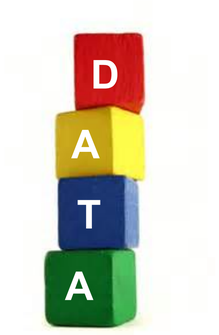
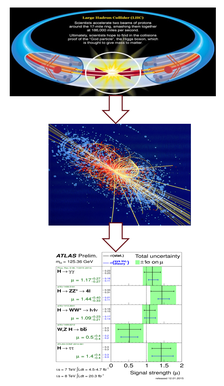
METADATA FOR FINE-GRAINED PROCESSING AT ATLAS

Jack Cranshaw (Argonne National Laboratory), David Malon (Argonne National Laboratory), Peter van Gemmeren (Argonne National Laboratory)
on behalf of the ATLAS Collaboration



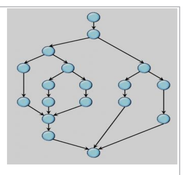
ATLAS METADATA

- Every point of data processing in ATLAS produces, extends, or propagates metadata. This metadata tracks conditions, bookkeeping, provenance, etc.
- During processing ATLAS metadata is carried along with the data for robustness across resources.
- Data volumes and computing resource developments are challenging the scalability of the current ATLAS metadata architecture.
- This contribution describes ATLAS solutions to this problem in the context of the multiprocessing framework currently in use for LHC Run2 as well as development underway for the ATLAS multithreaded framework (AthenaMT).



APPEND VS UNION

A PARALLEL WORLD



- *Metadata challenges of increased parallelism.*
  - **File Splitting**: File metadata does not fit well to a flexible event distributor.
  - **Partial Jobs**: When using opportunistic resources they may disappear suddenly and metadata must be collected to make the output usable.
  - **Global State**: In a single process, a global state such as 'all events processed' can be easily assigned.

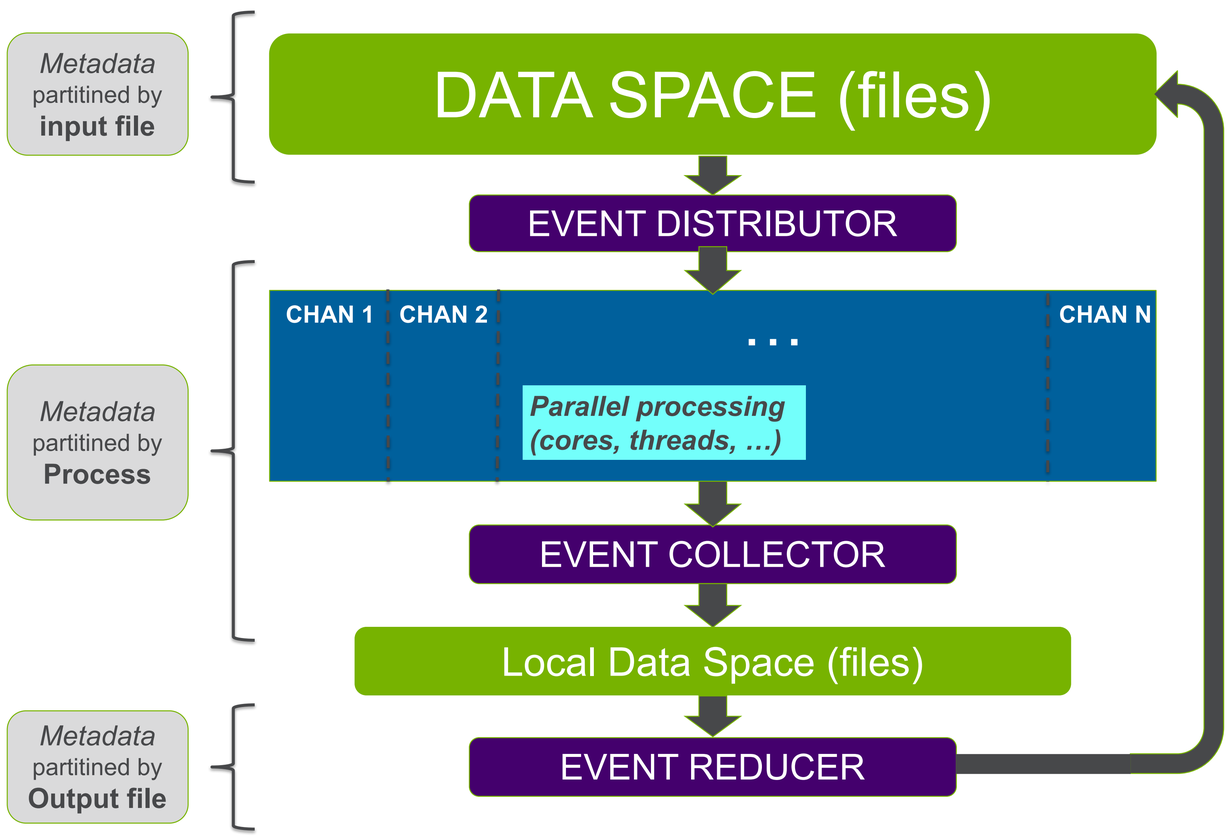
EXAMPLE 2: DEFERRING GLOBAL STATE DEFINITION

- The current ATLAS metadata model assumes that each job will define the global state of the bookkeeping for each file it uses as input.

  → *This breaks down when the processing is split among cores or threads*.
- The simplest solution seems to be to defer assigning the global state to a process downstream of the parallel execution. Since all parallel processes produce output that needs to be merged, this also fits within the current model. Two things need to be done:
  - *Dumbing down*: Subjobs index their bookkeeping by some input parameter, e.g. a file id and don't worry about global state.
  - *Smartening up*: Let the merge/reduce extend the metadata based on the input.

  → *Attempts to test this system are still under development.*

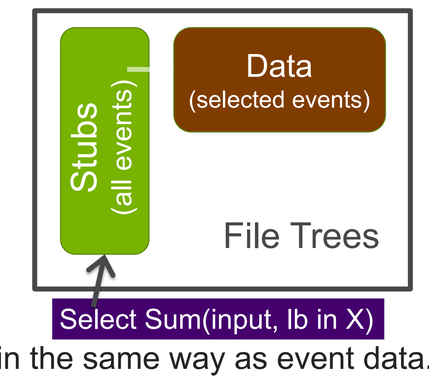
Processing Metadata in Parallel



EXAMPLE 3: DEFERRING BOOKKEEPER SUMMATION

Bookkeeping in software is like scalers on the detector. Accounting for overlaps and combining data is a problem.

→ *Why not delay the summation?*



- A simple idea: *Event Stubs*
- Use a simple struct for event id.
- Bookkeeping metadata can just be appended in the same way as event data.
- Better data integrity as the event id's can be checked for overlaps whereas simple sums cannot.

  → *A first version of this was tested over the summer at the level of writing the stubs themselves.*

EXAMPLE 1: HANDLING PARTIAL JOBS

In Run 2 ATLAS has begun deploying a new system called the Event Service which delivers blocks of events (rather than files) to a pool of processors.

- Processors might be told to stop processing and output the events they processed – along with correct metadata. *How do we handle this?*

→ By extending the 'incident' system used for metadata in the current ATLAS framework.

- In that framework, metadata transitions are triggered by 'incidents' which are transitions not triggered by the event loop.
  - The standard order for serial processing of files:
    - *Begin File, Metadata Stop, End File*
- The Event Service use case can be accommodated by extending the list of possible incident sequences, e.g.
    - *Begin File, Metadata Stop (checkpoint), Metadata Stop (checkpoint), … Metadata Stop* (process terminated).

→ *A solution along these lines is currently being tested for deployment with the event service to solve problems with event bookkeeping.*

- Further work is needed for luminosity bookkeeping and properly defining the global state of 'complete' or 'incomplete'.

UPCOMING CHALLENGES

- Most of the examples shown here are developed for the multiprocessing environment being used by ATLAS for Run 2 (2015-1018).
- Most of them will make life easier in a multi-threaded environment, but a multithreaded environment will also have new challenges.
  - Incidents being fired in one thread triggering actions in another thread. A model for doing this exists and needs to be tested with metadata use cases.
  - Moves toward multithreaded output.
    - Can we remove the synchronization point before parallel data streams are merged?
- Better and more transparent use of diverse metadata sources such as the ATLAS Metadata Interface (AMI), Panda databases, Object stores, and file resident metadata which may also provide better metadata provenance tracking.

CONCLUSIONS

- A parallel processing environment requires a more fine-grained ATLAS metadata framework, but these can build on features that already exist.
- Solutions that work for a multiprocessing environment
  - Should help with a transition to a multithreaded environment which is already well under way.
  - Have generally made the framework more robust and flexible.
- Solutions to these problems are necessary for ATLAS Run 2 to reach all of its goals and crucial for Run 3 success.

U.S. DEPARTMENT OF ENERGY
Argonne National Laboratory is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC.

Argonne NATIONAL LABORATORY