Argonne NATIONAL LABORATORY

ATLAS Update: *I/O Developments*

Peter van Gemmeren (ANL)

ROOT I/O Workshop, #3

U.S. DEPARTMENT OF ENERGY

ATLAS use of ROOT TTreeCache

- ATLAS stores most of its data in ROOT TTrees.
  - RDO, ESD, AOD and (POOL'*esque*) DPD have a **TBranch per collection** (configurable), which are read separately on demand.
    - Several **hundreds of disk-reads**, not ordered.
  - D3PD stores **attributes** in TBranches.
    - Many **thousand branches**, but only a (small) subset is needed by each individual analysis.
- **TTreeCache** optimizes read performance of a TTree by **minimizing the number** of and **ordering disk reads**:
  - In a '**learning phase**' it discovers the TBranches that are needed.
  - After that, a read on any TBranch will also read the rest of the TBranches in the cache with a **single read**.

TTreeCache Performance

- ROOT TTreeCache can have a **huge impact** on read performance:
  - Reduce number of disk-reads by **several order of magnitudes**
    - Impact depends on system setup and use case.
      - Details from IlijaV presentations.
- However, there are restrictions in the usability of TTreeCache:
  - **Performance Reporting** not as detailed and reliable as one could wish.
    - Should improve somewhat in next releases.
  - Only **one automatic TTreeCache** per TFile.
  - **Slow learning** phase.
    - No caching while learning.

ATLAS use case for multiple TTreeCaches

- ATLAS uses **several TTrees per file**:
  - Partly due to ROOT restriction that all TBaskets need to be in sync.
    - ATLAS does have objects that do not belong into a collection and vary in number by event (e.g.: conversion helper objects).
  - Payload data and references.
  - Main event tree holds **~90% of baskets** (ESD/AOD).
- ATLAS cannot 'manually' create additional TTreeCache.
  - Athena I/O is ROOT unaware, uses ROOT via POOL (or its successor).
    - For better or worse, ATLAS has resisted making its objects inherit from TObject to avoid dependency.
- ROOT being able to use caches for all the trees, should help ATLAS to lower **disk-reads by an order of magnitude**.

TMemFile for output

- TMemFile, ROOT way of combining/merging several TTrees to the same output TTree.
- With the migration away from POOL and potentially closer to ROOT, ATLAS may decide to leverage these features directly.
- However, there still are open questions:
  - Biggest obstacle for ATLAS: Inserting TTree entry in TMemFile will not return valid entry number (afaik), so we cannot easily create an **external token**
  - Not clear whether type specific function calls could be sufficient to **merge metadata objects**
  - And what about multiple **tree synchronization**?

TMemFile and ATLAS references

- ATLAS uses **externalize-able references** for navigation.
  - Flat Data Model, DataHeader stores '**tokens**' to all container of each event:
    - Intra- and Inter- file references
    - Across different representation (e.g.: AOD to ESD)
    - Different TTree/TBranch entry numbers (event to helper object)
- Token store **file id** (POOL Guid) and **TTree/TBranch entry numbers**.
- After writing an event/entry, TMemFile does not have ***final*** **entry number**.

Plans and outlook

- TTreeCache Plans:
  - TTreeCache learning phase improvements:
    - Cache during learning
      - Useful if basket size is small compared to learning phase.
        - » ATLAS uses small 5/10 event baskets for ESD/AOD
    - Minimize effects of misuse:
      - E.G., in ATLAS analysis some use TTreeCache, but do tree->GetEntry() without telling the cache to store all branches.
- TMemFile Outlook:
  - Investigate references when using TMemFile .
    - Implement de-referencing layer in ROOT?
      - Similar to POOL's '##Sections' table