PROCEEDINGS OF SCIENCE

ATLAS Event Data Organization and I/O Framework Capabilities in Support of Heterogeneous Data Access and Processing Models

David Malon,[a] Jack Cranshaw*,[a] Peter van Gemmeren,[a] and Marcin Nowak[b] on behalf of the ATLAS Collaboration†

[a]*Argonne National Laboratory*
[b]*Brookhaven National Laboratory*
E-mail: malon@anl.gov

Choices in persistent data models and data organization have significant performance ramifications for data-intensive scientific computing. In experimental high energy physics, organizing file-based event data for efficient per-attribute retrieval may improve the I/O performance of some physics analyses but hamper the performance of processing that requires full-event access. In-file data organization tuned for serial access by a single process may be less suitable for opportunistic sub-file-based processing on distributed computing resources. Unique I/O characteristics of high-performance computing platforms pose additional challenges. The ATLAS experiment at the Large Hadron Collider employs a flexible I/O framework and a suite of tools and techniques for persistent data organization to support an increasingly heterogeneous array of data access and processing models.

38th International Conference on High Energy Physics
3-10 August 2016
Chicago, USA

*Speaker.

†Argonne National Laboratory's work was supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under contract DE-AC02-06CH11357.

PoS(ICHEP2016)852

1. ATLAS Run 2 event data model

The ATLAS experiment[1] substantially redesigned its event data model for Run 2 of the Large Hadron Collider, orienting it more directly toward user analysis. Details of the Run 2 data model have been described elsewhere[2]. Among its key features are:

- a shared pattern for state definition for event data model objects;
- optimization for attribute-level retrieval and histogramming;
- column orientation, aggregating attributes across objects in a collection, with a preference for structs of vectors over vectors of structs;
- design with direct mapping of the transient data model to its persistent representation in ROOT[3] in mind;
- support for dynamic runtime addition of new attributes and decorations.

Such a design is well matched to end-user analysis use cases. There are a few limitations: event-by-event variation in content types is not supported (e.g., if one event contains a jet collection, every event must contain a jet collection, even if it is empty), and schema evolution support is deliberately limited to that provided by ROOT for the sake of simplicity and efficiency in late-stage analysis.

2. ATLAS persistent data model infrastructure

ATLAS employs a highly capable and very general persistent data model infrastructure that has been described in detail elsewhere [4] [5]. Among its features are:

- support for direct navigation to and retrieval of arbitrary data objects across supported persistence technologies;
- uniform reference model for event, sub-event and non-event data, in-file and cross-file references with runtime cross-file navigation, and runtime back navigation to upstream data;
- support for persistent state representation of arbitrarily complex transient data objects at a level above the choice of persistence technology;
- support for objects that serve as event entry points, which also record provenance, support access to upstream data, and allow restoration of the state of the transient event store independent of persistence technology.

The capabilities of this infrastructure are more extensive and more general than ATLAS has tended to exploit in practice. In Run 1, for example, while the software provided the possibility of runtime access to optional additional non-local data while processing an input dataset, such access would not have been readily supported by then-current production system components and site configurations. These capabilities, though, are well suited to a world of distributed object stores, and to an environment in which any data at any level of granularity should be readable from anywhere via wide-area access protocols.

3. Performance: tradeoffs and tuning

While the ATLAS Run 2 event data model is well suited to direct user analysis, it is less tailored to full-event processing (reconstruction, input to the derivation framework that produces ATLAS analysis data products, and so on) than to selective content retrieval. Among other considerations, a side effect of the model is substantial memory consumption for writing and reading when individual attributes have first-class status in the persistent data model. This effect is decidedly non-trivial when full events are processed, and would without optimization cause ATLAS reconstruction and other processing to exceed the available memory on many of the computing resources upon which it relies. While adding attributes and decorations as "dynamic" is appealing to users, such additions can exacerbate the problem, as each such attribute is stored in an output ROOT file in such a way that it contributes its own nonnegligible buffer and memory footprint consequences. Because of the requirement that all events have the same content objects, the value of such dynamic attributes may be diminished in practice in any case, as equivalent (albeit null) attributes must be added by back-filling or by other means to the events that lack them.

Performance tuning must balance such trade-offs. Examples of tunable parameters include buffer and (ROOT) basket sizes, commit intervals, and buffer flush settings. A program of careful measurement for a variety of use cases is required, and has been undertaken under various local and remote data access scenarios for the most important production workflows. Reordering of data within files can also help, and can be optimized for specific use cases. Efficient aggregation and de-aggregation of attributes in transient-persistent conversions (an area of ongoing work) also promises the potential for substantial savings in storage footprint.

4. Emerging workflows

Opportunistically-available resources are playing an increasingly important role in ATLAS computing. Efficient use requires a scatter-gather architecture capable of delivering one or a few events rather than full files of events to ephemerally-available resources. The ATLAS event service[6] supports such a fine- and variable-grained model. Feeding the very large numbers of processors on high-performance computing (HPC) platforms is another increasingly important use case.

I/O components have been successfully adapted to support the ATLAS event service model, with concomitant support for multi-process worker jobs [7]. There remains fertile ground for optimization, though, both in persistent data layout and in I/O components that support such processing. As the current production mapping of the ATLAS Run 2 data model to ROOT persistence aggregates objects of the same type across events, care must be taken by I/O components to avoid substantial inefficiencies. These may arise, for example, when different processes handle different events that have been compressed together in the same sets of buffers. A simple alternative persistent data model strategy to support wide-area event-by-event data distribution might store all data for a single event in a single contiguous block of bytes, making the process of sending one event to one processor and another event to another relatively straightforward. This contiguous block of bytes strategy is in fact employed in the so-called "bytestream’" files written by the trigger farm for offline reconstruction. A disadvantage to such an approach, though, is a larger storage footprint because of reduced compression (no compression across events). At LHC data volumes and given collaboration storage resource constraints, such a disadvantage may or may not be decisive. For processing that requires access only to a handful of event data attributes, there could be further disadvantages (e.g., reading unneeded data) that must be balanced with the use cases that read entire events.

These and other tradeoffs serve to emphasize the advantages of a flexible persistent data model architecture as well as a need for an ongoing program of optimization. Current ATLAS production does not adjust or tune the persistent data model differently to support event service versus more traditional workflows, or to adapt to an object store versus a more standard file system as an output destination, but it could.

5. Conclusions and next steps

Performance tuning in ATLAS is an ongoing activity, not only for I/O and persistence, but for almost all software used in large-scale event processing. The approach to I/O and persistence tuning must balance analysis and production use cases, wide area and local access, and standard and emerging workflows. Tuning choices may change as use case and workflow balances shift. Among strategies under investigation is a storage-chunk-aware event streaming service, which could distribute parcels of events matched to how events are bundled in the input file (as determined by flush settings and commit intervals and other I/O options). Among the challenges is to implement such a capability in a way that keeps framework components as technology-independent as possible so that ATLAS may benefit, when appropriate, from serialization and persistence technology developments both within and without the HEP software community.

References

[1] ATLAS Collaboration 2008, *The ATLAS Experiment at the CERN Large Hadron Collider*, *JINST 3* **S08003** [`doi:10.1088/1748-0221/3/08/S08003`].

[2] A. Buckley et al., *Implementation of the ATLAS Run 2 event data model*, *J. Phys.: Conf. Ser.* **664** (2015) no.7, 072045 [`doi:10.1088/1742-6596/664/7/072045`].

[3] R. Brun et al., *ROOT: An object oriented data analysis framework*, *Nucl.Instrum.Meth.* **A389** (1997) 81-86

[4] P. van Gemmeren et al., *Next-Generation Navigational Infrastructure and the ATLAS Event Store*, J. Phys. Conf. Ser. **513**, 052036 (2014) [`doi:10.1088/1742-6596/513/5/052036`].

[5] P. van Gemmeren and D. Malon, *Persistent Data Layout and Infrastructure for Efficient Selective Retrieval of Event Data in ATLAS*, Proceedings of the DPF-2011 Conference, Providence, RI, August 8-13, 2011 [`physics.data-an/1109.3119`].

[6] P. Calafiura et al., *The ATLAS Event Service: A new approach to event processing*, J. Phys. Conf. Ser. **664**, no. 6, 062065 (2015). [`doi:10.1088/1742-6596/664/6/062065`].

[7] P. van Gemmeren et al., *I/O strategies for multicore processing in ATLAS*, J. Phys. Conf. Ser. **396**, 022054 (2012), [`doi:10.1088/1742-6596/396/2/022054`].