An extensible infrastructure for querying and mining event-level metadata in ATLAS

D Malon¹, J Cranshaw¹, Q Zhang¹ for the ATLAS collaboration

¹Argonne National Laboratory, Argonne, Illinois 60439, USA

E-mail: malon@anl.gov, cranshaw@anl.gov, qzhang@anl.gov

Abstract. The ATLAS event-level metadata infrastructure supports applications that range from data quality monitoring, anomaly detection, and fast physics monitoring to event-level selection and navigation to file-resident event data at any processing stage, from raw through analysis object data, in globally distributed analysis. A central component of the infrastructure is a distributed TAG database, which contains event-level metadata records for all ATLAS events, real and simulated. This resource offers a unique global view of ATLAS data, and provides an opportunity, not only for stream-style mining of event data, but also for an examination of data across streams, across runs, and across (re)processings. The TAG database serves as a natural locus for run-level and processing-level integrity checks, for investigations of event duplication and other issues in the trigger and offline systems, for questions about stream overlap, for queries about interesting but out-of-stream events, for statistics, and more. In early ATLAS running, such database queries were largely ad hoc, and were handled manually. In this paper, we describe an extensible infrastructure for addressing these and other use cases during upload and post-upload processing, and discuss some of the uses to which this infrastructure has been applied.

1. Introduction

The ATLAS experiment at the Large Hadron Collider (LHC) writes event-level metadata records (event TAGs) for each real and simulated event. TAGs are event thumbnails, containing information identifying the event, details about which triggers the event has passed and at which levels, key physics quantities describing the event and the characteristics of its constituents (particles, jets, and so on), and, importantly, sufficient information to allow direct navigation to file-based event data at any upstream processing stage [1] (RAW, Event Summary Data (ESD), Analysis Object Data (AOD), ..., and for simulation, any stages upstream of ESD, including event generation, hits, digits, and simulated RAW data in either a bytestream (detector readout-like) or an object format known as RDO). TAGs are created when AOD are created and merged, deriving their content from the information therein, and written first into file-based datasets as part of standard ATLAS production processing. These file-based datasets are uploaded into Oracle databases to provide an integrated global view of ATLAS data. The chief Oracle instance is at CERN, but data are distributed and replicated to other sites as well. Reasons for this include capacity (CERN is not provisioned to host all TAGs from simulation in addition to all TAGs from collision data), performance and load balancing (some queries on TAGs can be time-consuming given the volume of ATLAS data, particularly if they require full table scans), robustness, and fail-over [2]. Users, however, are presented with a unified global view of the data,
with details of where data are in fact hosted and where queries may run left to the implementation. Details of ATLAS TAGs and TAG infrastructure have appeared in several papers in recent years, and will not be further discussed here. See, for example, references [3], [4], and [5] for details.

The focus in this paper is on the TAG database in particular, and upon the capabilities it provides to the collaboration in the areas of monitoring both for data quality and for physics, for error and anomaly detection, for discovery of correlations, and for statistics that could not otherwise be easily obtained by stream-by-stream analysis. Because it provides a global view of the data, the TAG database is a natural locus for detection and collection of cross-stream and out-of-stream, cross-run, and cross-processing information. The TAG database is also the data repository behind ATLAS event picking, and it is routinely used for event selection, as it contains sufficient information for direct navigation to file-based event data of interest at every processing stage through AOD. Such important uses, as well as a variety of services related to these capabilities, including automatic TAG-based skimming, have been described in [6] and elsewhere.

ATLAS has been using the TAGs for these and other purposes even before proton-proton collisions, for example, to understand detector commissioning data (and before LHC collisions became routine, TAG content was even used in some cases to deduce that collisions had occurred), but as the experiment and its infrastructure matures, a variety of checks and tests and statistics are increasingly being run and gathered routinely and automatically, and checks that were once run manually by experts are increasingly being made available for direct invocation by interested users. These uses and the machinery that supports them are the principal subject of this paper.

In section 2, we describe some of the specific purposes to which TAG-based monitoring and statistics gathering has been and will be applied. In section 3 we describe the infrastructure that implements this capability. In section 4 we provide conclusions and indicate directions for future work.

2. TAG-based monitoring, integrity checks, and statistics
While the range of current and potential uses for TAGs is extensive, many current monitoring uses fall into one of three categories:
• data integrity;
• rare events and anomalies;
• statistics and stability.

These categories are illustrative, but they are not exhaustive. Examples in each category follow.

2.1. Data integrity
Data integrity issues may arise for a variety of reasons. Here are a few examples.
• Errors occasionally arise in data flow and processing within the ATLAS trigger system. Not all of these are fatal, and events survive in spite of certain errors, such as truncated readout buffers, and in the presence of certain warnings, such as bunch crossing id check failures. Such events are marked internally, and the TAG database provides a means to routinely check and monitor the corresponding error bits, both to identify such events for further followup and as an early detection system when error or warning rates increase.
• It is rare but not impossible that multiple copies of an event may be processed by the ATLAS high-level trigger. This can happen, for example, if in response to a timeout an event is sent to the high-level trigger a second time, and both the original and the retried versions survive. It is likely that such events would be written to different files, and detecting such event duplication is difficult offline. Discovery of such duplication is a natural role for the TAG database, where one can check for uniqueness and even impose uniqueness constraints.
• ATLAS events may be written to multiple streams. This may happen, for example, when an event passes multiple triggers associated with more than one stream. In a multi-stream analysis, however, it is important not to process the same events twice. While it is not terribly difficult to eliminate such duplication at the analysis stage, the TAG database can both routinely measure stream overlap and identify events in another stream that should be skipped, or processed. Apart from duplicate detection and elimination, though, the TAG database provides another integrity check capability: a quick means to compare whether the same
event written to two streams and therefore reconstructed independently twice contains the same physics values in its TAGs.

- Analyses that involve luminosity or cross-section information must in general process all qualifying events in a contiguous run segment that ATLAS calls a luminosity block. If luminosity blocks are not integrally contained within a single file, analyses become quite sensitive to file loss, since loss of a single file or failure to access it may lead either to incorrect results or to the need to exclude data from reachable files. Some luminosity block splitting is expected and accounted for—for example, raw data from a single luminosity block are written to multiple files by the multiple writer processes associated with the ATLAS event filter. When splitting is inadvertent, as happened at one point in 2011 because of job configuration errors, the TAG database can detect this, and was in fact used specifically for the purpose of determining the extent of the splitting and the data files implicated after this incident. Since that time, the TAG database has provided a means to monitor such splitting and to alert experts to any recurrence.

2.2. Rare events and anomalies
The global view provided by the TAG database allows it to be used to identify unusual properties in the data that are difficult to detect when processing a specific selection of events in sequence. Following are a few examples in this category.

- There may be out-of-stream events of interest to a given analysis. For example, there may be events with electrons that are found by offline reconstruction, though they did not pass the expected electron triggers, and such events may appear in streams other than the one to which electron-trigger events were written.
- There may be anomalous or outlying values in any of the event attributes recorded in the TAG—anomalous either within the given stream or with respect to the data more generally.
- ATLAS provides for various debug streams and other monitors. For many runs these debug streams are empty, but the TAG database provides a means to check the properties of debug stream events, to see whether they should in principle be relevant to an analysis, and also to assess whether there might be a potential physics bias to events with processing characteristics that cause them to be routed to the debug stream.

Rare event and anomaly detection are flavors of data mining, for which event thumbnails are ideally suited [7]. Such uses are expected to grow over time with increased data and increased understanding of that data.

2.3. Statistics and stability
Because the TAG database sees all events for a run or a set of runs, it is also well placed to monitor changes to statistical characteristics of the data. Here are a few examples that we are implementing.

- Check the frequency of occurrence, grouping, or changes over time for errors detected upstream and marked in the TAG data.
- Check the level of and changes to stream overlap, as well as whether stream A events that are marked as also going to stream B are actually present in stream B.
- Monitor changes in physics quantities stored in the TAG data. Maxima, minima, and means of a variety of physics quantities are tracked. This can provide a quick way to spot problems in reconstruction or other processing issues.
- Look for differences between data processed using different versions of ATLAS software or in different processing campaigns.

3. An extensible query infrastructure
Even prior to the work described in this paper, ATLAS had developed a number of post-upload procedures for TAG databases, principally for reasons of operational robustness and database integrity [8]. The focus was principally operational, but some limited data integrity checks were part of the processing as well. It was important, for example, to know whether all data from a given run and stream had been uploaded, and in-stream event duplication was detected as part of uniqueness and key constraints on tables.
TAG upload is a Tier 0 process, handled in production like any other Tier 0 task. It is invoked after file-based TAG datasets are created (as part of AOD merging, as described in the introduction), or alternatively after file-based TAG datasets from remote processing—either from simulation or from distributed reprocessing of collision data—arrive at CERN by subscription. When the ATLAS Tier 0 begins a TAG upload task, a follow-on task is defined to handle post-upload processing. It runs after all jobs from the upload task have finished successfully, i.e., when all event TAGs from a given run and stream have been uploaded. Post-upload processing has included indexing and index management, partitioning, buffer management, dictionary handling, statistics, access control for users and services, table splitting, and insertion of data distribution information into a catalog database.

A growing list of checks implemented in PL/SQL is being developed and deployed—see the previous section for examples—and post-upload processing provides a natural locus for incorporation and automatic invocation of such checks, as well as for other and monitoring and statistics gathering. The initial emphasis has been upon checks of problems that have arisen in earlier processing (split luminosity blocks, event duplication, …) and upon checks of data quality coming out of the trigger. In support of the latter, a CoreFlags word in the TAG contains bit flags that are set under a variety of error or warning conditions (duplicate event candidates, timeouts, bunch crossing id check failures, and several more), and a service to check and decode the bits in this word in particular is in place, though the infrastructure is general enough to check any word in the TAG. A growing list of statistics, including maxima and minima and means of physics quantities are now also routinely calculated. Cross-stream checks are also part of the current suite, including measures of stream overlap and associated integrity checks, (optional) tests of equivalence of events that occur in multiple streams, and certain debug stream tests.

In-stream checks and statistics are readily handled by an architecture that allows production post-upload scripts to invoke a sequence of PL/SQL procedures simply as further steps in standard post-upload processing. Cross-stream checks can be triggered in much the same way, though they require in addition a means to detect which other streams have been uploaded, and to run the cross-stream checks against exactly those, as each stream from a given run is uploaded in sequence. The implementation is extensible as well: in effect, a variable-length array of procedures is invoked, with the list of PL/SQL tests taken from the current content of a package in the ATLAS SVN repository.

These tests and checks can be invoked directly, and not simply embedded in automatic procedures invoked during post-upload processing, and they can be tailored to check specific words and specific potential error conditions of interest. Interfaces to allow their invocation either from a command line or from a simple web interface are under development.

4. Conclusions and future directions
A TAG database is a unique resource to the ATLAS collaboration that provides capabilities in support of an integrated global view of ATLAS event data, across runs, across streams, and across processings, for both collision and simulation events. As such, it has proven its worth, not only for event selection, but for monitoring, for integrity checking, for statistics, and for cross-stream and cross-run information that could not easily be obtained in any other way. The collaboration has taken advantage of these capabilities since its earliest data taking. These capabilities are increasing being integrated into an extensible system of automatic monitoring and checking and statistics gathering that should prove ever more useful as data volumes increase. The suite of checks performed, information monitored, and statistics gathered continues to grow, and much work remains, on automatic notification when error conditions or significant anomalies are discovered, and more generally, on how best to effectively convey such information to appropriate ATLAS experts.

Acknowledgments
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.

References
[1] Jones R and Barberis D 2008 The ATLAS computing model J. Phys.: Conf. Ser. 119 072020
[2] Viegas F, Malon D, Cranshaw J, Dimitrov G, Nowak M, Nairz A, Goossens L, Gallas E, Gamboa C, Wong A, and Vinek E 2010 The ATLAS TAGS database distribution and management: operational challenges of a multi-terabyte distributed database J. Phys.: Conf. Ser. 219 072058

[3] Malon D, Cranshaw J, Karr K, Hrivnac J, and Schaffer A 2006 A flexible, distributed event level metadata system for ATLAS Proc. Computing in High Energy and Nuclear Physics 2006 (Mumbai: Macmillan) pp 316-9

[4] Cranshaw J, Cuhadar-Donszelmann T, Gallas E, Hrivnac J, Kenyon M, McGlone H, Malon D, Mambelli M, Nowak M, Viegas F, Vinek E, and Zhang Q 2010 Event selection services in ATLAS J. Phys.: Conf. Ser. 219 042007

[5] Cranshaw J, Goosens L, Malon D, McGlone H, Tique Aires and Viegas F 2008 Building a scalable event-level metadata service for ATLAS J. Phys.: Conf. Ser. 119 072012

[6] Malon D, Cranshaw J, Van Gemmeren P, and Schaffer A 2006 Sailing the petabyte sea: navigational infrastructure in the ATLAS event store Proc. Computing in High Energy and Nuclear Physics 2006 (Mumbai: Macmillan) pp 312-5

[7] Van Gemmeren P, Malon D 2010 Event metadata records as a testbed for scalable data mining J. Phys.: Conf. Ser. 219 042057

[8] Zhang Q 2010 Engineering the ATLAS TAG browser J. Phys.: Conf. Ser. 331 042046