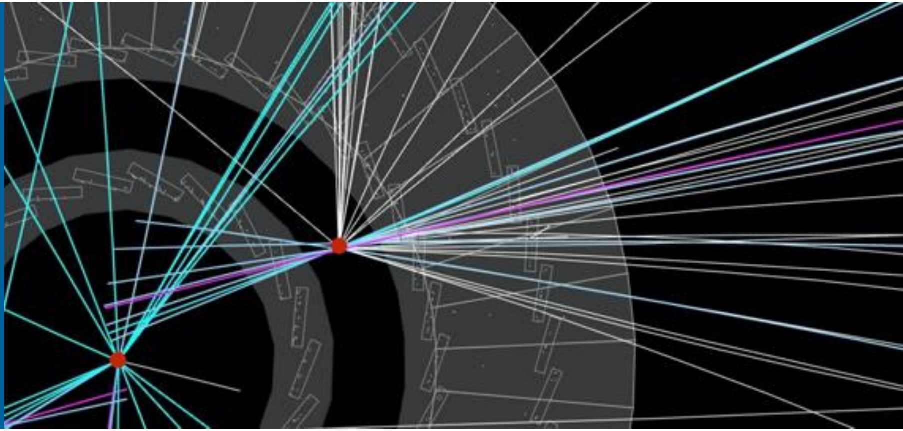
TRACK 3 - OFFLINE COMPUTING

Argonne NATIONAL LABORATORY

FRAMEWORK FOR CUSTOM EVENT SAMPLE AUGMENTATIONS FOR ATLAS ANALYSIS DATA



VAN GEMMEREN, PETER
Argonne National Laboratory

METE, ALAETTIN SERHAN

BURZYNSKI, JACKSON CARL

CATMORE, JAMES

HEINRICH, LUKAS

NOWAK, MARCIN

KRUMNACK, NILS

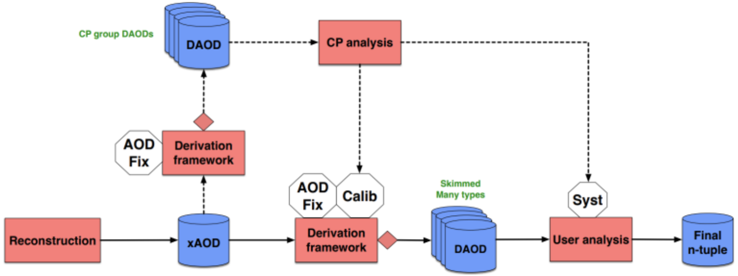
ON BEHALF OF THE ATLAS COMPUTING ACTIVITY

26th International Conference on Computing in High Energy & Nuclear Physics

RUN3 ANALYSIS MODEL, DATA AND CORE I/O

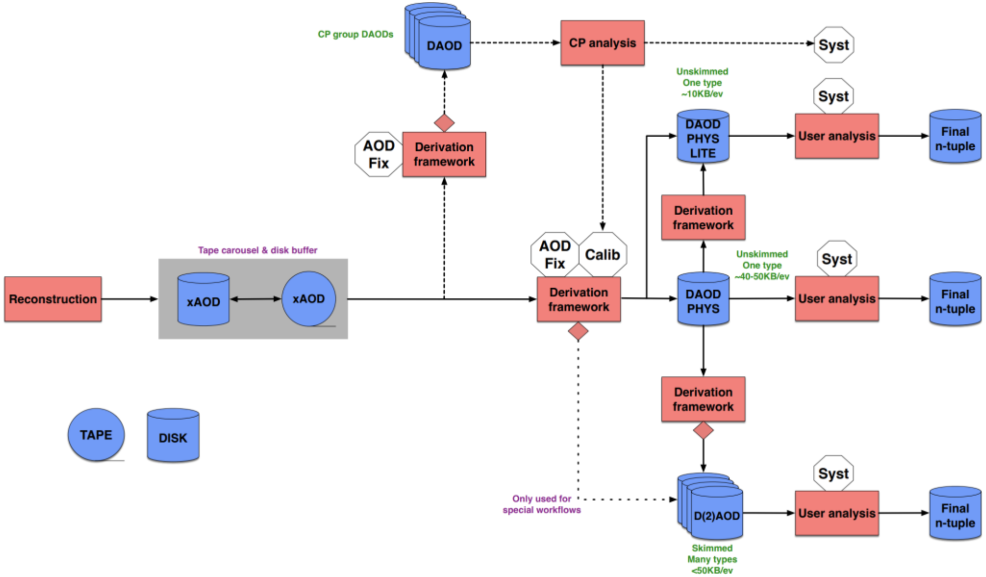
RUN2 -> RUN3 ANALYSIS MODEL

Run2: many [skimmed] DAOD streams, lots of data duplication



- Order 100 custom DAOD streams require total storage similar to primary xAOD
  - **Doesn’t scale to Run3**

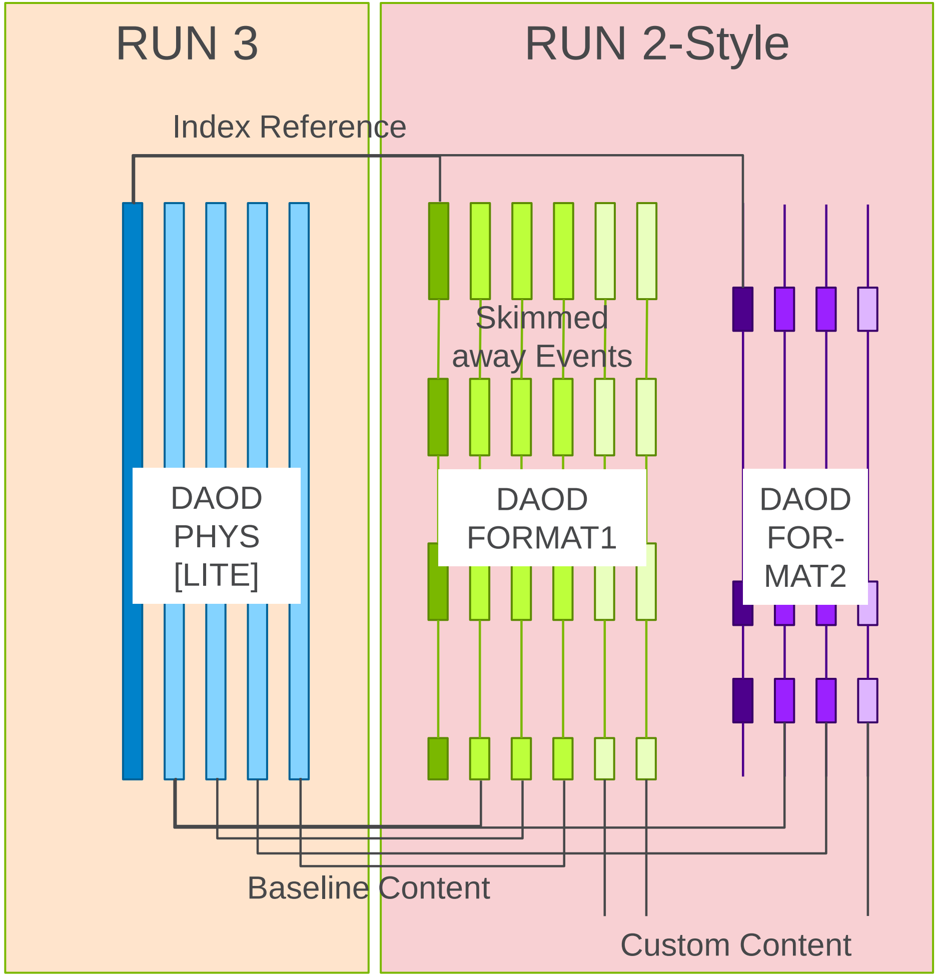
Run3: ~single [un-skimmed] DAOD_PHYS for most analysis



ANALYSIS MODEL DATA FOR RUN2 AND RUN3

For Run2, the ATLAS analysis model produced many streams of custom DAOD that contained attributes of interest for filtered events for analysis groups

In Run3, many of these custom formats are being replaced by the common DOAD-PHYS, that is written for all events.



RUN2 VS RUN3 ANALYSIS MODEL

- Introducing common formats is a big change
- Common formats produced for every event and available at ~fixed schedule
- Single analysis product saves storage (mandatory for HL-LHC)
  - Each additional column is (potentially) a big cost
    - Even if they are needed for a fraction of events only
- Baseline: analyses that cannot use PHYS/PHYSLITE* need to request own format
  - Trade skimming fraction for additional data
  - Duplicate data that already is in the common format

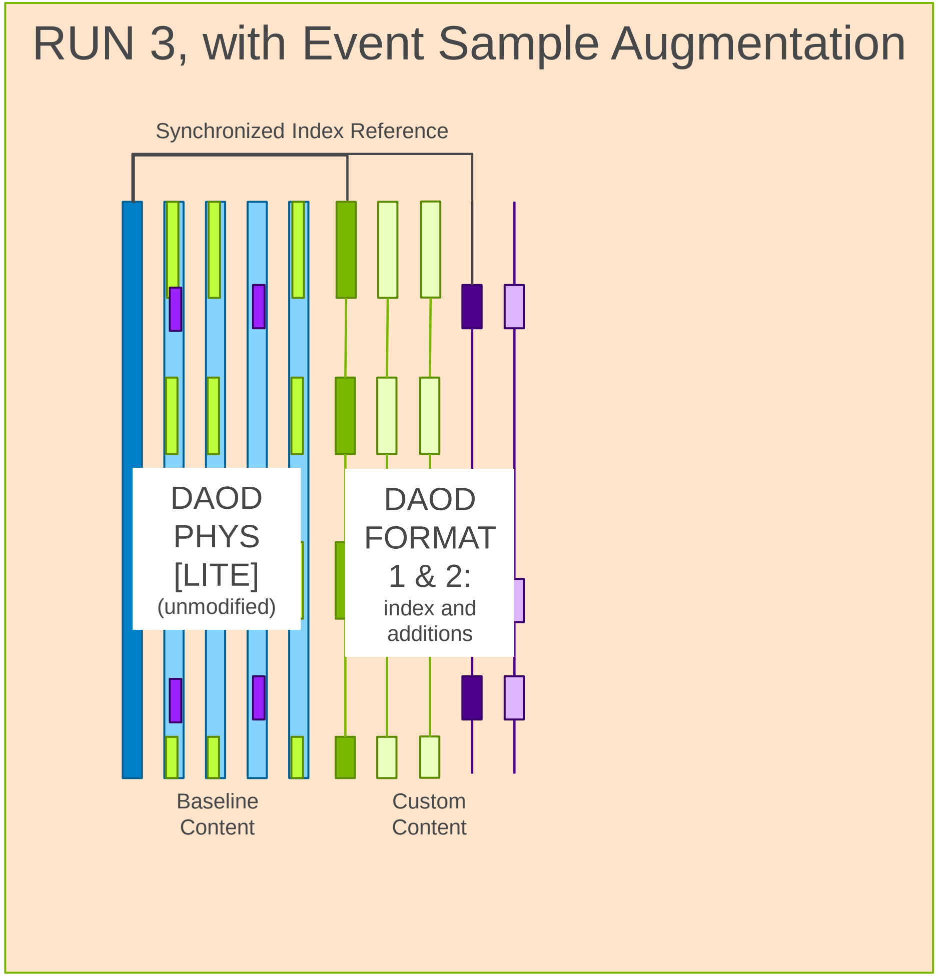
* See PHYSLITE - a new reduced common data format for ATLAS, Jana Schaarschmidt, Track 6/Marriott Ballroom IV at 2:30 PM

COMBINING RUN3 AND RUN2-STYLE DAOD

Write DAOD-PHYS, and different skims for DAOD-Format 1 and 2 into the same file.

Athena has a powerful navigational infrastructure to transparently support event-varying content and placement in different trees/files.

[1] Gemmeren, P & Malon, D. & Nowak, M.. (2014). Next-Generation Navigational Infrastructure and the ATLAS Event Store. Journal of Physics: Conference Series. 513. 10.1088/1742-6596/513/5/052036.



RUN 3: CORE I/O DEVELOPMENTS IN SUPPORT OF EVENT AUGMENTATION

Persistent object identification

- ATLAS' I/O uses navigational references that contain: Technology, DB/file id, container name and offset to access objects.
- In the past, the offset of objects stored in ROOT TTrees was given by their entry number:
  - Simple, robust and unique.
  - **Not relocatable**, if objects were moved/merged into a different tree, reference became invalid.
  - Not customizable, e.g., **can't be synchronized** among trees
- For Run3, ATLAS introduced a new column, with a unique 'index_ref', that is **relocatable** and can be **synchronized** between containers
  - Can support Event Sample Augmentation
  - Serves as ROOT index when making TTrees 'friend'.

ADDING EVENT SAMPLE AUGMENTATION TO DERIVATION

- Running Derivation to produce augmentations is simple:

```
ATHENA_CORE_NUMBER=4 Derivation_tf.py --CA --maxEvents '-1' --multiprocess 'True'
--sharedWriter 'True' --inputAODFile <file name>' --outputDAODFile 'pool.root'
--formats 'PHYS' 'LLP1' --augmentations 'LLP1:PHYS'
```

- Event filtering and additional content is defined by Long Lived Particles search, LLP1 Derivation kernel, similar to Run2
- This will produce augmented DAOD-PHYS that:
  - Can be used just like regular DAOD-PHYS
    - Unskimmed, most event data in single ROOT TTree.
  - Can read LLP1 additions for the selected event sample
    - Augmentations stored in separate, skimmed TTree that is ROOT friend to PHYS tree
    - Synchronized index_ref serves as TTree index

PHYSICS ANALYSIS HIGHLIGHT: LONG LIVED PARTICLES

LLP DERIVATION INTRODUCES 6 NEW CONTAINERS, IN ADDITION TO DAOD-PHYS

- Increase in event size ~40%, for skim of ~40% of events

- Old solution (~Run1), add 40% to all (100%) data
  - Increase for everyone
- Past solution (~Run2), write out 140% extra for 40% skim, ->56% more storage
  - Increase due to duplication
  - Everyone gets just what they want
- Event Sample Augmentation (>=Run3), write out +40% for 40% skim, -> 16% more storage
  - Smaller increase for everyone
  - Clients accessing only baseline DAOD-PHYS content will see no I/O degradation

OUTLOOK:

EVENT SAMPLE AUGMENTATION PLACEMENT IN SEPARATE FILES

OUTLOOK

Beyond same file augmentation?

- ATLAS developed infrastructure for in-file augmentation of custom event samples
  - Combines advantages of having single combined format with flexibility of (Run2-style) physics tailored multiple-stream DAOD data products
- Core software support for allowing placement in separate files is in progress
  - Increases flexibility of data placement substantially
  - Eliminates size increase of base-line/common DAOD
- Will require new data management/computational support to deliver augmentations
  - Has been challenging in the past
  - But there are new technologies which may be promising

BACKUP: ABSTRACT

FRAMEWORK FOR CUSTOM EVENT SAMPLE AUGMENTATIONS FOR ATLAS ANALYSIS DATA

For HEP event processing, data is typically stored in column-wise synchronized containers, such as most prominently ROOT's TTree, which have been used for several decades to store by now over 1 exabyte. These containers can combine row-wise association capabilities needed by most HEP event processing frameworks (e.g. Athena for ATLAS) with column-wise storage, which typically results in better compression and more efficient support for many analysis use-cases.

One disadvantage is that these containers, TTree in the HEP use-case, require to contain the same attributes for each entry/row (representing events), which can make extending the list of attributes very costly in storage, even if those are only required for a small subsample of events.

Since the initial design, the ATLAS software framework features powerful navigational infrastructure to allow storing custom data extensions for subsample of events in separate, but synchronized containers. This allows adding event augmentations to ATLAS standard data products (such as DAOD-PHYS or PHYSLITE) avoiding duplication of those core data products, while limiting their size increase. For this functionality, the framework does not rely on any associations made by the I/O technology (i.e. ROOT), however it supports TTree friends and builds the associated index to allow for analysis outside of the ATLAS framework.

A prototype based on the Long-Lived Particle search is implemented and preliminary results with this prototype will be presented. At this point, augmented data are stored within the same file as the core data. Storing them in separate files will be investigated in future, as this could provide more flexibility, e.g. certain sites may only want a subset of several augmentations or augmentations can be archived to disk once their analysis is complete.