I/O and Metadata

Jack Cranshaw
Argonne National Laboratory
November 9, 2015

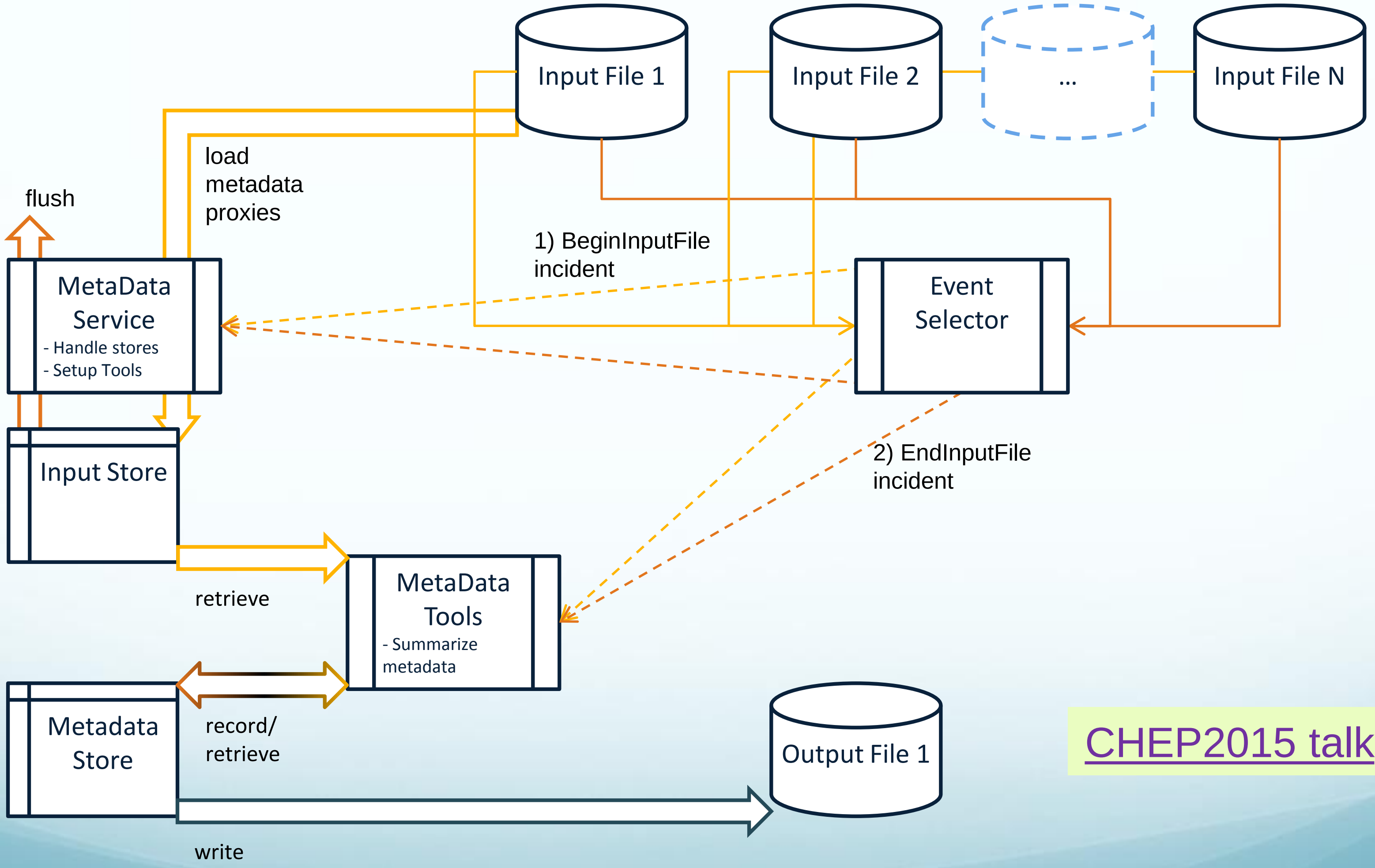
Outline

- Review
- Bookkeeping Issues
- Event Service issues
- Analysis support issues
- Merging issues
- Comments and consequences for future frameworks

Metadata Infrastructure

- Metadata is processed in athena but in addition to the framework state transitions it listens to Incidents fired by the I/O framework such as file boundaries, etc.
- This infrastructure uses Tools which listen to these Incidents and prepare output metadata in the output metadata store.
- A similar system of stores and incidents exists in some analysis frameworks allowing us to build dual-use tools.

Incident-driven metadata infrastructure



CHEP2015 talk

In-file Metadata

- Serves as a cache and a propagation tool. Stored in a MetaDataTree in the file. There is a MetaData DataHeader which sets up proxies and loads addresses.
- Metadata Itemlist
- Cache
  - IOV conditions
  - Trigger menus
  - Configuration parameters
- Propagation
  - Luminosity bookkeeping
  - Event bookkeeping

Problems with Lumi Accounting

- Updated xAOD version in production since Spring.
- After data taking started, it was noticed that all lumiblocks were being labeled as Suspect. This was fixed in the associated metadata tool in July.
- Problem of Suspect lumiblocks resurfaces in October due to AthenaMP.
  - Bug in incident firing confused tool. Fixed.
  - The tool was designed to read the number of input events from the input file. In a multi-process environment this leads to multiplication of the number during merging and consequently the lumiblocks are again labeled all as Suspect.
    - Detailed in ATEAM-235 Scope still under investigation. Can probably be fixed in tool, but points at some possible architectural issues (FF).

CutBookkeepers Issues

- There was a problem that was spotted in September where CutBookkeepers were double counting events in derivations. This was a bug in the way input streams and cycles were used, not a duplicate events issue. Fix detailed in ATLASG-242.
- Because CutBookkeepers count events in each worker, the counting problem with AthenaMP does not affect them AFAICT.
- I am migrating functionality from CutFlowSvc to a tool to both fit the Athena metadata I/O better and to have shared dual-use tools with the analysis frameworks.
- More in Analysis slide.

Metadata for Analysis

- I operate as the interface between Core and Train Coordination. Many of the issues have been metadata related.
  - Previously mentioned Bookkeeper issues.
  - New metadata needed by analysis
    - Root-readable versions of some IOV metadata
    - Simple name value pairs, standing jira ticket (ATLASG-173) for requests.
    - New uses for CutBookkeepers: PDFSumOfWeights
      - Tracked in ATLASG-308
      - Time critical for December reprocessing
      - Workaround in place, developing longer term solution, e.g. multiple cutbookkeeper tools.
- Extensibility is becoming more of an issue. Physicists like to store simple Root objects and just use hadd for merging.

Merging

- Our default for merging is full athena merge. Metadata is merged using metadata tools attached to the MetaDataSvc and listening to Incidents.

- General problem: *Metadata can be dropped during merging*.
  - Depends on metadata item list from release not from input. ATEAM-145
  - Tools may not recognize all input metadata.
  - Generally a silent error. No validation.

- Root-level (fast) merging
  - Metadata merging is not normally just a stacking process.
  - Nevertheless, in Run 1, we were able to use a hybrid merge where the event data was simply stacked using Root while the metadata was merged using athena.
    - Note: This is available for Run 2, but the event data output would not be optimized for reading in the same way as the direct athena output.
  - If we deferred the actual union of the metadata, then we could stack both event and metadata when merging, and delay the union/merge to when the data is read. Could be two stages of merge as well.
  - Should we have tools doing the merging or should the objects know how to merge themselves?
    - Tools are easier to evolve and specialize.

Pointing to Conditions Data

- As noted on the Analysis slide, physicists like to add simple metadata, histograms, etc. which is root native and could by itself be 'hadded' but does not necessarily play well with athena.

- Two differences between analysis and core show up.
  - One of them is that in analysis the events themselves can act as a file-level IOV database using pointers in the events, whereas in athena conditions we keep a separate IOVDb.
    - Both have pro's. Events are more independent with pointers. IOV means not checking every event. No planned changes.
  - Make IOV content root-readable.
    - One could think of sharing data in these frameworks if the payload in the IOV containers was root readable and addressable. Look at doing this for attribute list first.

Incidents

- We have many incidents and many classes which fire incidents.
- The IncidentSvc doesn’t report who fires, who listens, or whether an incident was missed or out of order.
- Discussion.