CERN-ACC-2018-0060

18 December 2018

Giovanni.Rumolo@cern.ch

Report from the ABP Computing Working Group (ABP-CWG) activities in 2017-18

G. Rumolo, X. Buffat, R. De Maria, L. Deniau, and G. Iadarola

Keywords: Numerical simulations, computing resources, code development

Summary

The Accelerators and Beam Physics (ABP) Group is responsible for studying and understanding the beam dynamics over the complete CERN Accelerator Complex through theoretical, simulation and experimental studies. For this purpose, over the years members of the ABP group have developed and/or become proficient users of a large number of simulation codes, needed for all studies covering hadron sources and linear accelerators, machine optics and single particle dynamics, coherent and incoherent collective effects. Use cases are meant to cover not only the current CERN accelerators, but also the design and operating scenarios of future accelerators (e.g. CLIC, FCC). The activities of development, deployment, maintenance and support of the accelerator physics computer codes are coordinated in the framework of the ABP Computing Working Group. The purpose of this note is to introduce the ABP Computing Working Group in terms of mandate and functions, and provide a compact summary of the activities carried out in 2017-18, while highlighting the points that require close follow up.

CERN-ACC-2018-0060
18/12/2018

1 Introduction

The ABP Computing Working Group (ABP-CWG) has started operating in September 2016 with the mandate to:

- Organize the communication within the ABP group on the subject of software and hardware resources and needs. Publish and share the information through a reference website and e-group.
- Represent the ABP group for software and hardware inside and outside CERN. Establish collaborations with partners to share experience and expertise, and take advantage from hardware made available.
- Establish and maintain the list of software[1] under ABP responsibility, or strategic

[1] Compilers, libraries, tools and websites are part of this list.

for ABP activities, used for studies in the fields of beam physics (simulation and data analysis). This includes software developed in-house, products provided by other laboratories or universities, and commercial products (usually with licenses).

- Establish and maintain the list of hardware available to ABP inside and outside CERN for beam physics studies.
- Identify and periodically review the needs for:
  - software development and upgrade to accurately simulate the beam processes in view of the optimisation of the operational performance of existing accelerators and the design of future machines.
  - hardware procurement to support the point above.
- Enhance transversal synergies within the ABP group, provide estimates of required resources to fulfil the identified needs and to ensure proper continuity and optimal use of available resources.
- Provide expertise, recommendations and guidelines[2] for ABP software development and maintenance, and hardware commitment through its members and partners. Establish a list of recommended tools and introductory documentations essential for daily use.

To fulfil the given mandate, the ABP-CWG activity in 2017 and 2018 has been to:

- Create and maintain a TWiki home page editable by any member of the ABP group, which collects the relevant information for both beam physics software developers and code users;
- Hold regular meetings (once or twice a month);
- Connect with IT by both participating in the IT Users Meetings for mutual communication and having ad-hoc meetings with competent IT representatives to follow up on specific questions (e.g., High Performance Computing (HPC) facilities, AFS phaseout, migration from LSF to HTCondor);
- Follow up and steer the set up of HPC facilities (notably MPI and Graphics Processors Units (GPU) clusters) responding to current and projected ABP needs.

2 Channels of information

A TWiki Home Page for the ABP-CWG has been created (`http://cern.ch/abp-computing`), which can be edited by all members of the ABP group and collects the relevant information following the scheme indexed on its home page. This page has been serving the main purpose

[2]Disclaimer: the goal here should be to create a platform with a robust internal support for code development within ABP and not to impose technical choices.

to share experience, but also to instruct newly hired group members about CERN computing matters and basic installations as well as to troubleshoot miscellaneous computing issues (e.g. migration from LSF to HTCondor).

Besides, regular meetings have been held throughout the years 2017 and 2018 (Indico category https://indico.cern.ch/category/8448/) in order to:

- Discuss arising computing matters, collect issues, disseminate potentially useful information in terms of computing resources and tools;
- Review the computer programs used in ABP, and try to make the point on their status, perspectives for improvement and desiderata in terms of resources.

The emphasis in 2017 was on the latter point, and a large number of codes were described by the authors, contributors or main users. The main points emerged from the discussions at these meetings have been summarised in the next section. In 2018, meetings were mainly organised to discuss specific computing matters, like the use of the HPC cluster, the illustration of SWAN (Service for Web based ANalysis), the collection of questions about the move from the present CERN Accelerator Logging Service (CALS) and Next CALS (NXCALS) in terms of accessibility of the data (speed, available API, connection from Technical Network). This last item is still very important to follow up due to its potential impact on an important fraction of software packages developed within ABP, which are oriented to data extraction and visualisation.

3 The beam dynamics codes and perspectives for development

A great deal of ABP activities are strongly founded on more or less complex simulation codes related to beam dynamics. These simulation codes represent to a large extent the interface of the ABP group with the rest of CERN and with the external world, both through the distribution and support of internally developed tools and through the active use of simulation results to design machines or interpret and predict machine observations to provide operational settings.

Typically, these codes either describe the propagation of a beam of particles inside different types of accelerators under a variety of different effects or calculate the relevant quantities to write the equations of motion of the single particles in the beam dynamics codes (e.g. optical functions, wake functions, electron cloud distributions). The particle beam is in some codes modelled as an ensemble of macroparticles (macroparticle codes), computationally intensive but quite general, while in other codes it is described through a distribution function and its decomposition in modes (Vlasov solvers), usually much faster but based on limiting sets of assumptions. Already in what concerns the present CERN accelerators, a broad range of cases need to be covered, spanning basically from the proton/ion source with plasma discharge and low energy beam transport to low energy linear accelerators and low-to-high energy synchrotrons in presence of all types of nonlinear or collective interactions, all the way to the beam storage at very high energy (6.5 TeV) to provide collisions in LHC with beam-beam interactions and synchrotron radiation. Besides, even larger hadron machines like the

Code name	Category	Development
LHC Online Model, MAD-X, MapClass, PyOptics, SixTrack, SixTrackLib	Optics and single particle tracking	In-house
ABCI, ACE3P, CST Particle Studio, GdFiDl, HFSS	Electromagnetic solvers (wake functions, beam coupling impedances)	External
ImpedanceWake2D, TLWall		In-house
IBSimu, Ninja, ONIX	Plasma discharge, low energy tracking	External
DELPHI, BimBim	Vlasov solver, circulant matrix	In-house
NHTS, MOSES		External
COMBI, FASTION, GuineaPig, PyECLOUD, PyHEADTAIL, PyPARIS, PyPIC, PySSD, RF-Track, SIRE, TRAIN	Tracking with collective effects (including space charge, beam-beam, electron cloud, impedance, ions, IBS, electron cooling)	In-house
PyORBIT		External
PLACET, PLACET2, PATH	Tracking for linear accelerators and linear colliders	In-house

Table 1: List of codes used or developed within ABP with short descriptions (category, central column) and type of development (at CERN or outside, right column).

FCC-hh or linear and circular lepton colliders (CLIC, FCC-ee), and their injector complexes, also need to be designed and studied to prepare the ground for future projects. An exhaustive list of beam dynamics codes and open projects in the field of beam physics is available on the official web page of the ABP-CWG, namely `https://cern.ch/abp-computing`, under Beam Physics Software Tools. Based on the information extracted from this web site, Table 1 provides a snapshot of the beam dynamics codes presently used and/or developed within ABP. A few other programs for data extraction and/or analysis (e.g. SUSSIX, PyTimber, PageStore) as well as extensive libraries of Python scripts to retrieve and visualise machine data or automatise Machine Development (MD) procedures have not been included in this list, although their development and dissemination equally represents a fundamental software activity under ABP.

As labeled in the last column of Table 1, some of these codes have been developed in-house, with the strong advantage of having an in-depth knowledge of the underlying physics models, full control on the source code as well as the capability of introducing modifications or extensions as needed. However, for some in-house codes the original development was made long ago and there have been over the years only few or intermittent efforts of modernisation, clean-up or performance improvement. An important problem for codes that have not undergone regular maintenance and/or upgrade is also the presence of unwanted legacy from temporary developers, which is often undocumented and untested. The fact that, in addition, the main developers may have retired or left CERN renders the code maintenance and management challenging in their present status. For the majority of in-house codes, however, the development is more recent and still actively ongoing.

Some other codes are (or have been) developed elsewhere and ABP members are (usually expert) users. These codes can either require a license or they are free to use. In the former case, typically only executable versions are available (e.g. CST Particle Studio), or they must be run by some other teams with ABP only providing input files (e.g, one of the codes for the source modelling). Most of the codes free to use are also open-source. In this case, ABP members have the opportunity of contributing actively to their development by adding features to better suit the studies to be made at CERN (e.g. PyORBIT).

Many of the ABP simulation codes exhibit a clear path to further development, which requires an investment in terms of human and hardware resources. From the point of view of the physics described in the codes, new features have been implemented and tested in the existing codes (e.g. low energy tracking in SixTrack; new elements like RFQ, space charge and full deployment of multi-bunch in PyHEADTAIL; multi-species tracking in the PyECLOUD-PyHEADTAIL suite) and more can be added to enlarge the range of coverable cases. Besides, as the correct modelling of all the beam dynamics phenomena is instrumental to predict and interpret characteristic beam behaviors or adverse effects on machine systems, in recent years a significant effort has been made for the development of ever more unified, high-performance and flexible numerical simulation tools. However, the process is long and the momentum needs to be kept up in order to steadily advance and not fall back, losing the acquired status.

Flexibility is needed to cover the broad range of simulation scenarios that are of interest for the LHC, its injector chain and future accelerators. In this respect it has been mainly agreed to abandon the idea of a single block code with rigid user interface in favour of a set of tools, typically in the form of Python libraries. The user can benefit from the power of Python scripting to define arbitrarily complex simulation setups combining different libraries and use the multitude of freely available Python packages to perform post-processing, data analysis and storage, plotting. The Python language is also used for a large fraction of the implementation, which proved to ease significantly the process of development and maintenance. Compiled languages, such as C and FORTRAN, keep being used for computationally intensive tasks, using different tools, e.g. cython and f2py to interface the resulting modules to the Python code.

To improve the performance of the different codes, there has been lately a concerted effort to move towards parallel computing based on CPU multiprocessing (e.g. MPI), CPU multithreading (e.g. Open-MP) and GPU. The goal is to simulate situations that, due to the complexity of the model or the number of effects that need to be included, would require prohibitively long computing times with the traditional single processor implementation. Some examples are given here below:

- In the PyECLOUD-PyHEADTAIL suite, electron cloud instability simulations have been organized through multiple processes in different CPUs taking place in a ring structure. Each process represents the interaction of the beam with a fraction of the accelerator. The different beam slices are propagated along the ring of processors, just as they would propagate in the real accelerator. At the end of each turn all particles in each bunch need to be collected by a master process for global longitudinal tracking and for updating the bunch slicing. The parallelization is realized by means of an additional layer, also implemented in Python, in order to keep physics and parallelization decoupled while at the same time minimizing the number of changes in pre-existing tools.

- Space charge simulations have been accelerated using GPU within PyHEADTAIL. In this case, a context manager has been put in place to switch between GPU and CPU contexts. These simulations are based on the three essential steps of distributing macroparticles as a charge density on mesh points, solving the Poisson equation (2D or 3D), and calculating the electric fields at the macroparticle positions from the potential at the mesh points. An important performance gain using GPU has been demonstrated for all three steps, leading to an overall improvement by over one order of magnitude in terms of computing time when using large numbers of mesh points.

- The simulation of impedance driven coupled bunch instabilities as well as realistic multi-bunch feedback systems have become possible with MPI parallelization at the bunch level in PyHEADTAIL. Hereby, individual bunches are spread across multiple processors. Bunch generation, tracking and signal recording have been massively parallelised. Although the interaction through impedances and the signal processing channels of the feedback systems require communication across processors, the overall implementation allows profiting considerably from the parallelization and overcoming limitations in both memory and computational speed. Thus, multi-bunch simulations with several thousands of bunches have become possible e.g. to study future large colliders.

- Concerning single particle tracking, a new standalone C library with Python bindings, called SixTrackLib and based on classical techniques of nonlinear tracking (original SixTrack engine), has been designed to enable GPU computation and to be possibly plugged in codes dealing with other effects, thus replacing simplified transport models with realistic element-by-element tracking. This library, under active development, is now being optimized for execution speed and validated on LHC and SPS (while being applicable also to low energy rings).

- Colliders feature several bunches in each of the two beams, which interact electromagnetically head-on at the interaction points and long-range within the common vacuum chambers. To model these beams requires a code capable of dealing efficiently with the memory representing the bunches, the communication between them in order to compute their respective nonlinear fields, as well as with the different computational requirements needed for each interaction type. A hybrid parallelisation scheme has been designed for COMBI, based on MPI to share the load between compute nodes and OpenMP to profit from efficient shared memory within each node. This allows the study of the preservation of beam quality and stability in the presence of beam-beam interactions as well as the interplay with the other effects included in COMBI.

Thanks to all these efforts it is nowadays possible to efficiently simulate scenarios previously deemed inaccessible, such as multi-bunch simulations of long trains under the effect of long-lived wake fields, simulations of combined electron cloud formation and multi-bunch beam evolution, space charge simulations in presence of impedance, instability or space charge simulations over long energy plateaus.

A team of dedicated resources with specific code development objectives would be the key to keep up beam dynamics code development and maintenance as a core activity of the ABP group. A few points for future development both in the near and long term are outstanding and could be pursued by deploying and focusing an adequate number of resources, in particular:

- The migration of the existing Python codes (and scripts to analyse machine data) to Python 3 (motivated by the end of support for Python 2 at the end of 2019);
- A general performance analysis for algorithm optimisations (possibly in close interaction with IT experts from IT department or brought in on specific projects);
- The unification of duplicate codes (notably, in the realm of collective effects, see Table 1) by focusing on a few selected central tools and programming/testing the required modules to preserve the present level of detail (see next item);
- The construction and extraction from existing codes of potentially sharable library modules, such that any functionality they embed will no longer remain the exclusive property of one code only but can be re-used also in other contexts;
- The continuity of the maintenance of specific features (not always in the form of modules) in the existing codes beyond the departure of the developer(s) (inheritance), or alternatively the reset and clean-up, if deemed necessary for long-term code maintainability and reliability;
- The establishment of a general tracking engine with different options (degrees of complexity) and its interface with other multi-purpose codes.

The items above represent a non-exhaustive part of the list of tasks over the next few years necessary to consolidate the present status of know-how in numerical simulations for beam dynamics within ABP and create the basis for a robust set of tools (building bricks) to serve as a common framework for future accelerator physics studies.

4 Link to IT

In order to keep up to date with all the news from CERN IT, L. Deniau participated in all the quarterly IT Users Meetings throughout 2017-18 and provided detailed summaries highlighting the main information relevant for ABP, while giving voice to ABP concerns on specific items during the meetings. A few questions with large impact for the ABP members in terms of computing environment, such as the AFS phaseout, migration from LSF to HTCondor, set up of HPC facilities, have been closely followed up with additional meetings with IT contact experts.

In 2017, the migration from LSF to HTCondor to launch batch jobs was coordinated through all its phases. The new system was first tested with both SixTrack and PyHEADTAIL sample jobs. At the same time, the existing LSF groups were mapped into HTCondor e-groups to transfer the existing system of priorities to the new system. Then the effective migration of 50% of the resources was put in place through May 2017 and the full migration was ended by June 2017. Several issues appeared during the first few months of exploitation, the most common of which were jobs failing to launch or stopped/rescheduled, or impossibility to access AFS, usually due to kerberos token issues. However, all the issues could be systematically tackled and solved with the relevant IT experts until the new phase of full and smooth deployment of the new batch system was finally reached by the end of 2017.

Concerning the AFS phaseout, an AFS use survey was run across ABP in 2017 to feed IT with clear information about the significant impact of this phaseout without an appropriate replacement file system in place. The candidate infrastructure to replace AFS, namely EOS, has proved to have attractive features like ample storage space, but needs a stable and performant file system to access it. However, the present file system associated to EOS, named EOS FUSE, still exhibits reliability and speed issues. As of end 2018, EOS FUSE keeps being under development, with no official release dates and no clear statement about the solution of the issues encountered. It also remains unclear whether EOS FUSE will be accessible from the technical network, as is now AFS. The AFS phaseout is therefore a critical issue, which needs close monitoring over the next few months, as the announced phaseout deadline has been set to end of Long Shutdown 2 (LS2), i.e. 2020.

Following the collection of the needs for HPC throughout different groups within the Accelerator and Technology Sector (ATS), the purchase and set up of an HPC cluster at CERN took place over 2017. The usage of this cluster is constantly reviewed by IT with the relevant users in order to check the level of occupancy as well as the adequateness of the resources to the present needs CERN-wide.

5 Hardware resources

The new frontiers of code development described in one of the previous sections have only become reachable thanks to an important ramp up of the dedicated hardware resources that CERN has put in place over the last few years. More precisely, the following actions have been taken to provide code developers with the necessary hardware framework:

- The deployment at CERN of a large HPC infrastructure to support parallel computation across the ATS. The HPC facility proper consists of 2 Infiniband clusters of 1440 cores and a batch cluster of about 2000 cores with low-latency Ethernet interconnects (see Fig. 1). It is intended for MPI jobs spanning between 32 and 1000+ cores and complements about 150000 shared cores available on the regular High Throughput Computing (HTC) batch facility. This HPC facility has been set up under the strong request from the ATS sector, after a survey on the needs for this type of cluster has been conducted over several groups across departments and has demonstrated the existence of a potentially large pool of users. After its start of operation at the end of 2017, regular meetings between IT and representatives of the major user groups have been taking place, in which the general usage of the cluster was analysed (always above

Figure 1: HPC cluster at CERN. Courtesy of N. Høimyr.

50%) and its adequacy to the current demand assessed. The current MPI capacity is shared between the active user groups in the BE, EN, TE and HSE departments. Fair-shares of the batch-system ensure the correct prioritization in the case of congestion and queuing. Additional CPU requirements expected to emerge over the next months (especially in view of the upcoming LS2 with the related increase of volume of simulations) have been compensated with additional multi-core capacity under HTCondor (see next item).

- The deployment of dedicated batch nodes with 16 or 32 cores on the HTCondor facility to run parallel multi-thread jobs using up to 32 cores. A further increase of the capacity with 16 and 32 core HTCondor batch nodes was provided in September 2018 by 40 additional IT server purchases with a contribution of HSE funds, following a review of the available MPI and HPC capacity.

- The collaboration between the ABP group and the INFN-CNAF center, in Bologna (Italy), which houses an HPC cluster of 832 cores with fast connection and provides administration support for CERN. The cluster has been for most part financed under the LHC Injectors Upgrade (LIU) and High Luminosity LHC (HL-LHC) projects umbrella and is intensively used for heavy space charge and electron cloud simulations. This cluster began operation in summer 2017 with 384 cores (12 servers of 32 cores each), bought under Phase One of the collaboration agreement KE3108 [1]. The performance of these machines and its scalability with the number of processors was checked following the steps of the acceptance procedure reported in the collaboration agreement (space charge test runs conducted with the PyORBIT code) and the results were published in the form of a CERN note [2]. The success and the documentation of the acceptance tests allowed triggering Phase Two of the agreement, which foresaw the enlargement of the cluster by an additional 256 cores. While the procedure was unfortunately slowed down by a flood, which damaged 4 servers and caused the remaining 8 servers to be moved for over six months to another location, the purchase of the new servers in 2018 could be made at a much more competitive price than expected. This allowed for the addition of 448 cores to the cluster (instead of the planned 256) at a global cost 25% lower than what was originally planned. Since summer 2018, the HPC cluster at CNAF is running full steam with 832 cores and the collaboration agreement has been amended to extend its period of validity till the end of 2022.

- The purchase of GPU servers by the ABP group. The first one was bought in 2012 under sponsorship of LIU PS and features one Nvidia Tesla C2075 2 TFlop (single precision) GPU. Over the years, this server was extensively used for space charge development in PyHEADTAIL. The one procured in 2017 features four Nvidia Tesla V100 7 TFlops (double precision) GPUs. It is presently running at CERN and widely used for GPU code development within ABP (space charge, SixTrackLib). The plan is to ship it to CNAF as a temporary delivery and have it managed as an additional resource within the existing HPC cluster described at the above point.

Details about these hardware resources and instructions on how to obtain accounts and run jobs can be found on the official web page of the ABP-CWG, namely `https://cern.ch/abp-computing`, under Computing Resources.

6 Conclusions

The ABP-CWG has been working effectively over 2017-18 with tangible results in terms of distribution of the information related to computing matters within ABP, direct link of ABP to the IT department, analysis of the beam dynamics codes within ABP and assessment of the needs for development, procurement of modern hardware for HPC. The necessary hardware infrastructure to promote intensive HPC software development has been thus made available, thanks to which numerous development activities have been successfully undertaken and future priorities could be clearly identified (also in the long term). The new computing infrastructure has been the cornerstone to generate an important HPC upgrade of the existing computational tools for beam dynamics and, provided that the necessary resources are allocated and the objectives are correctly set, will serve the purpose to foster continued progress in this direction during LS2 and well beyond.

7 Acknowledgements

The authors would like to thank all those who actively contributed to the activities described in this paper, in particular Nils Høimyr, Ben Jones, Carolina Lindqvist, Pablo Llopis Sanmillan (IT-CM); Daniele Cesini, Antonio Falabella, Gaetano Maron (CNAF Bologna); Gianluigi

Arduini, Hannes Bartosik, Lotta Mether, Adrian Oeftiger, Guido Sterbini (BE-ABP); Kevin Li (BE-OP). All the speakers at the ABP-CWG meetings are also warmly thanked for sharing the information of their codes with us and for contributing to disseminate information on miscellaneous computing matters within ABP. Finally, many thanks also to the regular attendees of the ABP-CWG meetings for their active participation and inputs.

References

[1] Collaboration agreement Contract nr. KE3108/BE "Related to the development of computing resources for beam dynamics simulations relevant for the upgrade of LHC and its injectors"

[2] H. Bartosik and G. Rumolo, "Performance of space charge simulations using the High Performance Computing (HPC) cluster", CERN-ACC-Note-2017-0048 (2017)