ATLAS I/O REVIEW: HPC RELATED TOPICS

ATLAS Sites Jamboree and HPC strategy:

Peter Van Gemmeren (Argonne National Laboratory (US))

Outline

- During 2018, ATLAS assembled an I/O Roadmap https://cds.cern.ch/record/2632001 directed primarily at Run 3,
  - *but with due consideration to ensuring we are going in a direction compatible with Run 4 needs*
- Many Core and SPOT meetings were dedicated to discussion and development.
- Discussed during June SW&C week @ DESY: https://indico.cern.ch/event/646942
- Input to I/O review 09/26-27: https://indico.cern.ch/event/717880/

HPC TOPICS

Processing Environments: HPC

- High Performance Computing (HPC) are different from the grid.
  - *The HPC compute nodes are interconnected with low latency high speed interconnects.*
  - *There are large shared file systems optimized for streaming read and write operations.*
    - Some systems also include local SSD storage or well connected SSD storage in the form of a Burst Buffer.
  - *Traditionally, these systems were designed to have no outside connectivity from the compute nodes.*
    - There is a trend to have systems with some outside connectivity.
    - *Major portion of compute power will be provided by accelerators (GPU, FPGA, etc).*

HPC Usage

- In order to use the HPC resources efficiently, ATLAS production workflows will need to change.
  - *In Run 2, ATLAS production workflows on HPCs were predominantly simulation.*
  - *For Run 3, some machines will run more than simulation and perhaps almost all production workflows.*
  - *By Run 4, (HL-LHC), HPC resources could be running all productions workflows: event generation, simulation, reconstruction, full/fast chain simulation and perhaps even derivation production.*
    - *Wider HPC deployment is a large ingredient to mitigating the HL-LHC processing challenge.*

HPC Usage I

- ATLAS production workflows should make use the of the inter-node communication networks using MPI libraries for shared readers and writers to avoid overloading shared file systems.
  - *In order to minimize and avoid random access to the shared file system, reads (and writes) should be organized to and process as many contiguous blocks as possible.*
    - *Currently done via ROOT TTreeCache and using AutoFlush/AutoSave settings.*
- A process outside of Athena should be used synchronize the activity between the compute nodes.
  - *This process might be ATLAS-specific.*
    - *Currently SharedWriter/Reader for AthenaMP within compute node*

HPC Usage II

- The overall system (production system, Athena including I/O) may have to scale to up to 3000 nodes each with 10's to 100's of cores to adapt to the HPC.
  - *ATLAS workflows running on the HPC resources should be structured to reduce the number of intermediate (transient) outputs.*
    - *SharedWriter and ongoing development of TMPIFile.*
- HPC batch systems predominantly use whole compute node scheduling;
  - *in order to use CPU resources efficiently ATLAS workflows targeted for HPC sites will have to adjust for this.*
  - *The I/O system needs to be tuned for whole node running not single core running as is done, for example, with merge jobs currently.*

HPC Requirements

1. Use information from the production system and Athena to configure the I/O system for the topology of the compute node ie:
    - Memory hierarchy
    - Existence (or not) of accelerators (GPGPU, FPGA, TPU etc)
    - Storage hierarchy (Ram Disk, SSD, Burst Buffer, Shared File system)
2. Efficient use of Storage hierarchy
    - Make efficient use of fast transient storage
    - Minimize random access reads/writes to the large shared file system; more streaming reads and writes
3. The system should be able to scale horizontally up to 3000 nodes each with 10's to 100's of cores to adapt to the HPC size and local workload management system
4. Metadata handling system if it does not come entirely from infile metadata

Planning:
I/O for HPC environments

High priority:

1. *Develop a means to support event references when sending event data to another process, local or remote, for writing.*
2. *Once a referencing mechanism has been developed, assess whether implementations such as ROOT TBufferMerge meet ATLAS requirements.*

Medium priority:

1. *Extend current shared writer infrastructure to operate in an MPI environment, with processors sending data to be written to an off-node writer.*

Low priority:

1. *Prototype alternatives for allowing a single reader or a small number of readers to serve input data to multiple processes.*

Progress:
I/O for HPC environments

- High priority:
  - *Develop a means to support event references when sending event data to another process, local or remote, for writing.*
    - *Introduced new APR/POOL technology (RootTreeIndex), adding new TBranch to event TTree with custom UID, which is inserted into references and can be used instead of row number*
- Medium priority:
  - *Extend current shared writer infrastructure to operate in an MPI environment, with processors sending data to be written to an off-node writer.*
    - *Development of TMPIFile for ROOT will in future allow SharedWriter like capabilities across nodes. Prototype exists and is being tested for ROOT inclusion.*

HPC RELATED TOPICS

Planning:
Event streaming services

- The following items are on the ESS R&D side. The I/O developers should continue discussions with the ESS team, although at the moment of writing this document no I/O developments were identified.
  1. *Prototype and grow server-side event selection and marshalling capabilities, beginning with chunk-size-aware event range delivery.*
     - *Provide data Storage Parameter, scheduled.*
  2. *Expand to elementary event filtering capabilities, and to server-side slimming (delivering only the needed event data objects), as the corresponding metadata capabilities are developed and extended to support such selections.*
     - *Infrastructure for decision-based event selection (also for mini AOD), in progress*

Planning: Heterogeneous computing and serialization

- This is an important strategic goal and the I/O developers should be actively involved:
  1. *Collaborate in incipient and planned projects to allow ATLAS code to exploit GPUs and other coprocessors.*
     - *Some work being done as part of the Core software group*
  2. *Contribute to the development of a strategy to stream ATLAS data for such processors, and to adapt the ATLAS EDM as needed to support such processing. Integrate such serialization developments into a coherent ATLAS approach to data streaming for both transient and persistent purposes.*
     - *Some work being done as part of the Trigger experts in collaboration with I/O.*
  3. *Develop a means to stream data efficiently from persistent storage, directly or nearly so, to processing units, with minimal conversion or reformatting.*

Outline

- ATLAS I/O review completed with extensive involvement of the ATLAS community (Thanks!) including HPC experts.
- HPC use cases/environment was considered in the planning of future I/O development.
- Some progress on HPC related priorities is being made.