Handling of metadata in the variety of ATLAS workflows

Vakho Tsulaia (LBNL)

US ATLAS S&C Meeting
Chicago, August 2, 2016

Introduction: Purpose of this session

- New event processing workflows are becoming increasingly popular in ATLAS
  - **AthenaMP** is our mode of operation on Tier-0 and it is widely used for running a variety of production workloads on the Grid
  - The **Event Service** is being used for running Geant4 simulation at several Grid sites and at HPCs
- The Future Framework is going to be Multi-Threaded
  - **AthenaMT** is currently being actively developed in preparation for Run3
- New and emerging workflows are breaking pieces of the old model of metadata handling. Consequently, **the correct handling of metadata in these workflows has become a critical task**
- In this session:
  - Identify areas in the metadata infrastructure requiring update/redesign/rewriting
  - Share ideas and plans about short- and long-term developments

Input file peeking and output file merging

Slide by G. Stewart. In-file metadata meeting at CERN, June 13, 2016

- Input file peeking is annoyingly slow
  - Mini-athena job ran to do T/P conversion of DataHeader
    - 700MB job to get a few 100 key/value pairs
- AthFile is practically un-maintained since Sebastien left
- Improvements by Sasha used by transforms can't be used by reconstruction
  - Not enough information, a bit hacky
- We would really like to see a lightweight in-file metadata scheme that can be interrogated in <<1s with a very small standalone reader
  - Also would open up possibilities for analysis workflows
- Metadata merging should be *well defined* for each metadata item and uniformly implemented, e.g., `item1.join(item2)`
  - Should help avoid issues with merging fragility seen in the past

AthenaMP

- We rely on AthenaMP for running reconstruction at Tier-0 and for exploiting multi-core resources of the Grid

- AthenaMP has been around for a while, nevertheless we sometimes discover serious issues with it …
  - … and some of these issues are related to meta-data handling

- **Example:**
  - In late 2015 Will Buttinger discovered a **serious problem with AthenaMP in regards to the accumulation of metadata** (collection of lumi-blocks)
  - In order to come up with a fix it was necessary to involve several experts, and it took us **~1 month to implement such fix**
  - Despite of this effort, the fix looked more like a **temporary patch/hack**, while **the implementation of a proper solution was put off**

Event Service

- Specifics of the Event Service:
  - Events are being distributed to AthenaMP workers in chunks (**Event Ranges**). Each range contains either one or several events
  - Event processing output for all events in a given range is written to the disk into a **separate file**
  - Files produced by Event Service jobs are **merged later or by specialized merge jobs**
- An important consequence of this mode of operation is that throughout the job **each AthenaMP worker produces multiple output files of a given type**
- This functionality is implemented by **Output File Sequenecer** – a generic mechanism, currently used only by the Event Service
  - Allows closing ouf output files during the runtime of a job
  - Writes events and **metadata** produced so far
  - Opens new file for the remaining events

Event Service (contd.)

- Handling of event data by the Event Service is rather straightforward
  - Events never span file boundaries
  - Event Store is cleared after execute
- In-file metadata is not so simple
  - Accumulated/summarized over the runtime of the job
  - Or propagated from Input to Output file
  - Done by metadata tools, invoked by **BeginFile** and **EndFile** incidents
  - Originally designed to be writen out only at finalize, metadata store is cleared at the end of the job
  - Not all clients play nicely and some modules used to hold on pointers to the metadata store. Some of such clients already fixed
  - Output File transition was limited by policy to Input File transition
  - Meta Data Tools can prepare for possible transition during EndFile incident

Event Service (contd.)

- Geant4 Simulation is presently the only use-case for the Event Service
  - Extension of the Event Service functionality to other workflows is a fairly non-trivial task because of number of reasons
- Dealing with metadata for Simulation jobs in the Event Service was a relatively simple simple task
  - Simulatioin needs only **IOVMetaData** and **EventStreaminInfo**
  - **NO Lumi** or **EventBookkeeper**
- On the other hand proper handling of metadata for running e.g. Reconstruction in the Event Service will **require a significant effort**
  - (Re)Design, development, testing …

AthenaMT

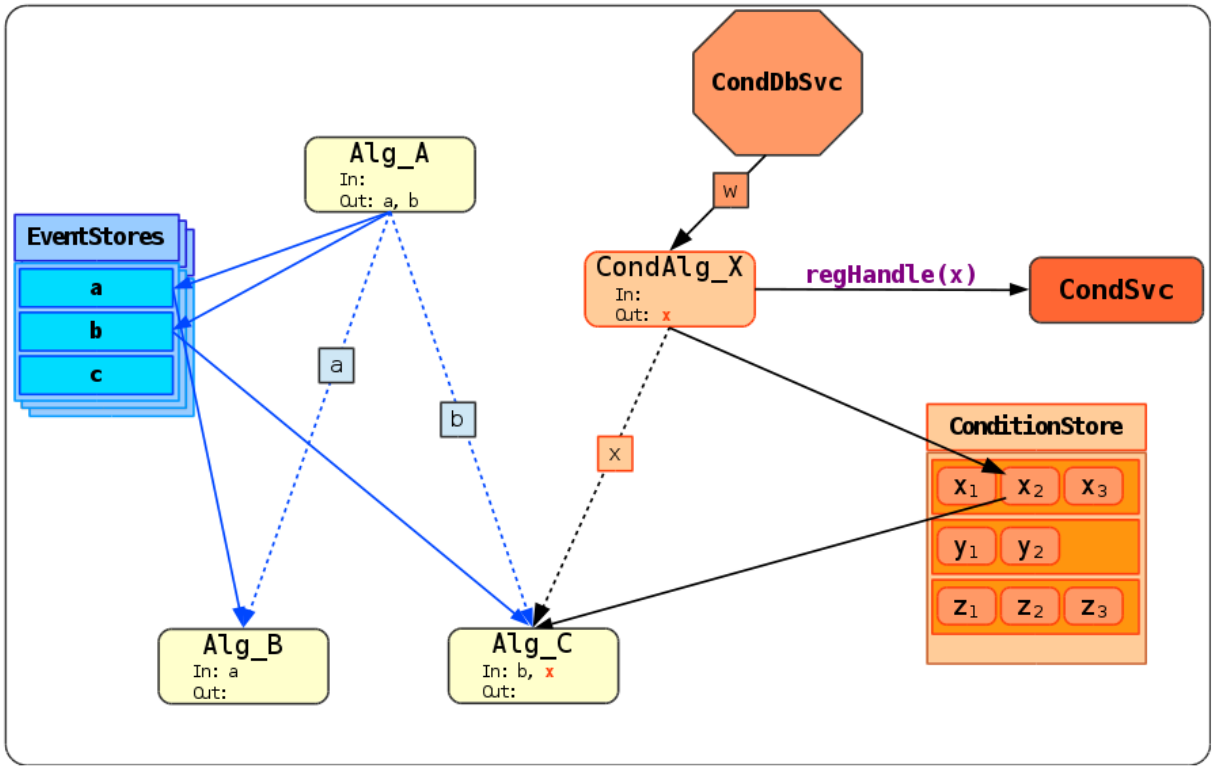
- The development of multithreaded Athena – AthenaMT – is well underway. We are making good progress in several areas
  - Reported in the ATLAS Software TIM, Glasgow, June 2016
- The transition from serial to multithreaded infrastructure requires significant changes in almost all components of the Athena framework
- Among these modifications there are some which are particularly relevant to metadata handling:
  - New infrastructure for **Conditions Data Access**
  - New infrastructure for dealing with **Incidents**

Conditions access in MT

- Conditiond Data Objects moved from Detector Store to a **Conditions Store**

- Several instances of any Conditions Data Objects will be kept in the ConditionsStore, indexed by IOV

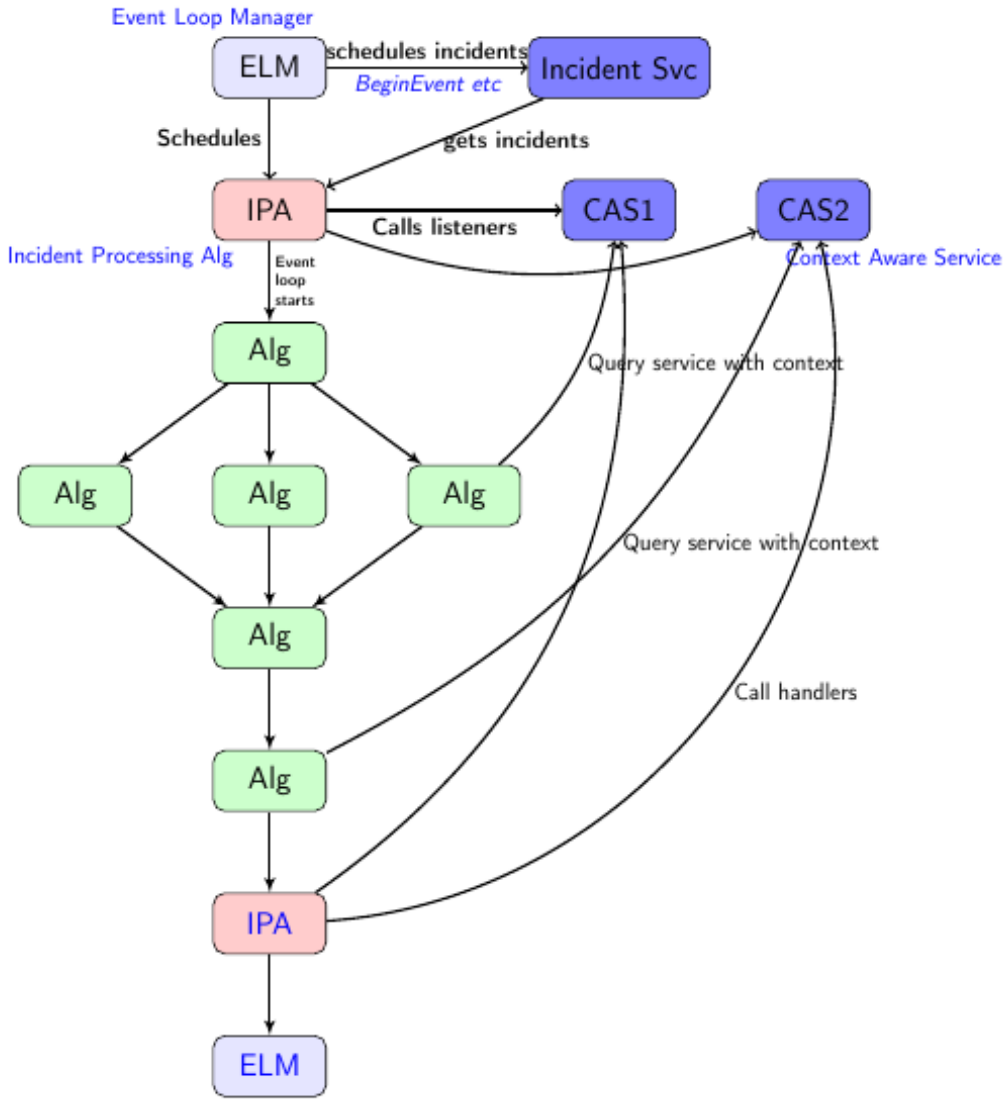
- Callback functions in Conditions Clients are being replaced with Algorithms, scheduled by the framework



Work of Charles Leggett

Incidents in MT

- The Event Loop Manager **schedules Incidents**
- Incident Processor Algorithm (IPA) is scheduled. It gets incidents and calls handlers
- Algorithms and Tools query services with context



Work of Sami Kama

Metadata in MT

- The part of the metadata infrastructure which relies on incidents will have to be modified either to use the new incident handling mechanism or not to use incidents at all

- Can metadata reuse some of the Conditions Access infrastructure?
  - At the same time we should avoid making Conditions Access system generic and sacrifice its performance

- More questions need to be answered:
  - How to implement MetaDataStore in MT?
  - Several metadata components accumulate state. How should these components function in MT? (this is a general question)
  - …

- First we need to have a clear strategy how to address these questions/issues and then we need to proceed with writing first prototypes, building new infrastructure, migrating the clients etc.

Metadata TIM in June

- Metadata work areas discussed in the Database & Metadata TIM at CERN, June 2016 (https://indico.cern.ch/event/515805)

- We agreed to follow up on the architecture and performance discussions in the Core Software meetings
  - Merging optimizations and performance numbers for reading
  - Changes required for the Event Service
  - Metadata storage for performance and ease of merging
  - Processing in a multithreaded environment, synergies with conditions
  - Revied use of incidents

- These ideas have not yet been materialized into concrete discussions mostly because of Summer travels and unavailability of several people
  - We should give this high priority and start the discussions ASAP

Summary

- Correct handling of metadata in existing, emerging and future workflows has already become a critical task
- Some of these workflows (e.g. AthenaMT) require fundamental redesigning of the way we deal with in-file metadata
  - Metadata handling in AthenaMT provides both challenges and opportunities
  - We should think about better ways to do things and not just port old hacks
- Some of the metadata work for MP/new workflows and MT solve the same problem
  - There is work which can help both Run2 and Run3. Also should help HPC
- ***Most of the developments can leverage the existing I/O infrastructure.***
- Jack will talk about some of the specific ideas and work areas