workflow database to estimate whether more generic jobs should be submitted for a particular site. For ordinary jobs, the job factory must be able to prioritize use of pledged academic and lab sites over commercial cloud sites and specialized sites like NERSC, which are managed by HEPCloud, to allow those services to be used for specialized workflows. Inspection of files waiting to be allocated in the database may be more appropriate for HEPCloud managed sites, so that work requiring their features may be satisfied.

Once a generic job lands on a worker node, it contacts the Workflow Allocator described later which determines which unallocated files from one stage best match that worker node.

13.7 Workflow Database

The central Workflow Database stores definitions of requests and their stages, and cached information about files, replicas, sites, and storage services. Together, all of this information is used by queries to determine what work to carry out where. It is implemented as an SQL relational database.

13.8 Information Collector

The Information Collector agent runs periodically to obtain a list of eligible sites and storage services from CRIC and other information sources, including any downtime notifications. This information includes a table recording whether data access from a particular site to a particular storage is classed as being at the same site, a "nearby" site, or is merely elsewhere on the grid but not blocked by firewalls or policies (for example at an HPC site with no off site access.) The concept of "nearby" is defined as being accessible with an acceptable level of inefficiency.

13.9 Finder

The Finder agent uses the input dataset definition in a request to construct a list of input files for the first stage of that request. Typically this involves making queries to MetaCat to obtain a list of files in the given dataset. This list is cached within the Workflow Database so it can be used as part of further SQL queries. As the files are identified, the location of their replicas are also obtained and cached from Rucio. Again, this allows the proximity of replicas of unprocessed files to be used in deciding what work a job should do.

13.10 Archiver

The Archiver agent has responsibility for removing information about requests, stage, files, and replicas once they are no longer needed, archiving whatever is needed for future reference to long term storage. It is intended that important information about how and where files were processed will be stored in

the MetaCat database by the generic jobs themselves, but some higher level information about the operation of the workflow system and the processing of requests will be saved by the Archiver.

13.11 Workflow Allocator

Once a generic job arrives at a worker node, it contacts the Workflow Allocator service which determines which unallocated file from one stage best matches that worker node. This matching includes the characteristics of the worker node job slot (memory, time limit, etc.), and whether the site is eligible to access a replica of that data file. The matching takes into account that some stage definitions allow access to remote input files anywhere on the grid, and others require files to be at a "nearby" site. Replicas are prioritized based on whether the worker node and replica are at the same site, "nearby", or elsewhere but still eligible.

Details of the request and stage and the bootstrap script to be run are then provided to the generic job. The script can use these details to request a series of files to process with the applications the script invokes. Each input file successfully processed by an application is reported to the Workflow Allocator so that the input file's status can be updated from Allocated to Processed. Unprocessed input files are returned to the unallocated state for processing in another job.

If the stage is not the final stage for that request, each output data file is also inserted into the list of files associated with the next stage for that request, in the unallocated state.

13.12 User Commands

Command line tools are provided to operators and users to allow queries of Workflow Database contents and the creation of requests. These tools are envisaged to be of most use during testing of new workflows and for short workflows during analysis.

13.13 Workflow Dashboard

The same functionality as the command line tools outlined above is provided by a Workflow Dashboard web interface. This allows more sophisticated searches in the Workflow Database to monitor the progress of running requests, and allows members of the operations team to examine the state of the system at the level of individual files and replicas which are due to be processed, or have recently been processed to enable debugging of problems with sites or workflow definitions.

The Workflow Dashboard is also intended to be used for all large scale productions, and allows submitters to draft request definitions, circulate their proposal with colleagues for checking, before submitting them to any formal approval and checking procedure. The lifecycle of a workflow request accommodates manual approval and prioritization of large requests while smaller requests can be approved automatically according to preset limits. A library of bootstrap script templates and requests will be provided as part of the dashboard to help users make sensible choices for common types of workflow.

13.14 Workflow System Prototype

A prototype implementation has been produced, which includes the Workflow Database, Workflow Allocator service, command line tool, generic job factory, finder and information collector, and the monitoring aspects of the Workflow Dashboard.

Using this prototype, we have created requests based on existing productions, populated them with lists of files and replicas held by Rucio, and then processed the files within generic jobs executing the bootstrap script included in the stage definitions, at sites, using matching performed by the Workflow Allocator, and registered the resulting files in MetaCat and Rucio.

13.15 Implementation Plan

During 2022, we plan to continue improve the components of the prototype until it becomes a viable system which can be used for production processing of ProtoDUNE II data. This will allow further comparisons between the functionality requirements of the production team and users, and the Workflow System design.

Part IV

Integration and Evolution

Chapter 14

Services Overview

14.1 Introduction

DUNE computing is dependent on a number of services that are not operated by DUNE itself. Some of these are provided by the host laboratories (CERN and FNAL), such as experiment LAN's and primary raw data storage, others are provided by the various remote sites where computing and data storage are done, and still others are operated by commercial companies and hosted in the cloud.

14.1.1 Computer Security

Computer security has multiple layers, as we work with multiples sites, each with their own security requirements. DUNE has a named computing security officer who is responsible as the contact person for reporting potential bad actors or security problems based on DUNE virtual organization (VO) distributed computing activity. DUNE will maintain traceability for each Distinguished Name (DN) in order to respond to security issues and take appropriate action. The security office is also responsible for ensuring custom DUNE software or services follows good security practices. For common shared tools, patches and security updates are propagated from central maintainers out to WLCG/OSG sites. At larger sites, DUNE relies upon the local admins to monitor, report, and address local security issues that are independent of common services or tools. Overall, this model is very comparable to the models used by other experiments operating on the WLCG/OSG.

14.2 Host Lab Provided Services

DUNE relies on the host laboratories, Fermi National Accelerator Laboratory (Fermilab) and European Laboratory for Particle Physics (CERN), to provide a wide variety of central services. These include web sites (for example, EDMS at CERN and docdb at Fermilab for document management), databases and oversight of cyber security. Compute and storage services are more widely distributed across the collaboration, see 7.1.1 for more detail. The networking services are described in 12. We summarize the other services briefly.

14.2.1 Web Services

The conference-scheduling service (currently Indico), SharePoint, the DUNE Document Database, the DUNE Wiki, and the main `dunescience.org` web page, are all hosted at Fermilab. Fermilab also hosts authentication and authorization facilities such as VOMS, and manages the business relationship with CILogon.org to provide X.509 certificates and WLCG JWT tokens for batch authentication. Fermilab and CERN both provide electronic logbook services and host the web sites for a number of monitoring services.

14.2.2 Database Services

Fermilab maintains the collaboration database, which tracks the membership of the DUNE collaboration, and runs the Frontier Experiments RegistRY (FERRY) database, which records compute permissions for DUNE collaborators. Fermilab hosts the underlying databases for the data management services Rucio and MetaCat (MetaCat), as well as the legacy sequential access via metadata (SAM) workflow management.

14.2.3 Compute Support Services

In addition to the storage and computer services provided by collaborating institutions through their national resources, central coordination and distributions support is needed. Fermilab provides the JobSub service for batch job submission, the Production Operations Management System (POMS) workflow service to submit campaigns, and the GlideinWMS and HEPCloud services to access remote sites including high performance computing and commercial clouds. It provides system administration and hardware maintenance of the various interactive and batch clusters (FermiGrid). It also provides continuous integration facilities (Jenkins), and build service machines.

Fermilab currently provides the main instance for several distribution services, namely CERN VM File System (CVMFS) and its associated Rapid Code Distribution Facility for user code. Streaming of large auxiliary data files is accomplished using the Open Science Data Federation (until recently known as StashCache), which also uses the CVMFS user interface.

Fermilab also maintains the monitoring and log retrieval services for the batch and storage systems, collectively known as FabrIc for Frontier Experiments MONitoring (FIFEMON).

Fermilab is assisting DUNE and other experiments to transition to the Spack[1] code packaging system. Significant effort is also provided on managing the art framework and Liquid Argon Software (LArSoft) project.

14.2.4 Storage Services

Fermilab and CERN provide the primary archival tape storage for raw data, currently delivered through the Enstore and CERN Tape Archive (CTA) tape library systems. A tape copy of reconstructed and simulated data is also currently maintained at Fermilab and in future, at several collaborating sites in Europe. At Fermilab the dCache disk caching system is the front end to this system, and also provides some standalone disk. Fermilab, Brookhaven National Laboratory (BNL) and CERN also provide a

[1] Spack©, https://spack.io/

number of data management and data movement services, including CERN-FTS, Fermi-File Transfer Service, SAM, Rucio, MetaCat, and the Data Dispatcher.

14.3 Collaboration Contributed Services

DUNE receives compute and storage resources from a number of sites around the world, not all of which are formally DUNE collaborators, as detailed in the section on the Computing Contributions Board Section 2.3.

CERN provided services include: the CERN Indico, the engineering document management system (EDMS) document management system, the MONIT monitoring system, Computing Resource Information System (CRIC) - which is the master list of all DUNE compute and storage resources as well as the means to track allocations, and the Experiment Test Framework (ETF) testing service which routinely tests all remote compute sites on DUNE's behalf.

While CERN and Fermilab remain the major providers of support services, some monitoring and control activities are moving to collaborating institutions, notably the Rucio monitoring, now hosted by Edinburgh and the new workload system at Manchester. We anticipate a move to an even more distributed system as the experiment evolves.

14.4 Cloud-Hosted Services

DUNE uses the GitHub service for its code management, the Slack and Microsoft Teams services for interactive communication, the Overleaf service for editing LaTeX documents, and the Zoom teleconferencing application. The cloud-hosted ServiceNow application is used for communication between DUNE liaisons and the two Fermilab Computing Divisions about outages and changes to service and also for internal DUNE use in tracking of issues and workflow requests such as production requests and data movement requests.

Chapter 15

Information Systems and Monitoring

This section describes the monitoring of services and resources on the grid. The actual monitoring of individual jobs that are submitted by DUNE is provided as part of the Workflow System (see Chapter 13).

As the global infrastructure is broadly similar to that used by the current LHC experiments, DUNE plans to reuse the relevant monitoring tools (Experiment Test Framework (ETF), performance Service-Oriented Network monitoring ARchitecture (perfSONAR)) developed for this purpose by the Worldwide LHC Computing Grid (WLCG) which we cover in this section.

Going forward, DUNE plans to be agile, keep up with upcoming developments in this field, and develop new tools as needed. For example, the widespread deployment of IPv6 [144] networking may require re-evaluation of some of our computing tools. The ETF and perfSONAR tools are already IPv6 compatible and can be deployed to test resources offering IPv6 connectivity in either pure or dual-stack mode.

15.1 Tools

15.1.1 CRIC

DUNE intends to use Computing Resource Information System (CRIC) as a central source of information about sites and their compute and storage services. CRIC is routinely used by ATLAS and CMS with Open Science Grid (OSG) and other WLCG resources, and is also familiar to site administrators.

An evaluation of CRIC is being done, with information about sites obtained from the configurations of the OSG pilot factories. This is browsable via the CRIC web dashboard, and also used to generate XML files in the virtual organization (VO) Feed format required by the ETF testing framework used for monitoring.

15.1.2 Experiment Test Framework (ETF)

DUNE jobs run at a number of computer data centers across the grid. Checking the status of these jobs is one way of keeping an eye on whether a given site or resource is functioning normally. Given that issues can and usually do have multiple sources, it is essential to have an independent resource monitoring system to monitor and catch issues quickly.

ETF[145, 146] is a system of tools hosted by CERN that regularly test all the available resources for different experiments, i.e., VOs. It has been developed and run on behalf of the WLCG Virtual Organizations (VOs) for well over a decade. The ETF framework runs tests customized to the VO's application and systems to provide a true picture of the available resources and their reliability. The ETF tests run hourly and feed into the MONIT framework, which keeps history information and the most recent test logs to enable debugging and metric measurement.

The ETF framework currently involves

1. high-level functional testing of about 90 hosts defined in the GlideinWMS configuration;

2. dashboard (checkmk) to show results; and

3. plugins conforming to Nagios[1] standard.

For DUNE, customized tests are in place to verify a minimal simulation functionality on the worker nodes and to check that a given worker node has IPv6 networking. A customization to perform a lightweight test of access to Rucio servers from the worker node is in progress. Beyond this, DUNE-specific development will be required from time to time to synchronize with updates to the framework both from the ETF / MONIT and the DUNE software ends.

15.1.3 PerfSonar

Data distribution requires both high-performance networking between the sites and sophisticated debugging tools to identify and solve problems that arise. The WLCG network monitoring and debugging infrastructure that DUNE will adopt and deploy is perfSONAR, an open source toolkit collaboratively developed by groups in Europe and the Americas that keeps a clean separation between network-related and other metrics. It is installed on dedicated hardware (perfSONAR boxes) at each resource site. The data from these perfSONAR boxes can be aggregated by a variety of tools and algorithms to visualize the connectivity in different ways, as described in in various (kibana, grafana, checkmk, ...) ways, enumerated below.

1. `https://psetf.opensciencegrid.org/etf/check_mk/index.py`

2. `https://toolkitinfo.opensciencegrid.org/`

3. `https://monit-grafana-open.cern.ch/`

4. `https://atlas-kibana.mwt2.org/s/networking/app/dashboards`

[1] Nagios®, `https://www.nagios.com/`

5. https://perfsonar.uc.ssl-hep.org/

6. https://sand-ci.org

These tools are currently well supported within WLCG with expertise shared widely and regularly among administrators. Given that networking is a core need for WLCG as it is for DUNE, it is anticipated that perfSONAR will continue to be supported as long as DUNE requires it.

15.2 FIFEMON

DUNE relies on the FabrIc for Frontier Experiments MONitoring (FIFEMON)[147] family of monitoring tools which have been developed at Fermilab over the past decade or so. These dashboards show the CPU usage and efficiency and storage usage and traffic of all the experiments on a user by user basis.

The dashboards are built on open-source technology, and use common ingest tools such as Apache Kafka, Logstash, and Prometheus to collect data, the Graphite and ElasticSearch data stores to store it, and the Grafana and Kibana display utilities to display it. Grafana is used for static displays, while Kibana has an active search function in which custom graphs can be made using a query language.

There is information on batch usage by user and aggregate of the whole experiment. CPU efficiency and memory usage are plotted. In addition storage total usage as well as the volume of reads and writes, as well as the age of files, are visualized. We also have custom visualizations to display data movement from data center to data center and

Chapter 16

Authentication and Authorization

16.1 Obtaining Access to DUNE Computing

Computing access for collaborators is a multi-step process. New collaborators are first proposed by institutions (with senior collaborators requiring approval by the institutional board). They are then added to the collaboration database. Once this is done, they may apply for access to computing resources at Fermi National Accelerator Laboratory (Fermilab), and the DUNE secretariat verifies that they are members of DUNE. The central repository Frontier Experiments RegistRY (FERRY) is used to populate the user list for the Fermilab interactive login machines and to determine if a user can obtain DUNE Grid X.509 credentials. When a collaborator's Fermilab ID expires, or they leave the DUNE collaboration, their access to computing is automatically removed.

While the ProtoDUNE experiments NP02 and NP04 are running at the European Laboratory for Particle Physics (CERN), collaborators working on those activities may apply for interactive computer access at CERN, as well.

16.2 Current State of Authentication and Authorization

Interactive login to Fermilab is strongly authenticated via Kerberos 5 credentials. DUNE collaborators must be currently registered to have accounts at Fermilab.

Authentication for batch submission and access to storage is currently done via X.509 certificates and proxies on the Open Science Grid (OSG) in the U.S. and Worldwide LHC Computing Grid (WLCG) Computing Grid (WLCG) elsewhere in the world. Automated processes at Fermilab dynamically generate X.509 certificates on behalf of the user at the time of batch submission, from the CILogon-Silver certificate authority[1]. This authentication is used to submit batch jobs both to the local Fermilab clusters known as FermiGrid and to the rest of the distributed computing sites. The proxies are also used to access files stored on the global storage elements around the world. X.509 certificates are issued by members of the International Global Trust Federation (IGTF). They are then verified by the DUNE

[1] `https://www.cilogon.org/news/cilogonsilverenablesfederatedaccesstoopensciencegrid`

VOMS server. Certificates issued by other IGTF authorities can be associated with a user by request.

Access to DUNE web sites hosted at Fermilab is controlled by the Fermilab Single Sign-on (SSO) facility. Some sites (e.g., the DUNE wiki) require a Fermilab SSO. Some sites, e.g., document database (DocDB), require either a Fermilab or CERN SSO or an X.509 certificate. Both of these applications also currently allow read access via a group username and password.

16.3 Planned Changes to Authentication Currently Under Way

The OSG and WLCG are both in the process of changing over to JWTs for authentication of batch jobs and storage. This authentication protocol has significant adoption in industry. A common schema has been agreed on between the various issuing organizations. Fermilab has set up a service to issue these tokens on behalf of DUNE and early tests have successfully demonstrated access to storage elements and batch job submission with these tokens. The token issuer is populated with information from the FERRY server. The tokens include a unique user identifier and a list of capabilities that the user is allowed to have, including which areas in the storage element the user is allowed to write. The basic token infrastructure is available now but we expect a phased transition beginning in 2022, during which the X.509-based authentication infrastructure will continue to be available for some number of years.

Interactive web sites will continue to use Fermilab SSO for the foreseeable future, while gradually allowing more Identity Providers. It is likely that JWT tokens will be used to access various non-interactive web services besides the compute and storage.

16.4 Requirements for Authentication and Authorization

DUNE is an international collaboration and it is important that all collaborating scientists be able access to the data, codes and documents that result from our common efforts. To this end DUNE Computing Management has submitted to Fermilab the following requirements:

- DUNE collaborators without a Fermilab SSO must be able to see and edit internal DUNE web pages and submit to the collaboration document server.
- Computing staff who work at collaborating institutions but who are not themselves DUNE collaborators must be allowed access to the computing documentation.

There are a number of technical solutions for federated web identity, including the Edugain federation of identity providers. For now, we are working to extend access to collaborators with CERN credentials. Fermilab is responding, and as of this writing, users with CERN credentials can now use those credentials to access the DUNE DocDB. Fermilab is in the process of federating its SharePoint instances with CERN, and the DUNE wiki and Indico sites are next in line.

Chapter 17

Code Management

17.1 Liquid Argon TPC Code Management

Software for simulation, data read-in, and reconstruction has a great deal of commonality across the liquid-argon time projection chamber detectors planned to be deployed by the DUNE collaboration. These include the horizontal- and vertical-drift far detector modules, the 35-ton prototype, ProtoDUNE-SP, ProtoDUNE-DP, ProtoDUNE-2-HD, ProtoDUNE-VD, ICEBERG, and the pixel near detector (ND) and its prototypes. The same software stack used by DUNE and the protoDUNE detectors is used for wire and pad readout, the photon detectors, and the cosmic-ray taggers used in some of the prototypes.

Despite the commonalities, the interfaces to event generators, Geant4, art framework and the data handling systems require effort to build and maintain, and data products need to be thought out carefully. To reduce the development burden and expand the pool of expertise, DUNE has chosen to use the Liquid Argon Software (LArSoft) toolkit, which is built on NuTools and art.

LArSoft is supported by Fermi National Accelerator Laboratory (Fermilab)'s Scientific Computing Division (SCD) and is shared among several other experiments, such as ArgoNeuT, MicroBooNE, ICARUS and SBND. The pool of potential developers is quite large, with a variety of software experience and length of time commitment to maintenance. Substantial effort goes into maintaining coherence across the project with new releases of LArSoft made weekly. DUNE's LArSoft-based code stack follows a similar release schedule.

DUNE's LArSoft-based software historically had been collected in a single git repository called dunetpc, starting in the early days of the collaboration. This repository was hosted using Fermilab's Redmine service. Compiling all of the code in this single repository became slower as more source files were added. In January 2022, a split of the dunetpc repository into ten smaller repositories was deployed on GitHub, taking advantage of GitHub's superior performance, feature set, and openness compared with Redmine. The old dunetpc repository is kept in a read-only state in Redmine for inspection purposes. The new top-level product is called dunesw. A UNIX product support (UPS) dependency graph made in January 2022 is shown in Figure 17.1, showing the dunesw stack, the LArSoft UPS products, and their dependencies. The January 2022 split of the DUNE components of the stack helps speed build

times for users who only are interested in modifying a small portion of DUNE software at a time. The division of code into several repositories also helps define the boundaries of maintenance responsibility.

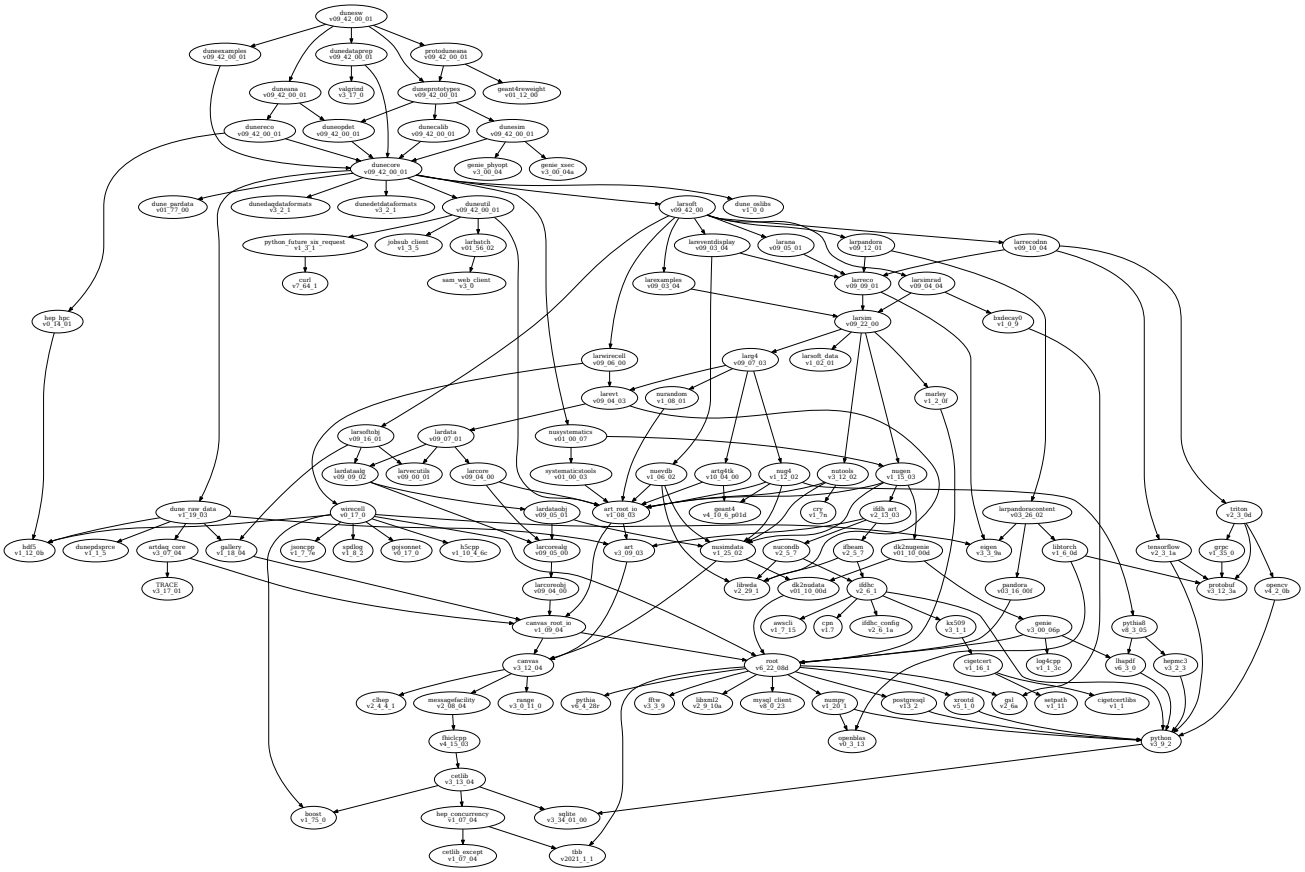


Figure 17.1: UPS dependency graph for the dunesw software stack, for version v09_42_00_01, January 2022. DUNE-specific products have "dune" in their names, and LArSoft product names begin with "lar". External products, such as ROOT, Geant4 and python are also shown. This diagram is an aid to users and managers who need to find code, as well as check out and build portions of it while setting up the remainder from pre-installed images.

Since the dunesw stack depends on LArSoft and sets it up via UPS, dunesw also has a weekly release schedule, and releases are tied to LArSoft's releases and use the same version tag names. LArSoft's pull-request system, based on the one used by the CMS experiment, performs automated checks that the proposed new code compiles and tests properly. Once the tests are successful, a review of the proposed new code is required before it can be merged into the repository. As of this writing, DUNE is considering moving to a pull-request model similar to that of LArSoft.

Code maintenance is required, even for unmodified algorithms, when the external environment changes, such as a compiler or build-system upgrades. DUNE software managers have maintained legacy software when this occurs, though improvements and additional features to be added to software require stakeholders to perform the work. Code that is no longer needed by DUNE may be deleted in order to speed build times, reduce maintenance effort, and save space. Deleted code is always present in previous installed releases and in the git repository in case it needs to be recovered or used as an example for

writing new software.

The build system used is `mrb`, which is provided by and supported by Fermilab's SCD. Software is built in UPS products which are distributed to the collaboration in two ways. The primary distribution mechanism is via a set of installed products in the CERN VM File System (CVMFS) which can be directly set up by end users of Scientific Linux 7 and CENTOS 8 without the need to install anything locally, except CVMFS itself. CVMFS maintains a cache on the user's computer to store copies of the released software files for repeated local access. CVMFS is also used to distribute released software to batch worker nodes. The second software distribution mechanism is via the `scisoft.fnal.gov` web server. When a release is made, the repositories are tagged and the release manager triggers Fermilab's Jenkins build servers to compile the software and install auxiliary files in UPS products. The UPS products are then tarred up and the tarballs are uploaded to the scisoft web server, along with a manifest file that describes which tarballs need to be downloaded and installed to get a fully functioning dunesw software stack.

In addition to code, some data files also are distributed via CVMFS. For example, photon lookup libraries are stored in several-hundred-megabyte ROOT files which are inconvenient to store in a git repository. They are loaded into a special repository in CVMFS and scisoft called `dune_pardata`. Even larger files are stored in StashCache which has a CVMFS interface for the user, but which retrieves files out of a dedicated area in dCache.

As operating systems have become more secure, some of the ways of distributing, setting up, and running HEP software, such as the use of the environment variables `LD_LIBRARY_PATH` and `DYLD_LIBRARY_PATH`, have conflicted with security features of at least one operating system. We are evaluating Spack as a replacement for `mrb` and UPS. Spack is a build system that also allows users to select from a list of installed versions, like UPS does. UPS is now approximately 25 years old, it is not an industry standard, and it is somewhat linked to Fermilab. Spack has a significant community using it outside of HEP, and thus there is a large volume of documentation available for it on the web. The DAQ group is already using Spack for online systems.

17.2 Near Detector Code Management

When DUNE starts running with beam, the expected ND configuration will consist of a liquid argon time-projection chamber (LArTPC) with pixel readout (ND-LAr), a magnetized steel muon spectrometer (The Muon Spectrometer (TMS)), and an on-axis beam monitor, System for on-Axis Neutrino Detection (SAND). A proposed future upgrade is replacement of the TMS with a subdetector that consists of a gaseous argon time-projection chamber (GArTPC) with pixel readout, an electromagnetic calorimeter (ECAL) and a muon system (ND-GAr). The software structure reflects the detector configuration.

The software contributions for each subdetector come from the groups that are designing and proposing to construct the subdetectors and their subsystems. Below is a summary of the code management strategies used for each subdetector, and also for the shared tools. In general, official repositories are hosted in GitHub with executable binaries and the associated libraries installed in CVMFS so that the software can easily be version controlled and run on the grid. These requirement must be met in order for software to be run at the production level and stored in group accessible areas.

17.2.1 ND-LAr Code Management

The ND-LAr code, larnd-sim and ndlarcv, is hosted in GitHub. This code handles the standard reconstruction and detector response mock-up. The ML and Pandora based reconstruction code is currently under private development, but will be made available this spring prior to production integration.

17.2.2 TMS Code Management

The TMS code, dune-tms, is hosted in GitHub. This code handles the reconstruction and detector response mock-up.

17.2.3 ND-GAr Code Management

The software for simulating and reconstructing events in ND-GAr is, at the time of writing, contained in two github repositories: Gaseous Argon Software (GArSoft) and GArAna. GArSoft builds on the functionality of art and NuTools, and it is maintained as the art API changes and when the build system is upgraded. Following LArSoft's pattern, it is built with `mrb` and set up with UPS. GArSoft functionality is described in Sections 3.5.4 and 3.9.3. GArAna provides a facility for making an analysis ntuple from information stored in GArSoft data products. Both repositories are hosted in GitHub. Continuous integration has not yet been set up for GArSoft and GArAna. Executable binaries and the associated libraries are built using Fermilab's Jenkins build servers, and the build artifacts are installed in CVMFS so that the software can easily be run on the grid.

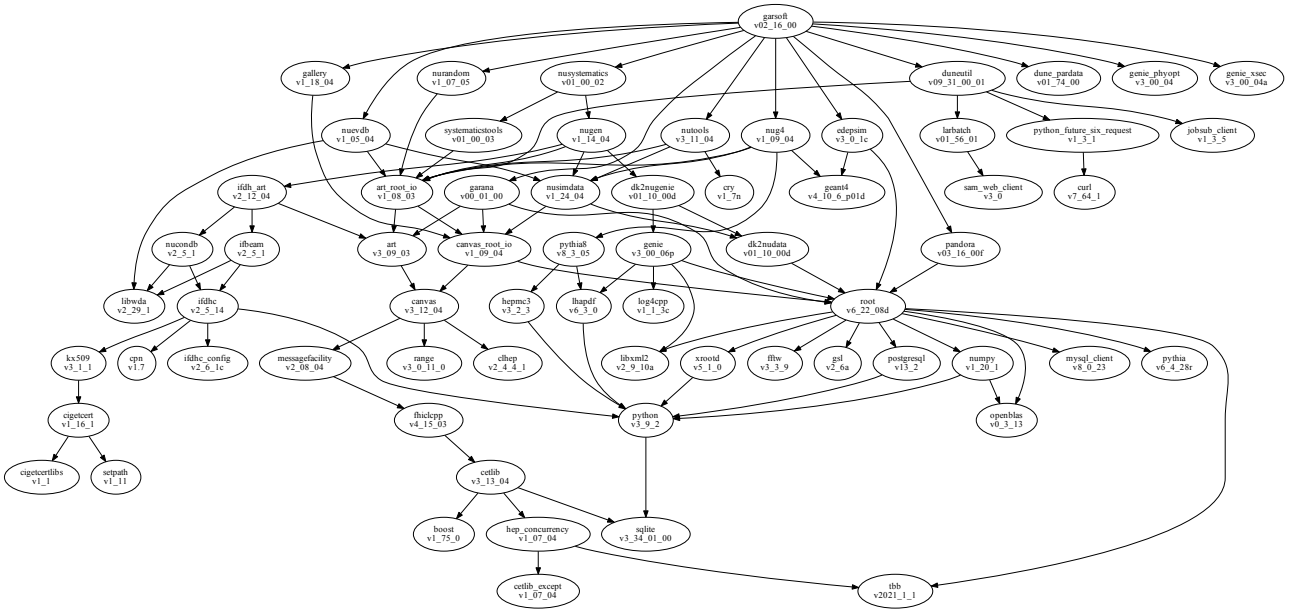


Figure 17.2: Dependency graph for the GArSoft software stack, for version v02_16_00, current as of March 2022.

17.2.4 SAND Code Management

The SAND code, ND-SAND-FastReco, is hosted in GitHub. This code handle the basic reconstruction and detector response mock-up. The full reconstruction and detector response code is currently under private development, but will be made available this spring prior to production integration.

17.2.5 Near Detector Common and Production Tools

A repository has been set up in GitHub to manage files that are needed for ND production. These include production scripts, detector geometry descriptions and flux specifications. It is versioned and installed in CVMFS like other DUNE repositories, so that production workflows can be reproduced at later dates. Additionally, common ND geometry code is stored in GitHub, dunendggd.

17.3 Continuous Integration

The stability of the functionality and performance of a large base of shared software requires attention to each change that is committed. Software releases require validation and approval before being used in analyses intended for publication. The rapid pace of software development early in the life cycle of the experiment, as well as increased activity as data are first collected and conferences come up, requires constant vigilance to ensure that bugs are not introduced and that software remains backwards-compatible.

To meet this requirement, an automated continuous integration (CI) system is currently in operation for the LArSoft-based code base. Similar systems, even using the same infrastructure, will be deployed for the ND and beam simulations system. The CI system consists of a set of servers that monitor commits pushed to the central code repositories. On each commit, a suitable delay of approximately 15 minutes allows aggregation of commits. The CI system can also be triggered by user interaction, via an authenticated request. Once triggered, the CI system compiles the changed code as well as any dependencies that are required. Currently, the CI system builds LArSoft and all experiment code from the head of the develop branch in the repositories.

The status of the build is stored in a logfile and summarized on a web page. If a commit causes the build to fail, software managers and the person who committed and/or pushed the commit are notified and the commit is blocked from being merged into the head of develop. LArSoft currently implements a pull-request model, in which experiment-appointed Level-2 managers comment on and sign off on changes to the central code, and Level-1 managers perform the actual merging. A proposed change will not even be sent out for approval unless the CI system can build it and validation tests are run to compare output that ought not to have been changed by the new code.

This second step, physics output validation, requires a more lengthy run through both unit tests and integration tests that run simulation and reconstruction workflows. These tests run on standard input files and have their random number seeds fixed to constants. The outputs are compared with reference histograms and a web page summarizes the comparison of the output logfiles, physics histograms, as well as run times and memory consumption, all of which are available on a web site that monitors these tests. History plots of variables such as run time and memory consumption are made available on the monitoring web site so that investigations of when the memory usage of a job jumped up can be

done without laboriously checking out, building, and running the software in the suspected timeframe of interest to find a particular change.

While the CI system is designed to ensure basic software quality as it is developed and checked in, the final validation of a software release must be performed and approved by the Deep Underground Neutrino Experiment (DUNE) collaboration. This process will involve significant person-power and it cannot be fully automated. At the time of writing, formal procedures are not yet in place, though it is expected that the physics coordinators and the relevant physics working groups will perform these tasks. Reviews of the physics performance of software releases must be scheduled with enough lead time for problems to be fixed, calibrations recomputed, re-approvals performed as needed, and samples generated for conferences and publications. The reviews must be open to the entire DUNE collaboration. The criteria for acceptance of a software release will depend on the target precision of the physics intended to be performed with it. Approved releases should be accompanied with documentation describing the quality of the data modeling, calibrated efficiencies, and recommended systematic uncertainties. Sufficient resources need to be allocated to physics working groups performing validations and calibrations so that they can achieve the goals on the required timescales.

Chapter 18

Training and Documentation

The DUNE collaboration has grown rapidly since its inception, and it currently has members from over 200 institutions. Even if the size of the collaboration were to remain constant, new members will always be joining while others move on to other positions and projects. The scale of the experiment demands a harmonized documentation effort, coupled with consistent training for both newcomers and existing users. Although treated separately in this section, the tasks of documentation and training are highly correlated. Both are crucial for the long-term success of DUNE.

18.1 Documentation

The documentation related to the DUNE's computing aspects will be accessible on a variety of platforms, each with specific goals and access policies.

One evolving issue is the move to tighter control of public-facing websites at Fermilab and elsewhere. Public facing websites are now subject to review before posting and this has led to much of the DUNE documentation becoming non-public except via single sign-on (SSO) access. Fermilab and CERN are working to make both Fermilab and CERN trusted credentials provide access to DUNE documentation but useful tools, such as the ability to find documents using google searches are becoming less and less available as documentation moves to protected areas. This has become an issue in the preparation of this document, as it is good practice to reference the many DUNE documents that support this report but are no longer outside of the SSO.

18.1.1 Wikis

The existing DUNE wiki [148] provides the landing page and starting point for "DUNE Computing." It acts as a portal gathering information and links about the DUNE Computing consortium groups, their activities, and related resources. The template was revamped in early 2021 with a "block" design for a more visual and compact layout. As with the rest of the DUNE wiki, access is restricted; users need a Fermi National Accelerator Laboratory (Fermilab) or CERN, Single Sign-On (SSO) service domain since the wiki contains information about computing access, node names, and other sensitive information.

The top level DUNE wiki has six main blocks: Organization, Computing Toolbox, Operations, Working Groups, Getting Started and Resources.

- The Organization block covers the consortium, the calendar of meetings, and lists the associated collaborations.
- The Computing Toolbox collects links to data access, analysis tools and algorithm frameworks.
- The Operations covers the operations groups.
- The DUNE Computing working groups are listed on the Working Groups area, and conveners of each working group will maintain their page.
- Getting Started has pointers to the tutorials, training sessions, and how-tos as well as the DUNE Slack channels and ServiceNow helpdesk.
- Last but not least, the Resources block gathers links about data policies and preservation, information protection, and the documents in progress.

18.1.2 Redmine

Fermilab's Redmine service was historically used to host DUNE far detector (FD) and ProtoDUNE software, wikis, and issue tracking. Prior to 2021, read access to DUNE Redmine was open to the public, but write access required authentication. Starting in the summer of 2021, all access to Redmine requires authentication. As of January 2022, the DUNE FD and ProtoDUNE software repositories and associated issue tracking have been migrated to GitHub.

Many of the other Fermilab Intensity Frontier Experiments and much of the general Fermilab software documentation still resides in Redmine.

18.1.3 Code Documentation

One of the challenges for any large experiment is documentation of the algorithms and software. Following the lead of the Belle2 experiment, DUNE plans to explore the implementation of the Sphynx[1] package for the automatic generation of code documentation. The generation of in-code comments and dedicated text files utilized by Sphynx will be the responsibility of code librarians and contributors of each sub-repository within the dunesw code stack.

18.1.4 GitHub

DUNE software repositories have migrated to the version-controlled platform GitHub. GitHub offers a ticketing system to facilitate debugging, revisions and updates from the community of users, though DUNE does not yet use GitHub's wikis extensively. The art and Liquid Argon Software (LArSoft) teams have transitioned from the Redmine system to using GitHub's ticketing system. The work to make this transition for DUNE is underway and follows the guidance of the HEP Software Foundation (HSF) community. The goal is for code to be publicly readable but protected from modification.

[1] https://www.sphinx-doc.org/en/master/

18.1.5 Code standards

DUNE does not yet have Deep Underground Neutrino Experiment (DUNE)-specific coding standards. We use the LArSoft standards which are documented at:

https://cdcvs.fnal.gov/redmine/projects/larsoft/wiki/The_rules_and_guidelines

and a longer list, including documentation and plug-ins and error handling:

https://cdcvs.fnal.gov/redmine/projects/larsoft/wiki/Developing_With_LArSoft#Guidelines

18.1.6 Frequently Asked Questions

We are actively looking for a reliable Frequently Asked Questions (FAQ)system. We are currently using Github issues and projects as a place holder while we identify and implement a permanent solution.

18.2 User support

In addition to formal documentation, users often have questions, either how to do things or problems with systems. DUNE currently relies on two systems to support users, Slack and ServiceNow.

18.2.1 Slack

DUNE uses the commercial Slack platform to communicate urgent user questions. The Slack systems is staffed by other users, including collaboration experts, who can provide quick answers to simple questions.

18.2.2 Service Now

ServiceNow is the official Fermilab issue reporting system. Problems, including operational issues that cannot be resolved with peer support through Slack are directed to the ServiceNow system.

18.3 Training

18.3.1 Goals of DUNE Training

DUNE's training aims to serve both newcomers and existing users, offering a smooth start for the former and continuous support for the latter. DUNE recognizes that the computing environment and tools used within a HEP experiment are unique and evolve over time, and thus they require specialized training. The goals are to teach the basics of the environment and software used for analysis, as well as best practices in programming and data management. The training is offered through various formats, tools and platforms, as well as partnering with other collaborations, all discussed below.

18.3.2 Training Sessions

The primary training for new DUNE members is done twice a year during dedicated sessions that extend over several days. These sessions run before or after collaboration meeting weeks. Such timing secures

a good attendance for both new collaborators and presenting experts. Since 2020, the tutorials have been given online via zoom over several 1/2 day sessions to accommodate multiple time zones. In particular, DUNE has a significant population of young scientists from India who cannot participate in person.

The format of the training is an alternation of lectures, where students can follow running commands themselves, with interleaved quizzes and hands-on sessions. Introductory instructions and homework exercises are sent to participants prior to the tutorials to ensure that trainees arrive able to access the computing resources and are prepared for the material.

The goal of this training is to make certain that new people have access to DUNE computing resources (at either FNAL or CERN) and understand the basics of logging in, storage areas, running applications, making minor modifications to code and submitting batch jobs.

All materials and documentation from past tutorials are retained in github and prominently linked on the DUNE wiki to serve as a library of self-education material.

However, due to the vast suite of software used by the DUNE collaboration for the various analysis steps and sub-detectors, additional training is needed and is currently provided on an ad-hoc basis by physics and hardware groups.

18.3.3 Training Tools

Training materials are now hosted in GitHub using the Software Carpentries framework. [149]. Google-Docs are used for anonymized real-time questions by the participants. Most participants use the Fermilab general purpose DUNE General Purpose Virtual Machines (dunegpvms), although materials for everything except the batch submission parts can be run at CERN (or even local clusters with CERN VM File System (CVMFS) access).

Experts from the training team provide written answers to the questions as they come in and the responses remain available from the Tutorial github page. The sessions are delivered over zoom and recorded, with edited videos posted to the github tutorials so that they can be reviewed later.

In the future, DUNE hopes to use Jupyter[2] notebooks, which provide an interactive environment for the trainees to run the examples, complete the exercises, and tweak portions of code to deepen their understanding of the analysis software. These notebooks will be accessible through JupyterHub servers provided by Fermilab and CERN, as well as by other hosting institutions equipped with computing resources and analysis facility tools. The notebooks will be archived and referenced on the github pages as self-training modules for newcomers as well as reference material for lecturers.

18.3.4 Audience for training

As part of the 2022 training, incoming participants were surveyed about their level of experience and needs. 35 out of 50 participants responded. We found that:

[2] Jupyter© `https://jupyter.org/`

- 50% of participants are graduate students, 25% are postdocs with the rest divided between undergraduates (8%), senior researchers (8%) and engineers and programmers.
- Almost all participants were already associated with some DUNE hardware effort.
- 16/35 (46%) were not yet associated with a physics analysis group.
- 31/35 (88%) were familiar with C++
- 29/35 (83%) were familiar with ROOT
- 28/35 (80%) were familiar with python
- 26/35 (74%) were familiar with github
- 19/35 (54%) were familiar with Jupyter notebooks
- a smaller number were familiar with specialized HEP codes such as geant4 and LArSoft.
- 31% had no experience with batch environments while another 29% were not confident in their abilities.
- 16 (46%) were already using Fermilab computing resources while 6 (17%) were using CERN facilities.
- Unfortunately 6 (17%) indicated that they did back up their code in response to a question about backup methods.

18.3.5 Partnering with other Collaborations

The HSF has started a continuous effort of harmonizing their tutorials under a common template called "training module" to create an introductory curriculum with the basic set of software needed to instill good programming practices right at the start. Several of their modules for beginners are offered by the Software Carpentry Foundation[3]. The growing list of modules can be found on the HSF website "Towards a HEP Software Training Curriculum" [4].

The DUNE Computing training group is working in collaboration with the HSF. The DUNE training points to the HSF's curriculum page so that newcomers can learn the prerequisites before attending a DUNE training. This in turn provides an audience for the HSF material, providing that organization valuable feedback.

18.3.6 From Trainee to Mentor, User and Lecturer

A key aspect of the success of the HSF workshops is its mentoring system. The mentors are alumni of one of the recent training sessions, typically young, easy-to-approach researchers. After a couple of training sessions, the mentors can be trained by experts to become lecturers, acquiring new skills with their favorite software. The DUNE Computing consortium plans to reproduce the HSF mentoring and training scheme by issuing frequent calls for mentors within the collaboration.

18.3.7 Future Formats

The DUNE Computing training group is also committed to surveying the needs and demands for particular skills the members would like to learn. Recent surveys have shown that LArSoft and Pandora

[3] https://software-carpentry.org
[4] https://hepsoftwarefoundation.org/training/curriculum.html

are popular topics, as are best practices for building analysis code, and machine learning techniques, such as Keras, Tensorflow and GPU libraries. The breadth of the tutorial wish-list is too large to efficiently cover during one bi-annual DUNE training, but continued input from tutorial attendees allow the planning of future topic specific lectures and tutorials. Moreover, incoming students are each working on a specific aspect of the DUNE project. A project-based approach such as a series of hackathons would be an ideal format for some of these more advanced topics. This would allow students to team up on an aspect directly linked with their analysis, guided by an expert.

We anticipate that the computing project will continue to provide infrastructure and support but not all content for algorithmic documentation by the physics groups.

Part V

Resources and Conclusions

Chapter 19

Resource Needs Summary

19.1 Hardware Resources

Chapter 6 describes the projected compute and storage needs through 2040; these resources are a shared collaboration responsibility and are requested and monitored via the Computing Contributions Board (CCB), described in Section 2.3. Chapter 12 describes networking needs and requirements. Our best estimates, developed in Chapter 6, indicate that DUNE's hardware resource needs are similar in terms of specifications to CMS or Atlas over the same time scales, but of order 10% of their size.

19.2 Software Development and Operations Resources

Building and managing a project of this complexity requires a large number of dedicated people. To reduce our development resource needs relative to the large LHC experiments, DUNE's strategy is to use common tools already developed for HEP wherever possible and to work collaboratively with other groups in development of new tools. Much infrastructure is already in place, but significant DUNE-specific effort will still be needed. Section 2.1 describes the top level organizational structure while Figure 19.1 shows a timeline for estimated human resource needs. Efforts are currently spread across a large number of highly expert people working part-time on DUNE computing, adding up to around nine FTE/year, in addition to the five dedicated DOE-funded postdocs. We anticipate that these early-career people will move into leadership roles in HEP computing in the future and will need to be replaced as they move on. In addition to the core group of experts, a small number of collaboration volunteers assist with operational tasks such as setting up and running simulation and reconstruction campaigns. Our ability to make use general collaboration members in operations will grow as we move from development to stable operations.

Although the current expert FTE number of $\sim$14 is close to the identified need for dedicated experts, there is some mismatch in skill sets. In particular, we anticipate needing several person-years of effort on framework development, after ProtoDUNE-II but well before the near and far detectors are operational, and this expertise is largely missing from the current mix of personnel. In addition, most of the existing expertise is either funded short-term through grants or is provided by professionals at facilities who also have other responsibilities. Long term, the project will require more stable funding and well defined

institutional commitments.

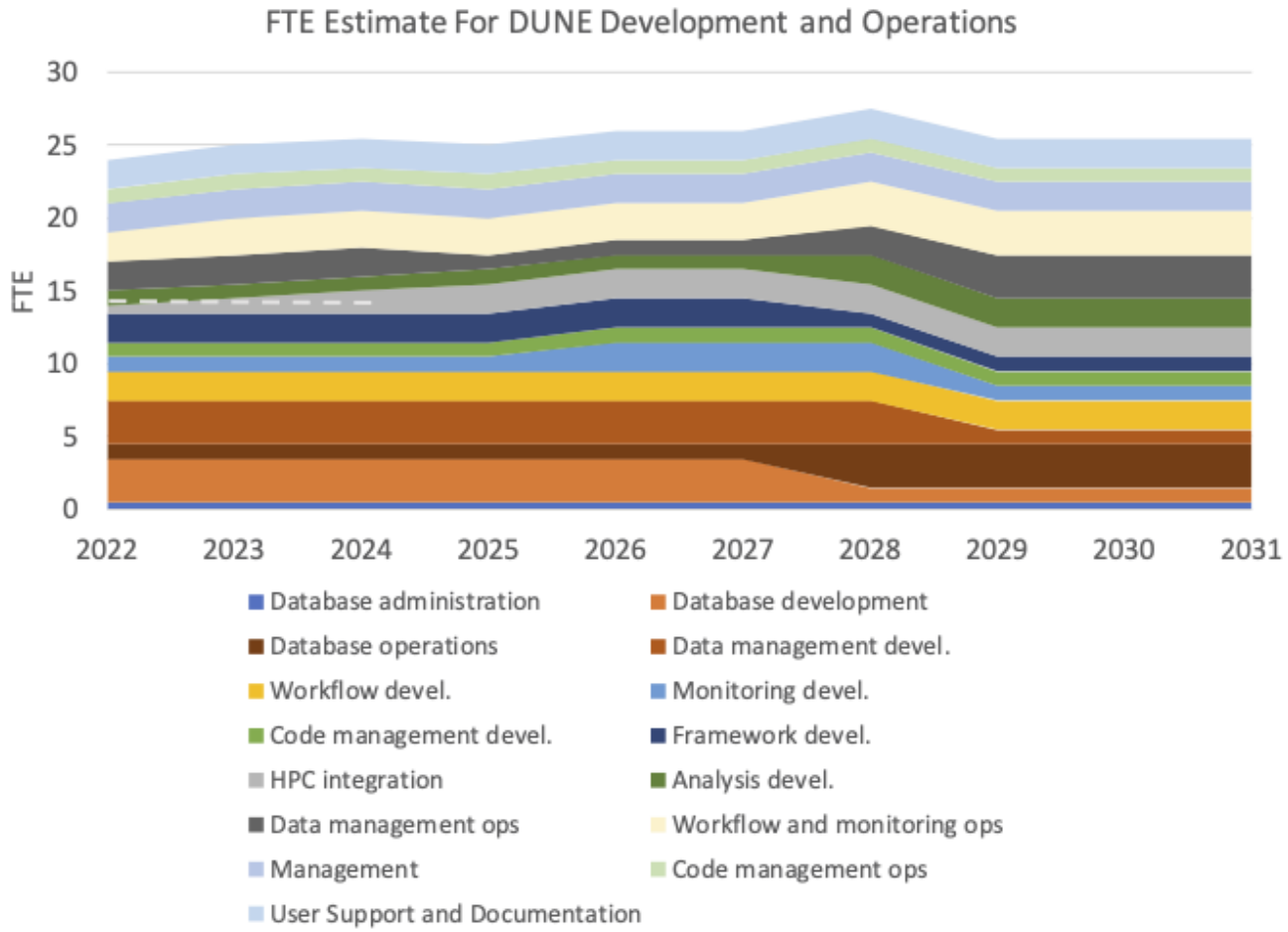


Figure 19.1: Estimated computing infrastructure personnel needs through 2030. The dark colors show development areas where experts are needed and the lighter colors show operations tasks where non-experts can contribute. The dashed line shows the estimated effort currently formally allocated to the project, including additional effort from 2022-2024 from the three-year DOE grant and enhanced UK project funding.

The roles listed in this chapter are limited to DUNE-specific projects and do not include the large number of personnel at multiple sites who support generic activities in areas such as storage and grid.

19.2.1 Technical Roles

Technical roles require substantial computing expertise. The numbers listed below are the estimated FTE, not FTE-years, as the needs will be ongoing. They appear as the darker areas in Figure 19.1, adding up to 16-18 FTE.

Database Design, Development and Management – 3.5 FTE

These roles include designing, developing, maintaining, and scaling databases for tasks within DUNE. Considerable effort is needed to interface with the large number of physics, calibration, data acquisition,

and hardware tasks. People are needed with expertise in databases, data acquisition, and project management. Here we include 0.5 FTE of database administration effort for direct database interventions and operations as it requires specialized skills and access to restricted resources. The majority of the development database roles are expected to evolve into corresponding database operations roles (see Section 19.2.2) as a way of retaining expertise and allowing future upgrades of the database infrastructure[1].

Distributed Data Management Development – 3.0 FTE

These roles include oversight of all software engineering and development activities for packages needed to operate distributed storage resources. The roles require a good understanding of the distributed computing infrastructure used by DUNE and collaborating sites, as well as the DUNE computing model, including:

- Storage management, integration and monitoring – 2.5 FTE

  These involve commissioning of new storage as Rucio storage endpoints (RSE's), data placement, construction and development of storage-usage monitoring and data-transfer monitoring, testing, commissioning of tape store services as they evolve, and participating in data transfer challenges.
- Core Rucio Development – 0.5 FTE

  Rucio development will continue in order to meet the emerging DUNE requirements. This ongoing work includes further development and integration of the lightweight client, proximity mapping for optimizing job placement, quality-of-service support to differentiate between disk, tape-backed disk, and tape, and further developments that will certainly emerge after the experience of ProtoDUNE.

Workflow Management Development 2.0 FTE

This involves ongoing design of the Workflow System needed by DUNE for distributed processing of data and simulation; and the implementation and testing of that design, both as systems developed by DUNE or adapted from elsewhere, and the creation of any necessary interfaces for particular classes of computational resources including high-performance computing (HPC) machines.

DUNE's computing model is designed to be less hierarchical than the classic LHC systems, which motivates the creation of systems that are application and network aware and take full advantage of the Rucio-based data management systems.

Monitoring Development – 1.0 FTE

This work includes oversight of all software engineering and development activities for packages needed to monitor distributed disk and compute resources. The role requires a good understanding of the distributed computing infrastructure used by DUNE as well as the DUNE computing model.

[1] It is noted that many experiments have reported difficulties in retaining database expertise, which can result in greatly increased risk over time as technologies evolve.

Code Management Development – 1.0 FTE

The code managers provide infrastructure to support applications for data processing, simulation, and analysis, and also to coordinate activities in the areas of development, release preparation, and deployment of software package releases needed by DUNE. They organize the overall setup of software packages needed for releases.

The application managers need to keep up with evolution in operating systems, build systems, and compilers, so this role has a significant development component.

Core simulation and reconstruction framework development – 2.0 FTE

This role requires substantial expertise in the design and deployment of sophisticated frameworks for HEP algorithmic workflows. It involves coordinating significant additional effort from common projects (Liquid Argon Software (LArSoft), art ...) and from collaboration algorithm development.

HPC integration – 2.0 FTE

Integrating DUNE's code to run on the diverse HPC systems available to us will require substantial administrative and technical development to negotiate resources and to adapt DUNE codes to run in very particular environments.

Analysis development – 1.0 FTE

This involves working with physics and detector groups on the development of specialized systems for efficient analysis of reconstructed data samples.

19.2.2 Operational Roles

In addition to the development activities listed above, people are needed to manage and supervise collaborators in operations tasks. These roles are more fluid, based on the status of the experiment and many do not require computing expertise; they can be filled by other collaborators after some training.

Distributed Data Management – 2.0 FTE

The distributed data managers are responsible for operational interactions with distributed computing disk and tape resources. The role includes but is not limited to helping to establish new storage areas, and to manage data replication, deletion, and movement.

Computing Coordination – 2.0 FTE

Coordinators manage the computing projects, and oversee the projection and distribution of resources needed to accomplish these projects' tasks. These roles serve largely as liaisons between the detector, physics, and computing resources of the experiment.

- Computing Consortium Leads

  This group includes the consortium leads as well as the liaisons to the DUNE detector and physics groups and to external entities such as Open Science Grid (OSG) and Worldwide LHC Computing

Grid (WLCG). The group's responsibilities include overall coordination of computing projects and negotiations regarding funding and resources with both agencies and institutions.

- Resource Board Chair

  This role is responsible for chairing biannual meetings of the Computing Resource Board, which includes representatives from the various national funding agencies that support DUNE, to discuss funding for and delivery of the computing resources required for successful processing and exploitation of DUNE data.

Workload Management operations – 2.5 FTE

- Distributed Workload Manager – 1.0 FTE

  The distributed workload managers are responsible for operational interactions with distributed computing resources. They are responsible for helping to establish grid and cloud sites. They are also responsible for the setup, launch, monitoring, and completion of processing campaigns (e.g., data processing, Monte Carlo (MC) simulation, and working group productions), executed on distributed computing resources for the experiment.
- Computing Shift Leaders – 1.0 FTE

  The shift leader is responsible for the experiment's distributed computing operations for week-long periods that run Monday through Sunday. Shift leaders chair regular operations meetings during their week and attend general DUNE operations meetings as appropriate.
- Distributed Computing Resource Contacts – 0.5 FTE

  Distributed computing resource contacts are the primary contacts for the DUNE distributed computing operations team and for the operators of large (Tier-1) sites and regional federations. They interact directly with the computing shift leaders at operations meetings.

Code Management Operations – 1.0 FTE

In addition to development associated with changes in systems, code librarians and application managers must continually prepare releases and deploy software packages needed by DUNE. Builds are currently done on a weekly basis.

User Support – 2.0 FTE

User support (software infrastructure, applications, and distributed computing) underpins all user activities of the DUNE computing project. User support personnel respond to questions from users on mailing lists, Slack-style chat systems, and/or ticketing systems, and are responsible for documenting solutions in knowledge bases, FAQs and wikis.

Chapter 20

Summary

This document has outlined the major use cases and challenges identified for Deep Underground Neutrino Experiment (DUNE) computing for time scales ranging from the upcoming year, e.g., ProtoDUNE-II, to long-term operations of the full DUNE far detector (FD) and near detector (ND). Substantial preliminary studies and design work have also been outlined.

20.1 Review of Challenges

In the introduction to this document (Section 1.5.2), the following challenges were introduced—some unique to DUNE, some more general, but all of them significant.

Large memory footprints - DUNE events, with multiple data objects consisting of thousands of channels and thousands of time samples, present formidable challenges for reconstruction on typical HEP processing systems. This report describes the use cases (Chapter 3), framework (Chapter 4) and compute requirements (Chapter 7) to address these use cases.

Storing and processing data on heterogeneous international resources - DUNE depends on the combined resources of the collaboration for large-scale storage and processing of data. Tools for using shared resources ranging from small-scale clusters to dedicated high-performance computing (HPC) systems are being designed but will need to be further developed and maintained. Chapters 6, 7 and 11 describe the data volumes, computing model, and data management plans.

Machine learning - Use of Machine Learning (ML) techniques can greatly improve simulation, reconstruction and analysis of data. However, integration of ML techniques into a software ecosystem the size and complexity of a large HEP experiment requires substantial effort beyond proof-of-concept and small-scale demonstration. Chapters 3 and 17 discuss some known applications and their management, but substantial effort will be needed to keep up with this rapidly evolving field.

Efficient and sustainable use of resources - The proposed global computing model allows DUNE to make flexible use of resources worldwide. Our workflow systems are being designed to ensure that CPU, network, and storage resources are well matched to achieve high efficiency. We are

working with the HEP community on improved documentation and training in best practices to ensure that our user community can do their work accurately and efficiently.

Keeping it all going - Many computing activities, both novel and mundane, need to continue over the full lifetime of the experiment. The chapters on databases(5), data management(11, 9), workflow(13), services (14), authentication (16), code management (17) and training and documentation (18) describe many of the systems that must be designed and then maintained over this time period.

In addition, while our computing challenges are not on the same scale as those of the large LHC experiments, substantial human effort, CPU resources, dedicated storage, and fast networks will need to be acquired and made easily available to the collaboration. And whereas DUNE will not be as resource intensive as the LHC experiments, it comes with unique computing challenges related to supernova neutrino bursts (SNBs), calibration, and other time-variable trigger records. Chapter 7 provides estimates of our needs while Sections 2.3, 7 and 19 describe our existing collaborative resources and future needs. The timetable shown in Figure 20.1 gives a rough estimate of the timeline for development, commissioning, and operations of different components of the DUNE Offline Computing and their relation to the experiment timeline.

20.2 Conclusion

In conclusion, we have identified the major challenges facing DUNE computing and have begun the process of designing and deploying solutions. We have preliminary estimates of resource needs and have broad connections to collaborating institutions and countries that have already provided substantial resources on a voluntary basis. This document lays out our understanding of the resources and activities that will be required to support timely data reconstruction and analysis as DUNE grows, and ultimately to allow the DUNE experiment to fully exploit its physics potential.

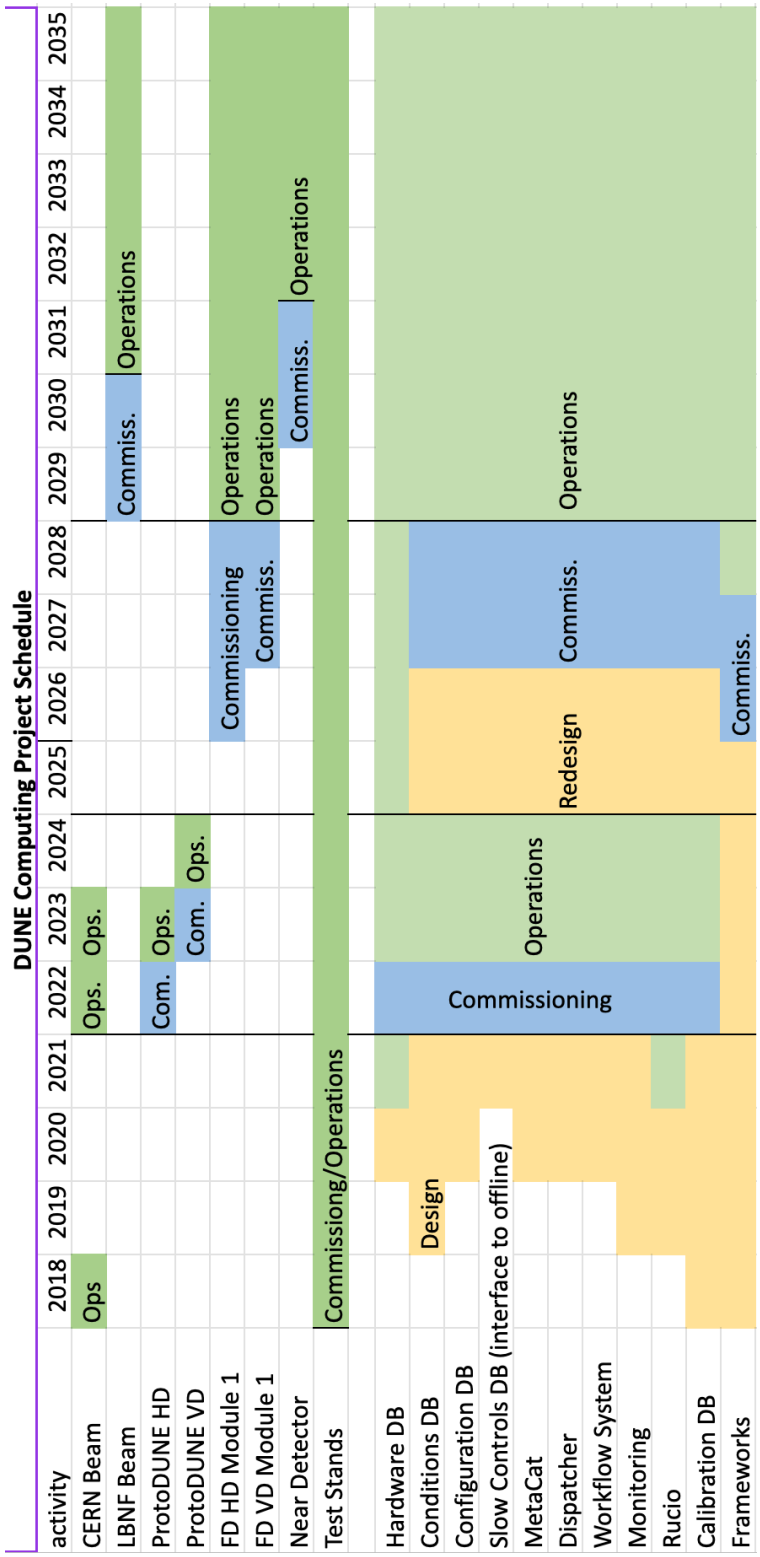


Figure 20.1: DUNE Computing Consortium timeline for development, commissioning, and operations of various services and tasks within DUNE Offline Computing.

Glossary

3D scintillator tracker (3DST) The core part of the 3D projection scintillator tracker spectrometer in the near detector conceptual design. 79

4850L The depth in feet (1480 m) of the access level for the DUNE underground area at SURF; called the "4850 level". 34

analog-to-digital converter (ADC) A sampling of a voltage resulting in a discrete integer count corresponding in some way to the input. 7, 8, 12, 50, 59, 60

anode plane assembly (APA) A unit of the horizontal drift technology (FD1-HD) detector module containing the elements sensitive to ionization in the liquid argon (LAr). Each anode face has three planes of wires (two induction, one collection) to provide a 3D view, and interfaces to the cold electronics and photon detection system. 6, 7, 10, 13, 15, 39, 40, 42, 49, 50, 60, 62, 63, 80–82, 106, 107, 205, 213

API Application Programming Interface. 101

ARAPUCA A photon detection system (PDS) design that consists of a light trap that captures wavelength-shifted photons inside boxes with highly reflective internal surfaces until they are eventually detected by silicon photomultiplier (SiPM) detectors or are lost. 106

ArgonCube The name of the core part of the Deep Underground Neutrino Experiment (DUNE) near detector (ND), a liquid argon time-projection chamber (LArTPC). 19, 211

art A software framework implementing an event-based execution paradigm. 36, 46, 78–82, 91, 173, 180, 183, 187, 196, 212

ATLAS One of two general-purpose detectors at the LHC. It investigates a wide range of physics, from the measurements of the Higgs boson properties to searches for extra dimensions and particles that could make up dark matter (DM). 214

Brookhaven National Laboratory (BNL) US national laboratory in Upton, NY. 29, 93, 101, 102, 121, 173

BSM beyond the Standard Model. 5, 18, 22

CAFAna Common Analysis File Analysis. 67, 69, 79, 80

CERN Advanced STORage manager (CASTOR) a hierarchical storage system with tape and disk developed at European Laboratory for Particle Physics (CERN). It is being replaced by CERN Tape Archive (CTA). 133, 140, 203

charged current (CC) Refers to an interaction between elementary particles where a charged weak force carrier (W^+ or W^-) is exchanged. 16, 22

Computing Contributions Board (CCB) a board made up of institutional representatives for larger countries and laboratories. It meets annually to negotiate collaboration contributions to computing infrastructure. 25, 28–30, 129, 193

CDR Depending on context, either "conceptual design report," a formal project document that describes the experiment at a conceptual level, or "conceptual design review," a formal review of the conceptual design of the experiment or of a component. 5, 110

European Laboratory for Particle Physics (CERN) The leading particle physics laboratory in Europe and home to the ProtoDUNEs and other prototypes and demonstrators, including the Module 0s. 6–8, 15, 23, 29, 30, 34, 35, 38, 78, 101, 106, 120, 121, 125, 132, 133, 139–141, 161, 164, 172–174, 178, 179, 202, 203, 205, 206, 210, 211, 213

conventional facilities (CF) Pertaining to construction and operation of buildings and conventional infrastructure, and includes cavern excavation. 206

CI continuous integration. 184, 185

CMS Compact Muon Solenoid experiment; one of two general-purpose detectors at the LHC.. 116

convolutional neural network (CNN) A deep learning technique most commonly applied to analyzing visual imagery. 72, 136

Columnar Object Framework For Effective Analysis (COFFEA) Columnar data analysis framework[101] developed at Fermi National Accelerator Laboratory (Fermilab). 69, 134

Common Muon and Proton Apparatus for Structure and Spectroscopy (COMPASS) a multipurpose experiment at CERN's Super Proton Synchrotron (SPS). 22

conditions DB The Conditions Database stores the conditions metadata needed for data processing and analysis. 94–96, 98, 99, 101–103

conditions metadata defined as the information necessary to understand the context of physics data, e.g., beam data or calibrations. i, 86, 93–96, 98, 99, 101–103, 202

COsmic Ray SImulations for KAscade (CORSIKA) a program for detailed simulation of extensive air showers initiated by high-energy cosmic ray particles. 40, 41, 48, 79

charge conjugation and parity (CP) Product of charge conjugation and parity transformations. 3, 203

cathode plane assembly (CPA) The component of the FD1-HD detector module that provides the drift HV cathode. 6

Central Processing Unit (CPU) A computing processor, when used as a unit of processing; generally refers to a single core. 120, 124, 130, 162, 163

charge-parity symmetry violation (CPV) Lack of symmetry in a system before and after charge and parity transformations are applied. For charge conjugation and parity (CP) symmetry to hold, a particle turns into its corresponding antiparticle under a charge transformation, and a parity transformation inverts its space coordinates, i.e. produces the mirror image. 5

Computing Resource Information System (CRIC) a framework providing a centralized (and flexible) way to describe which resources are being used by the experiment and how. 30, 129, 168, 174, 175

charge readout (CRO) The system for detecting ionization charge distributions in a detector module. 203

charge-readout plane (CRP) An anode technology using a stack of perforated PCBs with etched electrode strips to provide charge readout (CRO) in 3D; it has two induction layers and one collection layer; it is used in the vertical drift technology (FD2-VD) far detector (FD) and dual-phase (DP) designs. 12, 13, 15, 81, 82, 106, 214

CERN Tape Archive (CTA) a hierarchical storage system with tape and disk developed at CERN. It is replacing CERN Advanced STORage manager (CASTOR). 133, 140, 173, 202

CERN VM File System (CVMFS) A distributed file system designed for scalable, high-performance distribution of software to interactive and batch computers. 39, 133, 134, 140, 141, 164, 165, 173, 182–184, 189

convolutional visual network (CVN) An algorithm for identifying neutrino interactions based on their topology and without the need for detailed reconstruction algorithms. 52

data acquisition (DAQ) The data acquisition system accepts data from the detector front-end (FE) electronics, buffers the data, performs a trigger decision, builds events from the selected data and delivers the result to the offline secondary DAQ buffer. 5, 7, 17–19, 28, 32, 34, 35, 56, 96–98, 103, 105, 107–110, 151, 155, 158–160, 204, 205, 215, 216

Data Dispatcher communicates between running processing jobs and the data delivery systems. It provides file location information and basic bookkeeping on file access and transfers. 147, 156

data lake The not-yet-realized concept of a storage service with multiple levels of quality of service in which the end user can access data without knowing the data's source location. 129

data tier Differing data types produced in a processing sequence, for example, raw data, reconstructed, derived analysis sample, histograms. 110, 112, 135

DB database. 93, 95, 98–100, 102, 103

dCache A distributed, highly scalable (multi-PB) storage system, usable as both a standalone system and as a high-speed frontend to a tape storage system (such as Pseudo Network File System (PNFS) at Fermilab). 8, 139–141, 164, 165, 173, 182

dark matter (DM) The term given to the unknown matter or force that explains measurements of galaxy motion that are otherwise inconsistent with the amount of mass associated with the observed amount of photon production. 201

DOE U.S. Department of Energy. 29, 78, 92, 93, 101, 102, 193, 205, 210

data organization, management, and access (DOMA) data organization, management, and access efforts through the HEP Software Foundation (HSF). 129

dual-phase (DP) Distinguishes a LArTPC technology by the fact that it operates using argon in both gas and liquid phases; sometimes called double-phase. 6, 8, 12, 13, 203, 213

data quality monitoring (DQM) Analysis of the raw data to monitor the integrity of the data and the performance of the detectors and their electronics. This type of monitoring may be performed in real time, within the data acquisition (DAQ) system, or in later stages of processing, using disk files as input. 94, 103

Data Quality and Monitoring Database (DQMDB) Database storing the results of data-quality monitoring. 98

Deep Underground Neutrino Experiment (DUNE) A leading-edge, international experiment for neutrino science and proton decay studies; refers to the entire international experiment and collaboration. ix, 3, 5, 6, 16, 17, 22, 23, 25, 28–30, 32, 36, 37, 41, 43, 72, 75–83, 85, 88, 89, 91–96, 98–103, 105, 110, 112, 114, 124, 128, 131, 132, 134–140, 142, 152, 155, 160–163, 185, 188, 198, 199, 201, 204, 205, 208, 209, 211, 215, 216

DUNE General Purpose Virtual Machine (dunegpvm) Centrally managed virtual Linux systems at Fermilab with access to network attached and PNFS storage. Used for small-scale data analysis and algorithm development. 133, 189

dunesw The base DUNE software release. 187

electromagnetic calorimeter (ECAL) A detector component that measures energy deposition of traversing particles (in the DUNE near detector design). 21, 66, 68, 182, 207, 209, 211

engineering document management system (EDMS) A computerized document management system developed and supported at the CERN in which some DUNE project and collaboration documents, drawings and engineering models are managed. 28, 174

Experiment Hall North One (EHN1) Location at CERN of the NP02 and NP04 areas used for the ProtoDUNEs and for other test and prototyping activities for DUNE. 6, 212

Enstore A mass storage system developed by Fermilab that provides distributed access and management of data stored on tapes. 139, 140, 173

EOS (EOS) The XRootD-based distributed file system developed by CERN. 133, 140, 164

Energy Sciences Network (ESnet) The DOE's dedicated science network. 161

Experiment Test Framework (ETF) Worldwide LHC Computing Grid (WLCG) testing middleware that runs grid jobs that actively test distributed sites' services and capabilities, and reports back to monitoring services. 174–176

Frequently Asked Questions (FAQ) A software system for collecting and answering the most common questions about an activity. 188

far detector module The entire DUNE far detector design calls for segmentation into four modules, each with a total/fiducial mass of approximately 17 kt/10 kt. 5, 107, 116, 211, 216

far detector (FD) The 70 kt total (40 kt fiducial) mass LArTPC DUNE detector, composed of four 17.5 kt total (10 kt fiducial) mass modules, to be installed at the far site at Sanford Underground Research Facility (SURF) in Lead, SD, USA. 3, 5, 6, 15, 18, 19, 22, 32, 33, 35, 36, 38, 42, 44, 47–49, 63, 78, 80–82, 88, 90, 105–107, 110, 113, 114, 124, 132, 158–160, 187, 198, 203, 209, 215, 216

horizontal drift technology (FD1-HD) LArTPC design in which electrons drift horizontally to wire plane anodes (anode plane assemblies (APAs)) that along with the front-end electronics are immersed in LAr. 5, 6, 8, 15, 39, 60, 106, 107, 201, 203, 205, 213

vertical drift technology (FD2-VD) LArTPC design in which electrons drift vertically to PCB-based anodes at the top and bottom of the LAr volume, with a cathode in the middle. 5, 13, 15, 39, 41, 50, 60, 106, 107, 109, 203, 207, 212

front-end (FE) The front-end refers to a point that is "upstream" of the data flow for a particular subsystem. For example the FD1-HD front-end electronics is where the cold electronics meet the sense wires of the TPC and the front-end DAQ is where the DAQ meets the output of the electronics. 203

Fermi National Accelerator Laboratory (Fermilab) U.S. national laboratory in Batavia, IL. It is the laboratory that hosts Long-Baseline Neutrino Facility (LBNF) and DUNE, and serves as the experiment's near site. ix, 3–5, 7, 8, 12, 15, 18, 19, 22, 29, 30, 34, 78, 79, 91–93, 97, 101, 102,

120, 121, 125–127, 131–134, 141, 146, 156, 158–166, 172–174, 178–180, 182, 183, 186, 188, 189, 202, 204–206, 208–216

Frontier Experiments RegistRY (FERRY) a central database that keeps track of all scientific computing users at Fermilab, the experiments and groups of which they are members, and the various capabilities they are allowed. 173, 178, 179

Fast Fourier Transform (FFT) An algorithm that calculates the frequency components of a time-domain waveform in a computationally efficient manner. 51, 82

FHICL Fermilab Hierarchical Configuration Language; a standard configuration language for the storage, communication, and manipulation of scientific parameter sets. 39

Fabric for Intensity Frontier Experiments (FIFE) Fermilab computing infrastructure for Intensity Frontier (IF) Experiments. 120

FabrIc for Frontier Experiments MONitoring (FIFEMON) Comprehensive suite of job and storage monitoring information available for most Fermilab experiments, including DUNE. 173, 177

field programmable gate array (FPGA) An integrated circuit technology that allows the hardware to be reconfigured to execute different algorithms after its manufacture and deployment. 131

FS Depending on context, one of (1) the far site, SURF, where the DUNE far detector is located; (2) "Full Stream" relates to a data stream that has not undergone selection, compression or other form of reduction. 216

Far Site Conventional Facilities (FSCF) The conventional facilities (CF) at the DUNE far detector site, SURF, including all detector caverns and support infrastructure. 216

FTE full-time equivalent. A unit of labor for the project. One year of work from one person. 193–195

FTS3 File Transfer Service version 3, developed by CERN, and distinct from, the Fermilab File Transfer Service. 155, 157

g4lbnf LBNF neutrino beamline simulation program[30]. 38, 45

GÉANT GÉANT interconnects Europe's national research and education networking (NREN) organizations with the high bandwidth, high speed and highly resilient pan-European backbone.. 161

GArAna provides a facility for making an analysis ntuple from information stored in Gaseous Argon Software (GArSoft) data products. 183

Garfield A simulation program[68] developed at CERN for gaseous detectors. 48

Gaseous Argon Software (GArSoft) A software toolkit similar to Liquid Argon Software (LArSoft), but targeted at the gaseous argon time projection chamber and calorimeter of ND-GAr. ix, 39,

41, 46, 58, 66, 67, 79, 183, 206

gaseous argon time-projection chamber (GArTPC) A time projection chamber (TPC) filled with gaseous argon. 21, 182

geometry description markup language (GDML) An application-independent, geometry-description format based on XML. 39

Geant4 A software toolkit for the simulation of the passage of particles through matter using Monte Carlo (MC) methods. 39, 41–48, 57, 79, 112, 180, 181

Geant4Reweight Framework for evaluating and propagating hadronic interaction uncertainties in Geant4[49]. 42

Generates Events for Neutrino Interaction Experiments (GENIE) Software providing an object-oriented neutrino interaction simulation resulting in kinematics of the products of the interaction. 40, 41, 45, 46, 79

GiBUU Giessen Boltzmann-Uehling-Uhlenback Project; a unified theory and transport framework in the MeV and GeV energy regimes for elementary reactions on nuclei. 40

git a distributed version-control system, commonly used to manage software. 207

GitHub a commercial web service providing code version management, storage, and browsing via git. 144, 187, 189

GlideinWMS A system of submitting pilot jobs to grid computing sites, inside of which user jobs run, presenting a uniform setup across many different sites. ix, 162, 163, 173, 176

Graphical Processing Unit (GPU) Specialized computing hardware optimized for image processing. 57, 72, 75, 131

GPU As A Service (GPUAAS) a technique that allows many non-GPU-enabled compute nodes to share a GPU resource by sending it work over the network and waiting for results to be returned. 52

GRAIN In the System for on-Axis Neutrino Detection (SAND) detector, a small cryostat containing LAr installed upstream of the straw-tube tracker inside the electromagnetic calorimeter (ECAL). 21, 66

H2 CERN North Area hadron beamline used for ProtoDUNE-DP and FD2-VD prototypes and demonstrators. 212

H4 CERN North Area hadron beamline used for ProtoDUNE-SP and ProtoDUNE-SP-II. 212

HD horizontal drift TPC technology. 18, 81, 106–108, 215

Hierarchical Data Format (HDF5) Data format[127] widely used in Machine Learning (ML). 35, 136, 137

HEP high energy physics. 3, 23, 72, 75–82, 85, 91, 95, 96, 101, 102, 131, 135–137, 161, 162, 198, 208, 213, 214, 217

HEPCloud routes jobs to local or remote computing resources based on the policy for a particular experiment, workflow requirements, and cost and efficiency of accessing the various resources. It expands the resources available to include high-performance computing (HPC) centers and commercial cloud resources. 173

High Level Analysis at a Neutrino Detector (HighLAND) Analysis framework developed by the T2K collaboration. 67, 69

HL-LHC High-luminosity LHC. 32

high-performance computing (HPC) high-performance computing facilities; generally computing facilities emphasizing parallel computing with aggregate power of more than a teraflop. 23, 74, 75, 80, 81, 91, 93, 102, 131, 162, 165, 168, 195, 196, 198, 208

high-pressure gaseous argon TPC (HPgTPC) A TPC filled with gaseous argon; a possible component of the DUNE ND. 211

HEP Software Foundation (HSF) A foundation that facilitates cooperation and common efforts in high energy physics software and computing internationally. 23, 30, 80, 81, 89, 91, 92, 96, 101, 102, 187, 190, 204

High Throughput Computing (HTC) Computing facilities typically consisting of large numbers of commodity servers as opposed to a single large machine. Best suited for running large numbers of independent jobs in parallel, these facilities are what is usually meant by "grid computing". 124, 162, 166

high voltage (HV) Generally describes a voltage applied to drive the motion of free electrons through some media, e.g., LAr. 34, 59

HWDB hardware database. 99, 100

ICEBERG R&D cryostat and electronics (ICEBERG) Integrated Cryostat and Electronics Built for Experimental Research Goals: a double-walled cryostat built and installed at Fermilab for liquid argon detector R&D and for testing of DUNE detector components. 59

Intensity Frontier (IF) refers to HEP experiments, particularly in the U.S., that rely on high luminosity instead of high energy for discovery. Includes B-physics, neutrino, and muon experiments. 128, 206, 209, 214

IFbeam Database that stores beamline information indexed by timestamp. 35, 99

Intensity Frontier Data Handling (ifdh) The actual command invoked when using Intensity Frontier Data Handling Client (IFDHC), on the command line, e.g. ifdh cp source_file dest_file. 139

Intensity Frontier Data Handling Client (IFDHC) A multi-protocol tool for data transfer and file delivery in jobs. It is able to automatically select transfer protocols based on source and destination characteristics. 209

Indico Web-based meeting organization tool. 174

interval of validity (IOV) Interval over which something is valid. 94, 96, 98

IPv6 the most recent version of the Internet communications protocol that provides an identification and location system for computers on networks and routes traffic across the Internet. 130, 175, 176

inner readout chamber (IROC) inner (radial) readout chamber for gaseous argon TPC. 46

JobSub [122] the Fermilab IF job submission client that supports user submission of complex workflows to HT-Condor. ix, 162, 163, 173

JavaScript Object Notation (JSON) Open standard data interchange format that uses pair-value pairs and maps well onto python data formats such as tuples and lists. 96, 97, 114, 151, 152, 209

JWT JavaScript Object Notation (JSON) web token. 173, 179

KLOE KLOE is a e^+e^- collider detector spectrometer operated at DAFNE, the ϕ-meson factory at Frascati, Rome. In DUNE it will consist of a 26 cm Pb+scintillating fiber ECAL surrounding a cylindrical open detector region that is 4.00 m in diameter and 4.30 m long. The ECAL and detector region are embedded in a 0.6 T magnetic field created by a 4.86 m diameter superconducting coil and a 475 tonne iron yoke. 79

Local Area Network (LAN) Computing network confined to a relatively small geographic area. 67, 158, 160

liquid argon (LAr) Argon in its liquid phase; it is a cryogenic liquid with a boiling point of 87 K and density of 1.4 g/ml. 5, 6, 12, 13, 19, 22, 42, 43, 48, 105, 201, 207, 212, 216

Liquid Argon Software (LArSoft) A shared base of physics software across LArTPC experiments. 13, 36, 39–41, 43, 44, 46–48, 60, 62, 63, 69, 78, 79, 133, 136, 137, 140, 173, 180, 181, 183, 184, 187, 188, 190, 196, 206, 212

liquid argon time-projection chamber (LArTPC) A TPC filled with liquid argon; the basis for the DUNE FD modules. 5, 7, 12, 17–19, 21, 22, 36, 42, 56, 79, 82, 105, 182, 201, 204, 205, 209,

211, 215, 217

Layer-3 Networking protocol https://en.wikipedia.org/wiki/Network_layer. 161

Long-Baseline Neutrino Facility (LBNF) Long-Baseline Neutrino Facility; refers to the facilities that support the experiment including in-kind contributions under the line-item project. The portion of LBNF/DUNE-US responsible for developing the neutrino beam, the far site cryostats, and far and near site cryogenics systems, and the conventional facilities, including the excavations. 15, 38, 205

LBNF/DUNE-US Long-Baseline Neutrino Facility/Deep Underground Neutrino Experiment - United States; project to design and build the conventional and beamline facilities and the DOE contributions to the detectors. It is organized as a DOE/Fermilab project and incorporates contributions to the facilities from international partners. It also acts as host for the installation and integration of the DUNE detectors. 160, 210, 211, 216

Lawrence Berkeley National Laboratory (LBNL) US national laboratory in Berkeley, CA. 211

large electron multiplier (LEM) A micro-pattern detector suitable for use in ultra-pure argon vapor; LEMs consist of copper-clad PCB boards with sub-millimeter-size holes through which electrons undergo amplification. 12, 13

LHC Large Hadron Collider. 21, 29, 124, 128, 137, 161, 175, 199, 201, 202, 208

Linux Public Login User Service) (LXPLUS) The interactive logon service to Linux for all CERN users. 133

MAD-X framework that provides the de facto standard scripting language to describe particle accelerators, simulate beam dynamics, and optimize beam optics at CERN. 38

MARS The MARS code system is a set of MC programs for detailed simulation of coupled hadronic and electromagnetic cascades, with heavy ion, muon and neutrino production and interactions. 38

Monte Carlo (MC) Refers to a method of numerical integration that entails the statistical sampling of the integrand function. Forms the basis for some types of detector and physics simulations. 39, 66, 155, 164, 197, 207, 210

MetaCat (MetaCat) Metadata Catalog, a modern replacement for the file description portion of the sam metadata catalog. 147–156, 173, 174, 210, 211

MetaCat-dataset A fixed but mutable collection of files defined by queries to the MetaCat (MetaCat) data catalog. 152, 154

MINERvA Neutrino cross sections experiment at Fermilab, shut down in 2019. 45, 56

MINOS A long-baseline neutrino experiment, with a near detector at Fermilab and a far detector in the Soudan mine in Minnesota, designed to observe the phenomena of neutrino oscillations (ended data runs in 2012). 56

Machine Learning (ML) Machine Learning. 23, 72, 136, 198, 208

Module 0 The final pre-production instance of a detector; for the DUNE far detector modules, these will use the 800 t cryostats in NP02 and NP04. 13, 202

MONIT a suite of tools using open source technology used in the monitoring of the CERN IT data center and WLCG infrastructure.. 176

memorandum of understanding (MoU) A project management methodology that LBNF/DUNE-US uses to document agreement,s, e.g., between Fermilab and the Project, for how Fermilab will support the Project. More generally, a document summarizing an agreement between two or more parties. 30

multi-purpose detector (MPD) A component of the near detector conceptual design; it is a magnetized system consisting of a high-pressure gaseous argon TPC (HPgTPC) and a surrounding ECAL. 216

MetaCat Query Language (MQL) A query language which supports queries of the MetaCat data catalog, including parentage and logical functions such as union, join and subtraction. 154

network attached storage (NAS) Disk storage that is available on computers but shared between them. Relies on network file system (NFS) mounts rather than authenticated file transfer protocols. Usually found on interactive servers to provide space for home directories, app and data storage. 139, 142

near detector (ND) Refers to the collection of DUNE detector components installed close to the neutrino source at Fermilab; also a subproject of LBNF/DUNE-US that includes installation, infrastructure, and the cryogenics systems for this detector. 3, 5, 18–20, 22–24, 29, 32, 35, 36, 41, 53, 88, 90, 105, 106, 110, 111, 114, 116, 124, 158, 180, 182, 184, 198, 201, 208

ND-GAr component of the near detector with a core gaseous argon TPC surrounded by an ECAL and a magnet. 19–21, 42, 46, 56, 66, 67, 78, 79, 110, 111, 182, 183, 206, 211

ND-GAr-Lite a temporary muon spectrometer consisting of the magnet and steel flux return of ND-GAr, but with a simplified tracking chamber made with scintillating bars. 79

ND-LAr LArTPC component of the near detector based on ArgonCube technology. 19–21, 39, 42, 53, 55, 56, 79, 111, 182

NERSC National Energy Research Computing Facility at Lawrence Berkeley National Laboratory (LBNL). 74, 124, 131, 162, 165, 168

neutrino interaction generator (NEUT) A neutrino interaction simulation program library for the studies of atmospheric and accelerator neutrinos. 40

network file system (NFS) Industry-standard mechanism for mounting disks over a network. Provides regular UNIX file and directory access. 139, 211

NP02 The CERN North Area in Experiment Hall North One (EHN1) intersected by the H2 hadron beamline, the location of the 800 t cryostat used for ProtoDUNE-DP and for FD2-VD tests and prototypes; also used to refer to the 800 t cryostat in this area. 13, 155, 178, 205, 211, 213, 214

NP04 The CERN North Area in EHN1 intersected by the H4 hadron beamline, the location of ProtoDUNE-SP and ProtoDUNE-SP-II; also used to refer to the 800 t cryostat in this area. 155, 178, 205, 211, 213

National Research and Education Network (NREN) National level research computing network infrastructure. 130, 161

Nuisance An open source C++ framework for studying neutrino interaction cross sections[102]. 69

NuMI a set of facilities at Fermilab, collectively called "Neutrinos at the Main Injector." The NuMI neutrino beamline target system converts an intense proton beam into a focused neutrino beam. 35

NuTools shared code for LArSoft + NOvA + other neutrino experiments that use art. Includes beam and event generators and re-weighting packages. 180

NuWro neutrino interaction generator. 40

OpenStack An open source cloud software used to deploy instances of containers. 129

outer readout chamber (OROC) outer (radial) readout chamber for gaseous argon TPC. 46

Open Science Grid (OSG) Open Science Grid. 8, 23, 30, 120, 124, 128, 131, 162, 163, 166, 175, 178, 179, 196

Pandora The Pandora multi-algorithm approach to pattern recognition. 13, 51, 190

PCB printed circuit board. 13, 14, 203

photon detector (PD) The detector elements involved in measurement of the number and arrival times of optical photons produced in a detector module. 17–19, 22, 36, 49–51, 76, 105, 106, 116

photon detection system (PDS) The detector subsystem sensitive to light produced in the LAr. 201

performance Service-Oriented Network monitoring ARchitecture (perfSONAR) a network mea-

surement toolkit designed to provide federated coverage of paths and to help establish end-to-end usage expectations. 175–177

particle ID (PID) Particle identification. 51

Projection Matching Algorithm (PMA) A reconstruction algorithm that combines 2D reconstructed objects to form a 3D representation. 51

Pseudo Network File System (PNFS) A file system often used in large storage systems. Typically interaction is very similar to a regular NFS volume, but there can be some subtle and important differences. 133, 204

Production Operations Management System (POMS) A workflow management system available for all DUNE users to submit and monitor grid jobs as well as view job log files. ix, 8, 124, 162, 163, 173

POSIX Portable Operating System Interface - Unix standard for operating system interfaces. 138, 139

Postgres also known as PostgreSQL, Postgres is a free and open-source relational database management system used extensively for databases in HEP. 93, 102, 144

Package to Predict the FluX (PPFX) Fermilab-supported package that implements hadron production corrections to geant4 simulations and propagates uncertainties for the NuMI and LBNF beam lines. 45, 48

ProtoDUNE Either of the two initial DUNE prototype detectors constructed at CERN. One prototype implemented single-phase (SP) technology and the other DP. ix, 15, 18, 28, 30, 32–36, 39, 42, 44, 47–49, 51, 59, 69, 72, 78, 80, 81, 88, 91, 94–103, 106, 107, 114, 120, 128, 134, 136, 137, 140, 148, 151, 162, 163, 170, 178, 187, 195, 202, 205, 213

ProtoDUNE-DP The DP ProtoDUNE detector constructed at CERN in NP02. 6, 8, 12, 13, 15, 41, 42, 106, 155, 165, 207, 212

ProtoDUNE-HD ProtoDUNE with horizontal drift technology. This refers to the SP APA-based prototype to run in NP04 (in the ProtoDUNE-II phase). 15, 34, 35, 41, 134, 155, 156, 213

ProtoDUNE-II ProtoDUNE test runs at CERN in 2022-2023; includes ProtoDUNE-HD and ProtoDUNE-VD. 193, 198, 213, 214

ProtoDUNE-II The second run of a ProtoDUNE detector. 94, 100, 101

ProtoDUNE-SP The FD1-HD ProtoDUNE detector constructed at CERN in NP04. i, viii, 6, 7, 9, 10, 12, 13, 15, 35, 41, 42, 48–53, 60, 62–65, 67, 106–109, 113, 124, 142, 152, 155, 207, 212

ProtoDUNE-SP-II A second test run in the single-phase ProtoDUNE test stand at CERN, acting as a validation of the final single-phase detector design. 146, 150, 155, 207, 212

ProtoDUNE-VD ProtoDUNE with vertical drift technology. This refers to the charge-readout plane (CRP)-based prototype to run in NP02 (in the ProtoDUNE-II phase). 34, 35, 213

quality control (QC) The process (e.g., inspection, testing, measurements) of ensuring that each manufactured element meets its quality requirements prior to assembly or installation. 99, 100

Redmine an open source code repository and issue tracking tool that was historically used by many of the Fermilab general computing projects and IF experiments. 187

RNtuple The next generation of the ROOT I/O system[106, 107]. 79, 137

ROOT A modular scientific software toolkit. It provides all the functionalities needed to deal with big data processing, statistical analysis, visualization, and storage. It is mainly written in C++ but integrated with other languages such as Python and R. 35, 39, 45, 51, 66, 79, 135–137, 145, 181, 182, 214

Rucio Storage Element (RSE) A storage element that is known to the DUNE Rucio instance. 67, 139, 162, 164

Rucio Data management system originally developed by ATLAS but now open-source and shared across HEP. 30, 122, 132, 134, 139, 140, 142, 146–148, 150, 151, 155–157, 164–166, 173, 174, 176, 195, 214

S3 The Amazon cloud-based commercial storage service. 129

sequential access via metadata (SAM) A data-handling system to store and retrieve files and associated metadata, including a complete record of the processing that has used the files. 124, 128, 139, 140, 142, 146–152, 154–156, 162, 164, 165, 173, 174, 214

SAM-consumer A client process that requests information about file locations from a SAM-project, process the file and reports success or failure. 149, 150

SAM-dataset A dynamic collection of files defined by queries to the sequential access via metadata (SAM) data catalog. 146, 149, 152, 214

SAM-project A server process running centrally that maintains a predefined list of files and delivers information about their locations when asked by distributed processes. The project tracks success and failure of file processing. 149, 150, 214

SAM-snapshot A fixed collection of files corresponding to a SAM SAM-dataset at a particular point in time. 149

System for on-Axis Neutrino Detection (SAND) The beam monitor component of the near detector that remains on-axis at all times and serves as a dedicated neutrino spectrum monitor. 19–21, 66, 68, 79, 111, 182, 207

SBND The Short-Baseline Near Detector experiment at Fermilab. 22, 44

SCADA supervisory control and data acquisition. 34, 98, 99, 102

scientific computing division (SCD) Fermilab's Scientific Computing Division. 78, 79, 93, 99

secondary DAQ buffer A secondary DAQ buffer holds a small subset of the full rate as selected by a trigger command. This buffer also marks the interface with the DUNE Offline. 203

ServiceNow a commercial enterprise workflow system used by Fermilab for formal issue tracking and IT workflows such as user account preparation. 174, 187, 188

SingleCube A cubical time projection chamber 30 cm on a side with a single ND-LAr pixel readout tile, used for prototyping and tests. 79

silicon photomultiplier (SiPM) A solid-state avalanche photodiode sensitive to single photoelectron signals. 21, 47, 56, 201

Slack a commercial business communication platform. 188

supernova neutrino burst (SNB) A prompt increase in the flux of low-energy neutrinos emitted in the first few seconds of a core-collapse supernova. It can also refer to a trigger command type that may be due to this phenomenon, or detector conditions that mimic its interaction signature. 3, 5, 15–18, 22, 34, 49, 105, 106, 110, 158, 159, 199

Services for Optimized Network Inference on Coprocessors (SONIC) Framework for implementing machine learning algorithms on co-processors developed by Fermilab. 72

single-phase (SP) Distinguishes a LArTPC technology by the fact that it operates using argon in its liquid phase only; a legacy DUNE term now replaced by HD and VD. 6, 106, 213

SP module single-phase DUNE FD module. 6

Spack Software package manager for UNIX and mac OS systems. 173, 182

Simulation Program with Integrated Circuit Emphasis (SPICE) a general-purpose, open-source analog electronic circuit simulator. It is a program used in integrated circuit and board-level design to check the integrity of circuit designs and to predict circuit behavior. 48

single sign-on (SSO) Used at Fermilab to indicate that a group of services, such as DocDB or the DUNE Wiki share common sign-in credentials and active sessions. Fermilab services that say "Sign in with SSO username and password" mean to use your Fermilab Services or federated username and password. 186

StashCache A distributed caching federation that enables opportunistic users to utilize nearby opportunistic storage[32]. 38, 140, 142, 165, 173, 182

Sanford Underground Research Facility (SURF) The laboratory in Lead, SD, USA where the DUNE FD will be installed and operated; also where the LBNF/DUNE-US Far Site Conventional Facilities (FSCF) and the FS cryostat and cryogenics systems will be constructed. 3, 15, 18, 28, 34, 35, 82, 158–161, 205, 206

T2K T2K (Tokai to Kamioka) is a long-baseline neutrino experiment in Japan studying neutrino oscillations. 56, 208

TDR Depending on context, either "technical design report," a formal project document that describes the experiment at a technical level, or "technical design review," a formal review of the technical design of the experiment or of a component. 5, 8, 16

The Muon Spectrometer (TMS) A muon spectrometer for the Near Detector that will be installed for the initial running period of DUNE, before the multi-purpose detector (MPD) detector component is ready. 39, 55, 56, 110, 111, 182

time projection chamber (TPC) Depending on context: (1) A type of particle detector that uses an E field together with a sensitive volume of gas or liquid, e.g., LAr, to perform a 3D reconstruction of a particle trajectory or interaction. The activity is recorded by digitizing the waveforms of current induced on the anode as the distribution of ionization charge passes by or is collected on the electrode. (2) TPC is also used in LBNF/DUNE-US for "total project cost". 5, 7, 17–22, 35, 49–51, 56, 106, 207–209, 211, 212, 216

trigger candidate Summary information derived from the full data stream and representing a contribution toward forming a trigger decision. 216

trigger command Information derived from one or more trigger candidates that directs elements of a far detector module to read out a portion of the data stream. 215, 216

trigger decision The process by which trigger candidates are converted into trigger commands. 203, 216

trigger record A data record produced by the DUNE DAQ system. A Trigger Record can contain multiple interaction "events" or none. 7, 9, 10, 12, 32, 49

Unstructured Conditions Database (uconDB) Unstructured conditions database developed for Fermilab fixed target experiments[112]. 96, 97

UNIX product support (UPS) A software tool that sets up a consistent environment of versions of pre-installed products and their dependencies on UNIX-like platforms. 140, 180–183

VD vertical drift TPC technology. 13, 18, 106, 109, 215

virtual machine (VM) Emulator of a physical computer that allows multiple users to configure different operating systems while sharing physical hardware. 138

virtual organization (VO) A database containing a list of member names, certificate distinguishing information, and a list of permissions members have to access computing grid and data resources. 133, 140, 156, 172, 175, 176

VOMS Virtual Organization Membership Service (in grid computing). 133, 173, 179

Virtual Routing and Forwarding (VRF) Networking overlays that provide a logical routing infrastructure that allows flexible traffic engineering. 161

Wire-Cell Toolkit (WCT) A software toolkit with data flow processing components for LArTPC noise and signal simulation, noise filtering, signal processing, and tomographic 3D ionization activity imaging. 42, 50, 51

WFMS Workflow Management System. 77, 85

Worldwide LHC Computing Grid (WLCG) Worldwide LHC Computing Grid. 8, 23, 29, 30, 120, 124, 128–130, 140, 162, 166, 175–179, 196, 205, 211

xrootd a high-performance data system widely used in HEP to store and to distribute data to jobs. It allows streaming of data. 67, 124, 129, 133, 134, 137, 139, 140, 164

References

Some references to internal DUNE technical documents may be password protected. Please contact dune-computing-cdr@fnal.gov to request access.

[1] T. Kajita, "Discovery of neutrino oscillations," *Reports on Progress in Physics* **69** no. 6, (May, 2006) 1607–1635. https://doi.org/10.1088%2F0034-4885%2F69%2F6%2Fr01.

[2] **ArgoNeuT** Collaboration, R. Acciarri *et al.*, "First Observation of Low Energy Electron Neutrinos in a Liquid Argon Time Projection Chamber," *Phys. Rev.* **D95** no. 7, (2017) 072005, arXiv:1610.04102 [hep-ex].

[3] **DUNE** Collaboration, "TDR Volume 1: Introduction to DUNE," tech. rep., 2019. https://arxiv.org/abs/2002.02967.

[4] **DUNE** Collaboration, "TDR Volume 2: DUNE Physics," tech. rep., 2018. https://arxiv.org/abs/2002.03005.

[5] **DUNE** Collaboration, A. Abed Abud *et al.*, "Deep Underground Neutrino Experiment (DUNE) Near Detector Conceptual Design Report," *Instruments* **5** no. 4, (2021) 31, arXiv:2103.13910 [physics.ins-det].

[6] **DUNE** Collaboration, B. Abi *et al.*, "Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume II DUNE Physics," arXiv:2002.03005 [hep-ex].

[7] **DUNE** Collaboration, B. Abi *et al.*, "Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume I Introduction to DUNE," *JINST* **15** no. 08, (2020) T08008, arXiv:2002.02967 [physics.ins-det].

[8] **DUNE** Collaboration, B. Abi *et al.*, "Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume III DUNE Far Detector Technical Coordination," arXiv:2002.03008 [physics.ins-det].

[9] **DUNE** Collaboration, B. Abi *et al.*, "Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume IV Far Detector Single-phase Technology," arXiv:2002.03010 [physics.ins-det].

[10] **DUNE** Collaboration, "Deep Underground Neutrino Experiment Far Detector Conceptual Design Report, Single-Phase Vertical Drift Technology," dune doc, 2022. https://edms.cern.ch/file/2709820/1/CDR__Vertical_Single_Phase_FD_Technology_Final.pdf.

[11] **ArgoNeuT** Collaboration, R. Acciarri *et al.*, "Demonstration of MeV-Scale Physics in Liquid Argon Time Projection Chambers Using ArgoNeuT," *Phys. Rev.* **D99** no. 1, (2019) 012002, arXiv:1810.06502 [hep-ex].

[12] R. Acciarri *et al.*, "Design and construction of the microboone detector," *Journal of Instrumentation* **12** no. 02, (2017) P02017. http://stacks.iop.org/1748-0221/12/i=02/a=P02017.

[13] **ICARUS** Collaboration, F. Varanini, "ICARUS detector: present and future," *EPJ Web Conf.* **164** (2017) 07017.

[14] **DUNE** Collaboration, A. A. Abud *et al.*, "Design, construction and operation of the ProtoDUNE-SP Liquid Argon TPC," *JINST* **17** no. 01, (2022) P01005, arXiv:2108.01902 [physics.ins-det].

[15] **MicroBooNE** Collaboration, R. Acciarri *et al.*, "Noise Characterization and Filtering in the MicroBooNE Liquid Argon TPC," *JINST* **12** no. 08, (2017) P08003, arXiv:1705.07341 [physics.ins-det].

[16] **MicroBooNE** Collaboration, R. Acciarri *et al.*, "Design and Construction of the MicroBooNE Detector," *JINST* **12** no. 02, (2017) P02017, arXiv:1612.05824 [physics.ins-det].

[17] **DUNE** Collaboration, B. Abi *et al.*, "First results on ProtoDUNE-SP liquid argon time projection chamber performance from a beam test at the CERN Neutrino Platform," *JINST* **15** no. 12, (2020) P12004, arXiv:2007.06722 [physics.ins-det].

[18] "Production operations management system." https://cdcvs.fnal.gov/redmine/projects/prod_mgmt_db.

[19] G. Behrmann, D. Ozerov, and T. Zangerl, "Xrootd in dCache: Design and experiences," *J. Phys. Conf. Ser.* **331** (2011) 052021.

[20] **MicroBooNE** Collaboration, R. Acciarri *et al.*, "The Pandora multi-algorithm approach to automated pattern recognition of cosmic-ray muon and neutrino events in the MicroBooNE detector," *Eur. Phys. J.* **C78** no. 1, (2018) 82, arXiv:1708.03135 [hep-ex].

[21] **DUNE** Collaboration, C. Cuesta, "Core-Collapse Supernova Burst Neutrinos in DUNE," *PoS* **ICHEP2020** (2021) 590, arXiv:2011.06969 [physics.ins-det].

[22] **DUNE** Collaboration, B. Abi *et al.*, "Supernova neutrino burst detection with the Deep Underground Neutrino Experiment," *Eur. Phys. J. C* **81** no. 5, (2021) 423, arXiv:2008.06647 [hep-ex].

[23] D. Caratelli *et al.*, "Low-Energy Physics in Neutrino LArTPCs," in *2022 Snowmass Summer Study*. 3, 2022. arXiv:2203.00740 [physics.ins-det].

[24] **MINOS** Collaboration, D. Michael *et al.*, "The magnetized steel and scintillator calorimeters of the MINOS experiment," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **596** no. 2, (2008) 190 – 228. http://www.sciencedirect.com/science/article/pii/S0168900208011613.

[25] M. Adinolfi *et al.*, "The KLOE electromagnetic calorimeter," *Nucl. Instrum. Meth. A* **482** (2002) 364–386.

[26] H. Schellman *et al.*, "DUNE computing resource needs for 2022," 2022. https://docs.dunescience.org/cgi-bin/private/ShowDocument?docid=23419.

[27] **HEP Software Foundation** Collaboration, J. Albrecht *et al.*, "A Roadmap for HEP Software and Computing R&D for the 2020s," *Comput. Softw. Big Sci.* **3** no. 1, (2019) 7, arXiv:1712.06982 [physics.comp-ph].

[28] G. Karagiorgi, D. Newbold, A. Norman, and H. Schellman, "DUNE FD Interface Document: Software and Computing to Joint DAQ," DUNE doc 7123, Columbia, Bristol, Fermilab, and Oregon State, 2018. http://docs.dunescience.org/cgi-bin/ShowDocument?docid=7123&asof=2019-11-1.

[29] N.Mokhov, "The Mars Code System User's Guide," tech. rep., Fermilab, 2009. Fermilab-FN-628.

[30] "The G4LBNE Software." https://cdcvs.fnal.gov/redmine/projects/lbne-beamsim/wiki. Available at https://cdcvs.fnal.gov/redmine/projects/lbne-beamsim/wiki.

[31] N. Charitonidis and I. Efthymiopoulos, "Low energy tertiary beam line design for the cern neutrino platform project," *Phys. Rev. Accel. Beams* **20** (Nov, 2017) 111001. https://link.aps.org/doi/10.1103/PhysRevAccelBeams.20.111001.

[32] D. Weitzel, M. Zvada, I. Vukotic, R. Gardner, B. Bockelman, M. Rynge, E. F. Hernandez, B. Lin, and M. Selmeci, "Stashcache: A distributed caching federation for the open science grid," in *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning)*, PEARC '19. Association for Computing Machinery, New York, NY, USA, 2019. https://doi.org/10.1145/3332186.3332212.

[33] S. Agostinelli *et al.*, "Geant4 - a simulation toolkit," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **506** no. 3, (2003) 250 – 303. http://www.sciencedirect.com/science/article/pii/S0168900203013688.

[34] R. Chytracek, J. McCormick, W. Pokorski, and G. Santin, "Geometry Description Markup Language for Physics Simulation and Analysis Applications," *IEEE Trans. Nucl. Sci.* **53** (2006)

2892–2896.

[35] B. Viren. https://github.com/brettviren/gegede.

[36] C. Andreopoulos *et al.*, "The GENIE Neutrino Monte Carlo Generator," *Nucl. Instrum. Meth.* **A614** (2010) 87–104, arXiv:0905.2517 [hep-ph].

[37] T. Golan, J. T. Sobczyk, and J. Zmuda, "NuWro: the Wroclaw Monte Carlo Generator of Neutrino Interactions," *Nucl. Phys. Proc. Suppl.* **229-232** (2012) 499.

[38] K. Gallmeister, U. Mosel, and J. Weil, "Neutrino-Induced Reactions on Nuclei," *Phys. Rev. C* **94** no. 3, (2016) 035502, arXiv:1605.09391 [nucl-th].

[39] Y. Hayato, "A neutrino interaction simulation program library NEUT," *Acta Phys. Polon. B* **40** (2009) 2477–2489.

[40] C. Andreopoulos *et al.*, "The GENIE Neutrino Monte Carlo Generator," *Nucl. Instrum. Methods* **A** no. 614, (2010) 87–104.

[41] J. Wentz, I. M. Brancus, A. Bercuci, D. Heck, J. Oehlschlager, H. Rebel, and B. Vulpescu, "Simulation of atmospheric muon and neutrino fluxes with CORSIKA," *Phys. Rev. D* **67** (2003) 073020, arXiv:hep-ph/0301199.

[42] **CORSIKA 8** Collaboration, H. P. Dembinski, L. Nellen, M. Reininghaus, and R. Ulrich, "Technical Foundations of CORSIKA 8: New Concepts for Scientific Computing," *PoS* **ICRC2019** (2020) 236.

[43] V. A. Kudryavtsev, "Muon simulation codes MUSIC and MUSUN for underground physics," *Comput. Phys. Commun.* **180** (2009) 339–346, arXiv:0810.4635 [physics.comp-ph].

[44] J. K. V. Kudryavtsev, M. Richardson and T. K. Warburton, "Muon simulations for lbne using music and musun," tech. rep., LBNE, 2014. LBNE-doc-9673.

[45] O. Ponkratenko, V. Tretyak, and Y. Zdesenko, "The Event generator DECAY4 for simulation of double beta processes and decay of radioactive nuclei," *Phys. Atom. Nucl.* **63** (2000) 1282–1287, arXiv:nucl-ex/0104018.

[46] **GEANT4** Collaboration, S. Agostinelli *et al.*, "GEANT4: A Simulation toolkit," *Nucl. Instrum. Meth. A* **506** (2003) 250–303.

[47] J. Allison *et al.*, "Recent developments in Geant4," *Nucl. Instrum. Meth.* **A835** (2016) 186–225.

[48] Böhlen, T. T. and Cerutti, F. and Chin, M. P. W. and Fassò, A. and Ferrari, A. and Ortega, P. G. and Mairani, A. and Sala, P. R. and Smirnov, G. and Vlachoudis, V., "The FLUKA Code: Developments and Challenges for High Energy and Medical Applications," *Nucl. Data Sheets* **120** (2014) 211–214.

[49] J. Calcutt, C. Thorpe, K. Mahn, and L. Fields, "Geant4Reweight: a framework for evaluating and propagating hadronic interaction uncertainties in Geant4," *JINST* **16** no. 08, (2021) P08042, arXiv:2105.01744 [physics.data-an].

[50] X. Qian, B. Viren, and C. Zhang https://www.phy.bnl.gov/wire-cell/.

[51] "Wire-Cell Toolkit." https://github.com/WireCell.

[52] **DUNE** Collaboration, B. Abi *et al.*, "First results on ProtoDUNE-SP liquid argon time projection chamber performance from a beam test at the CERN Neutrino Platform," *JINST* **15** no. 12, (2020) P12004, arXiv:2007.06722 [physics.ins-det].

[53] **DUNE** Collaboration, B. Abi *et al.*, "Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume II: DUNE Physics," arXiv:2002.03005 [hep-ex].

[54] **MINERvA** Collaboration, L. Aliaga *et al.*, "Neutrino Flux Predictions for the NuMI Beam," *Phys. Rev.* **D94** no. 9, (2016) 092005, arXiv:1607.00704 [hep-ex]. [Addendum: Phys. Rev.D95,no.3,039903(2017)].

[55] L. Aliaga Soplin, *Neutrino Flux Prediction for the NuMI Beamline*. PhD thesis, William-Mary Coll., 2016. "http://lss.fnal.gov/archive/thesis/2000/fermilab-thesis-2016-03.pdf.

[56] C. McGrew. https://github.com/ClarkMcGrew/edep-sim.

[57] S. Soleti *et al.*, "Dune/larnd-sim," Mar., 2021. https://doi.org/10.5281/zenodo.4582721.

[58] D. Dwyer *et al.*, "LArPix: Demonstration of low-power 3D pixelated charge readout for liquid argon time projection chambers," *JINST* **13** no. 10, (2018) P10007, arXiv:1808.02969 [physics.ins-det].

[59] D. collaboration, "Physics analysis of data from a novel pixel-readout liquid argon Time Projection Chamber," *in preparation* .

[60] C. Hagmann, D. Lange, and D. Wright, "Cosmic-ray shower generator (cry) for monte carlo transport codes," vol. 2, pp. 1143 – 1146. 01, 2007.

[61] C. Hagmann, D. Lange, J. Verbeke, and D. Wright, "Cosmic-ray Shower Library (CRY)," tech. rep., Lawrence Livermore National Laboratory, 2012. UCRL-TM-229453.

[62] **ALICE** Collaboration, G. Dellacasa *et al.*, "ALICE: Technical design report of the time projection chamber," Tech. Rep. CERN-OPEN-2000-183, CERN-LHCC-2000-001, 2000.

[63] A. Hitachi, T. Takahashi, N. Funayama, K. Masuda, J. Kikuchi, and T. Doke, "Effect of ionization density on the time dependence of luminescence from liquid argon and xenon," *Phys. Rev. B* **27** (May, 1983) 5279–5285. https://link.aps.org/doi/10.1103/PhysRevB.27.5279.

[64] G. M. Seidel, R. E. Lanou, and W. Yao, "Rayleigh scattering in rare gas liquids," *Nucl. Instrum. Meth. A* **489** (2002) 189–194, arXiv:hep-ex/0111054.

[65] N. Ishida, M. Chen, T. Doke, K. Hasuike, A. Hitachi, M. Gaudreau, M. Kase, Y. Kawada, J. Kikuchi, T. Komiyama, K. Kuwahara, K. Masuda, H. Okada, Y. Qu, M. Suzuki, and T. Takahashi, "Attenuation length measurements of scintillation light in liquid rare gases and their mixtures using an improved reflection suppresser," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **384** no. 2, (1997) 380–386. https://www.sciencedirect.com/science/article/pii/S0168900296007401.

[66] M. Sneep and W. Ubachs, "Direct measurement of the Rayleigh scattering cross section in various gases," *J. Quant. Spectrosc. Radiat. Transf.* **92** no. 3, (May, 2005) 293–310.

[67] B. J. P. Jones, C. S. Chiu, J. M. Conrad, C. M. Ignarra, T. Katori, and M. Toups, "A Measurement of the Absorption of Liquid Argon Scintillation Light by Dissolved Nitrogen at the Part-Per-Million Level," *JINST* **8** (2013) P07011, arXiv:1306.4605 [physics.ins-det]. [Erratum: JINST8,E09001(2013)].

[68] R. Veenhof, "GARFIELD, recent developments," *Nucl. Instrum. Meth. A* **419** (1998) 726–730.

[69] **MicroBooNE** Collaboration, C. Adams *et al.*, "Ionization electron signal processing in single phase LArTPCs. Part I. Algorithm Description and quantitative evaluation with MicroBooNE simulation," *JINST* **13** no. 07, (2018) P07006, arXiv:1802.08709 [physics.ins-det].

[70] **MicroBooNE** Collaboration, C. Adams *et al.*, "Ionization electron signal processing in single phase LArTPCs. Part II. Data/simulation comparison and performance in MicroBooNE," *JINST* **13** no. 07, (2018) P07007, arXiv:1804.02583 [physics.ins-det].

[71] W. Gu, "Wire-cell signal processing in protodune," tech. rep. https://indico.fnal.gov/event/19185/contribution/1/material/slides/1.pptx.

[72] Z. Dong, K. Knoepfel, M. Lin, B. Viren, and H. Yu, "Evaluation of Portable Programming Models to Accelerate LArTPC Detector Simulations," in *20th International Workshop on Advanced Computing and Analysis Techniques in Physics Research: AI Decoded - Towards Sustainable, Diverse, Performant and Effective Scientific Computing*. 3, 2022. arXiv:2203.02479 [physics.ins-det].

[73] J. S. Marshall and M. A. Thomson, "The Pandora Software Development Kit for Pattern Recognition," *Eur. Phys. J.* **C75** no. 9, (2015) 439, arXiv:1506.05348 [physics.data-an].

[74] Baller, Bruce https://cdcvs.fnal.gov/redmine/documents/1026.

[75] R. Sulej and D. Stefan http://larsoft.org/single-record/?pdb=102.

[76] X. Qian, C. Zhang, B. Viren, and M. Diwan, "Three-dimensional Imaging for Large LArTPCs," *JINST* **13** no. 05, (2018) P05032, arXiv:1803.04850 [physics.ins-det].

[77] **MicroBooNE** Collaboration, P. Abratenko *et al.*, "Wire-Cell 3D Pattern Recognition Techniques for Neutrino Event Reconstruction in Large LArTPCs: Algorithm Description and Quantitative Evaluation with MicroBooNE Simulation," *JINST* **17** (2022) P01037, arXiv:2110.13961 [physics.ins-det].

[78] **MicroBooNE** Collaboration, P. Abratenko *et al.*, "Search for an Excess of Electron Neutrino Interactions in MicroBooNE Using Multiple Final State Topologies," arXiv:2110.14054 [hep-ex].

[79] **MicroBooNE** Collaboration, P. Abratenko *et al.*, "Search for an anomalous excess of inclusive charged-current ν_e interactions in the MicroBooNE experiment using Wire-Cell reconstruction," arXiv:2110.13978 [hep-ex].

[80] M. Wang, T. Yang, M. Acosta Flechas, P. Harris, B. Hawks, B. Holzman, K. Knoepfel, J. Krupa, K. Pedro, and N. Tran, "GPU-accelerated machine learning inference as a service for computing in neutrino experiments," arXiv:2009.04509 [physics.comp-ph].

[81] H. B. W. Group, "HEP-SPEC06 (HS06) Benchmark," 2022. http://w3.hepix.org/benchmarking.html.

[82] M. M. DUNE Near Detector Data Model Committee *et al.*, "DUNE Near Detector Data Model," DUNE doc 24735, 2020. https://docs.dunescience.org/cgi-bin/private/ShowDocument?docid=24735.

[83] F. Drielsma, K. Terao, L. Dominé, and D. H. Koh, "Scalable, End-to-End, Deep-Learning-Based Data Reconstruction Chain for Particle Imaging Detectors," in *34th Conference on Neural Information Processing Systems*. 2, 2021. arXiv:2102.01033 [hep-ex].

[84] **DeepLearnPhysics** Collaboration, L. Dominé and K. Terao, "Scalable deep convolutional neural networks for sparse, locally dense liquid argon time projection chamber data," *Phys. Rev. D* **102** no. 1, (2020) 012005, arXiv:1903.05663 [hep-ex].

[85] **DeepLearnPhysics** Collaboration, L. Dominé, P. C. de Soux, F. Drielsma, D. H. Koh, R. Itay, Q. Lin, K. Terao, K. V. Tsang, and T. L. Usher, "Point proposal network for reconstructing 3D particle endpoints with subpixel precision in liquid argon time projection chambers," *Phys. Rev. D* **104** no. 3, (2021) 032004, arXiv:2006.14745 [hep-ex].

[86] **DeepLearnPhysics** Collaboration, F. Drielsma, Q. Lin, P. C. de Soux, L. Dominé, R. Itay, D. H. Koh, B. J. Nelson, K. Terao, K. V. Tsang, and T. L. Usher, "Clustering of electromagnetic showers and particle interactions with graph neural networks in liquid argon time projection chambers," *Phys. Rev. D* **104** no. 7, (2021) 072004, arXiv:2007.01335 [physics.ins-det].

[87] **DeepLearnPhysics** Collaboration, D. H. Koh, P. Côte De Soux, L. Dominé, F. Drielsma, R. Itay, Q. Lin, K. Terao, K. V. Tsang, and T. L. Usher, "Scalable, Proposal-free Instance Segmentation Network for 3D Pixel Clustering and Particle Trajectory Reconstruction in Liquid Argon Time Projection Chambers," arXiv:2007.03083 [physics.ins-det].

[88] K. T. Dae Heun Koh, Aashwin Mishra, "Evaluating Deep Learning Uncertainty Quantification Methods for Neutrino Physics Applications," in *35th Conference on Neural Information Processing Systems*. 2021.

[89] C. Adams, K. Terao, and T. Wongjirad, "PILArNet: Public Dataset for Particle Imaging Liquid Argon Detectors in High Energy Physics," arXiv:2006.01993 [physics.ins-det].

[90] **MicroBooNE** Collaboration, R. Acciarri *et al.*, "Convolutional Neural Networks Applied to Neutrino Events in a Liquid Argon Time Projection Chamber," *JINST* **12** no. 03, (2017) P03011, arXiv:1611.05531 [physics.ins-det].

[91] A. Radovic, M. Williams, D. Rousseau, M. Kagan, D. Bonacorsi, A. Himmel, A. Aurisano, K. Terao, and T. Wongjirad, "Machine learning at the energy and intensity frontiers of particle physics," *Nature* **560** no. 7716, (2018) 41–48.

[92] **MicroBooNE** Collaboration, C. Adams *et al.*, "Deep neural network for pixel-level electromagnetic particle identification in the MicroBooNE liquid argon time projection chamber," *Phys. Rev. D* **99** no. 9, (2019) 092001, arXiv:1808.07269 [hep-ex].

[93] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature* **323** no. 6088, (1986) 533–536. https://doi.org/10.1038/323533a0.

[94] J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable parallel programming with cuda: Is cuda the parallel programming model that application developers have been waiting for?," *Queue* **6** no. 2, (Mar, 2008) 40–53. https://doi.org/10.1145/1365490.1365500.

[95] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds., pp. 8024–8035. Curran Associates, Inc., 2019. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[96] P. V. C. Hough, "Machine Analysis of Bubble Chamber Pictures," *Conf. Proc. C* **590914** (1959) 554–558.

[97] N. Otsu, "A Threshold Selection Method from Gray-level Histograms," *IEEE Transactions on Systems, Man and Cybernetics* **9** no. 1, (1979) 62–66. http://dx.doi.org/10.1109/TSMC.1979.4310076.

[98] M. Ester, H. peter Kriegel, J. S, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pp. 226–231. AAAI Press, 1996.

[99] C. Backhouse and D. Rocco, "A data summary file structure and analysis tools for neutrino oscillation analysis at the NOvA experiment," *J. Phys. Conf. Ser.* **664** no. 7, (2015) 072038.

[100] Backhouse, C., "The CAFAna framework for neutrino analysis," 2021. https://www.snowmass21.org/docs/files/summaries/CompF/SNOWMASS21-CompF5-003.pdf.

[101] N. Smith *et al.*, "Coffea: Columnar Object Framework For Effective Analysis," *EPJ Web Conf.* **245** (2020) 06012, arXiv:2008.12712 [cs.DC].

[102] P. Stowell *et al.*, "NUISANCE: a neutrino cross-section generator tuning and comparison framework," *JINST* **12** no. 01, (2017) P01016, arXiv:1612.07393 [hep-ex].

[103] **NOvA, R. Group** Collaboration, M. A. Acero *et al.*, "An Improved Measurement of Neutrino Oscillation Parameters by the NOvA Experiment," arXiv:2108.08219 [hep-ex].

[104] **T2K Collaboration** Collaboration, K. Abe *et al.*, "Improved constraints on neutrino mixing from the t2k experiment with 3.13×10^{21} protons on target," *Phys. Rev. D* **103** (Jun, 2021) 112008. https://link.aps.org/doi/10.1103/PhysRevD.103.112008.

[105] C. Green, J. Kowalkowski, M. Paterno, M. Fischler, L. Garren, *et al.*, "The art framework," *J.Phys.Conf.Ser.* **396** (2012) 022020.

[106] J. Blomer, P. Canal, A. Naumann, and D. Piparo, "Evolution of the ROOT Tree I/O," *EPJ Web Conf.* **245** (2020) 02030, arXiv:2003.07669 [cs.DB].

[107] **ROOT Team** Collaboration, G. Amadio *et al.*, "Software Challenges For HL-LHC Data Analysis," arXiv:2004.07675 [physics.data-an].

[108] Frameworks Task Force, "Software Frameworks Taskforce Report," tech. rep., DUNE, 2021. https://docs.dunescience.org/cgi-bin/private/ShowDocument?docid=21934.

[109] Frameworks Task Force, "Software Framework Requirements - HSF Review Findings," tech. rep., HSF, 2021. https://docs.dunescience.org/cgi-bin/private/ShowDocument?docid=24423.

[110] Frameworks Task Force, "Summary of DUNE Workshop on Parallelism - response to HSF findings," tech. rep., DUNE, 2021. https://docs.dunescience.org/cgi-bin/private/ShowDocument?docid=24426.

[111] S. Gollapinni, K. Mahn, and J. Maneira, "DUNE Metadata Task Force Report Draft ,". https://docs.dunescience.org/cgi-bin/private/ShowDocument?docid=22984.

[112] I. Mandrichenko, "Conditions Databases at FNAL," tech. rep., Fermilab", 2019. https://lss.fnal.gov/archive/2019/poster/fermilab-poster-19-137-scd.pdf.

[113] M. Bracko, M. Clemencic, D. Dykstra, A. Formica, G. Govi, M. Jouvin, D. Lange, P. Laycock, and L. Wood, "HEP Software Foundation Community White Paper Working Group - Conditions

Data," arXiv:1901.05429 [physics.comp-ph].

[114] M. J. *et al.*, "Ifbeam documentation." "https://cdcvs.fnal.gov /redmine/projects/ifbeamdata/documents", 2016.

[115] M. Verzocchi, "LBNF / DUNE Parts Identifier," CERN EDMS 2505353, DUNE, 2021. https://edms.cern.ch/document/2505353.

[116] **DUNE** Collaboration, B. Abi *et al.*, "Prospects for beyond the Standard Model physics searches at the Deep Underground Neutrino Experiment," *Eur. Phys. J. C* **81** no. 4, (2021) 322, arXiv:2008.12769 [hep-ex].

[117] **DUNE** Collaboration, D. Totani *et al.*, "A measurement of absolute efficiency of the ARAPUCA photon detector in liquid argon," *JINST* **15** no. 06, (2020) T06003, arXiv:2008.05371 [physics.ins-det].

[118] J. Klein, "DUNE Data Volumes," DUNE doc or LBNF doc 14983, 2019. https://docs.dunescience.org/cgi-bin/private/ShowDocument?docid=14983.

[119] Schellman, H., "Updated data volume estimates for DUNE through 2040," tech. rep., DUNE, 2022. https://docs.dunescience.org/cgi-bin/private/ShowDocument?docid=24732&version=1.

[120] Schellman, H., "Numbers for data size estimates," tech. rep., DUNE, 2019. https://docs.dunescience.org/cgi-bin/private/ShowDocument?docid=16028&version=1.

[121] "CMS Offline and Computing Public Results," 2022. https://twiki.cern.ch/twiki/bin/view/CMSPublic/CMSOfflineComputingResults.

[122] D. Box, "Fife-jobsub: a grid submission system for intensity frontier experiments at fermilab," in *Journal of Physics: Conference Series*, vol. 513, 3, p. 032010, IOP Publishing. 2014.

[123] 2022. https://kubernetes.io/docs/setup/best-practices/cluster-large/.

[124] 2022. https://coffeateam.github.io/.

[125]

[126] J. Blomer, "A quantitative review of data formats for HEP analyses," *J. Phys. Conf. Ser.* **1085** no. 3, (2018) 032020.

[127] "Hierarchical Data Format 5," 2021. https://portal.hdfgroup.org/display/HDF5/HDF5.

[128] E. Rodrigues, "The Scikit-HEP Project," *EPJ Web Conf.* **214** (2019) 06005, arXiv:1905.00002 [physics.comp-ph].

[129] FIFE, "Understanding storage volumes," tech. rep., 2020. https:

//cdcvs.fnal.gov/redmine/projects/fife/wiki/Understanding_storage_volumes.

[130] A. P. Millar, G. Behrmann, C. Bernardt, P. Fuhrmann, D. Litvintsev, T. Mkrtchyan, A. Petersen, A. Rossi, and K. Schwank, "dCache: Big Data storage for HEP communities and beyond," *J. Phys. Conf. Ser.* **513** (2014) 042033.

[131] Andrew Norman, "DUNE Data Management Plan," tech. rep., DUNE, 2017. https://publicdocs.fnal.gov/cgi-bin/ShowDocument?docid=23.

[132] R. A. Illingworth, "A Data Handling System for Modern and Future Fermilab Experiments," *J. Phys. Conf. Ser.* **513** (2014) 032045.

[133] M. Barisits *et al.*, "Rucio - Scientific Data Management," arXiv:1902.09857 [cs.DC].

[134] **DUNE** Collaboration, I. Mandrichenko, "MetaCat - metadata catalog for data management systems," *EPJ Web Conf.* **251** (2021) 02048.

[135] Baritsis, M.and others, "Rucio: Scientific Data Management," *Computing and Software for Big Science* **3** no. 11, (2019) .

[136] A. Kiryanov, A. A. Ayllon, M. Salichos, and O. Keeble, "FTS3 - A File Transfer Service for Grids, HPCs and Clouds," *PoS* **ISGC2015** (2015) 028.

[137] J. Zurawski, B. Brown, D. Carder, E. Colby, E. Dart, K. Miller, A. Patwa, K. Robinson, L. Rotman, and A. Wiedlea, "High energy physics network requirements review (final report, july-october 2020)," (6, 2021) . https://www.osti.gov/biblio/1804717.

[138] G. L. Miotto, A. Thea, A. Kaboth, K. Biery, J. Brooke, P. Hamilton, J. Klein, P. Rodrigues, R. Sipos, and A. Tapper, "Daq project specification document: Trigger and data acquisition (tdaqf) system specifications," tech. rep. https://edms.cern.ch/ui/file/2679120/2/DUNE_DAQ_System_Specifications.pdf.

[139] M. Mengel, S. White, V. Podstavkov, M. Wiersma, A. Mazzacane, and K. Herner, "Production Operations Management System (POMS) for Fermilab Experiments," *EPJ Web Conf.* **245** (2020) 03024.

[140] I. Sfiligoi, D. C. Bradley, B. Holzman, P. Mhashilkar, S. Padhi, and F. Wurthwein, "The pilot way to grid resources using glideinwms," in *2009 WRI World congress on computer science and information engineering*, vol. 2, pp. 428–432, IEEE. 2009.

[141] P. Mhashilkar, M. Altunay, E. Berman, D. Dagenhart, S. Fuess, B. Holzman, J. Kowalkowski, D. Litvintsev, Q. Lu, A. Moibenko, *et al.*, "Hepcloud, an elastic hybrid hep facility using an intelligent decision support system," in *EPJ Web of Conferences*, vol. 214, p. 03060, EDP Sciences. 2019.

[142] K. Herner, A. F. A. Hernandez, S. Bhat, D. Box, J. Boyd, B. Coimbra, V. Di Benedetto, P. Ding, D. Dykstra, M. Fattoruso, *et al.*, "Advances and enhancements in the fabric for frontier

experiments project at fermilab," in *EPJ Web of Conferences*, vol. 214, p. 03059, EDP Sciences. 2019.

[143] G. M. Kurtzer, V. Sochat, and M. W. Bauer, "Singularity: Scientific containers for mobility of compute," *PLoS ONE* **12** no. 5, (5, 2017) .

[144] H. ipv6 task force, "Current status of ipv6 Task Force," twiki resource NNNN, HEPIX, 2021. `https://twiki.cern.ch/twiki/bin/view/LCG/WlcgIpv6`.

[145] CERN, "ETF documentation web page," Read the Docs NNNN, 2016. `http://etf.cern.ch/docs/latest/index.html`.

[146] M. Babik, "Current status of ETF testing," 2020. `https://indico.cern.ch/event/970604/#13-etf-update`.

[147] K. Herner, D. Box, J. Boyd, V. DiBenedetto, P. Ding, D. Dykstra, M. Fattoruso, G. Garzoglio, T. Levshina, M. Kirby, A. Kreymer, A. Mazzacane, M. Mengel, P. Mhashilkar, V. Podstavkov, K. Retzke, and N. Sharma, "The FIFE project at Fermilab: Computing for experiments," in *Proceedings of 38th International Conference on High Energy Physics — PoS(ICHEP2016)*, p. 176. 02, 2017.

[148] "MediaWiki home page.". `http://www.mediawiki.org/`.

[149] 2022. `https://dune.github.io/computing-training-basics/index.html`.