

The ATLAS TAGS Database distribution and management - Operational challenges of a multi-terabyte distributed database

F Viegas¹, D Malon², J Cranshaw², G Dimitrov³, M Nowak⁴, A Nairz¹, L Goossens¹, E Gallas⁵, C Gamboa⁴, A Wong⁶ and E Vinek⁷

¹CERN, CH-1211 Genève 23, Switzerland

²Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439, USA

³DESY, D-22603 Hamburg, Germany

⁴Brookhaven National Laboratory, P.O. Box 5000 Upton, NY 11973-5000, USA

⁵University of Oxford, The Denys Wilkinson Building, Keble Road, Oxford OX1 3RH, United Kingdom

⁶Triumf, 4004 Wesbrook Mall, Vancouver, BC, V6T 2A3, Canada

⁷University of Vienna, Dr.-Karl-Lueger-Ring 1, 1010 Vienna, Austria

Abstract. The TAG files store summary event quantities that allow a quick selection of interesting events. This data will be produced at a nominal rate of 200 Hz, and is uploaded into a relational database for access from websites and other tools. The estimated database volume is 6TB per year, making it the largest application running on the ATLAS relational databases, at CERN and at other voluntary sites. The sheer volume and high rate of production makes this application a challenge to data and resource management, in many aspects. This paper will focus on the operational challenges of this system. These include: uploading the data from files to the CERN's and remote sites' databases; distributing the TAG metadata that is essential to guide the user through event selection; controlling resource usage of the database, from the user query load to the strategy of cleaning and archiving of old TAG data.

1. Introduction

The TAG files hold event level metadata in a POOL file format[5], and are produced at the same time and rate as the AOD production. Additionally at the Tier-1 sites, further reprocessing of the same data will create extra versions of TAG files. The TAG data is the only event data that is uploaded in a relational database, and its rate of production is detailed in the ATLAS Computing Model [1]. When this data is loaded in an Oracle relational database, it occupies up to 3kB per event, making the rough calculation of storage in a nominal year of $2.0\text{E}+09$ events to 6 TB per year per pass. With the goal of keeping the 2 latest passes of data, this makes the TAG relational database the biggest consumer of storage and computing resources in ATLAS.

In this paper we present the challenge of dealing with such a large amount of data, from the TAG POOL file format until a usable relational version of this data can be presented and queried efficiently by a large number of users.

2. The challenge

The challenge presented in making data available to the end users in a manageable way has many aspects, namely:

- Uploading the data into a relational format, at CERN and remote sites, must keep up with the production rate.
- Access to this data can be done almost in a “free form” SQL, no restrictions or patterns, other than a selection using Run Number and Physics Stream, which means that all the columns in the database have to be indexed.
- Surges in data input can quickly impair the hardware resources of the relational database, careful resource usage control must be exercised.
- The potential total volume of data is large. An archiving policy is being devised to manage disused data, not only because resources are limited but also to improve performance.

This paper describes the work completed, under way and planned to tackle these aspects of the TAGS relational data and meet the demands of the ATLAS experiment physicists, in terms of fast availability, efficient querying and usability of the TAG database event metadata.

3. Uploading and distributing the TAG database

In the ATLAS Computing Model, the final stage of processing produces the "TAG" files (containing the Event Metadata) from the content of the ESD and AOD files [3].

TAGS are stored in CASTOR (CERN Advanced STORAGE manager) [2]. Event Metadata will be uploaded into the database relational collections by a process at CERN that controls Tier-0 operations [4]. The uploading of the collections is done using the POOL (Pool Of persistent Objects for LHC) [5] Collection utilities.

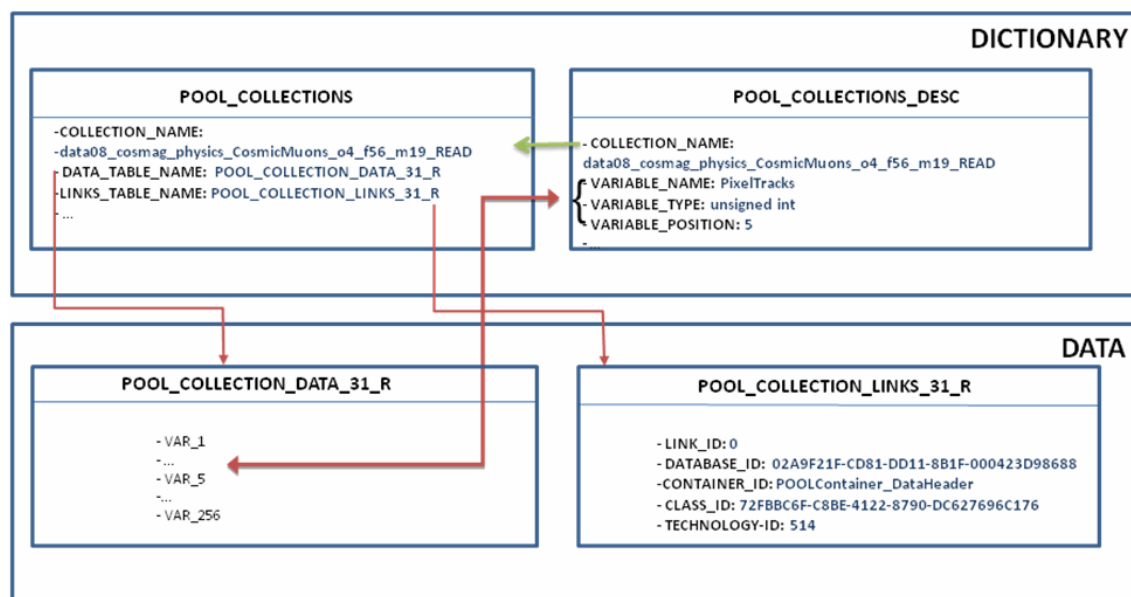


Figure 1. POOL Relational Collection Model

3.1. Format of TAG data in the database: the POOL Relational Collection model

The POOL Relational Collection model is very simple and flexible. It consists, in terms of schema, of a dictionary part, with two tables that map the collections to tables and attributes to columns, and the actual DATA and LINKS table that hold the uploaded event metadata.

This model leaves complete freedom in the naming of the collections, and the naming of the attributes, as these are derived from the ROOT [6] file structure of the TAG files, and will keep the same information, should the collection be extracted to a TAG ROOT file again.

3.2. Upload and distribution of TAG data using the Tier-0 System

The TAG data contained in the datasets produced after processing are uploaded into Oracle using the Tier-0 Job System. Each site will have a task defined symmetrically, and a python transform contained in the ATLAS Release will call the POOL Collection Utilities to translate ROOT Collections into Relational Collections, as described above. TAG datasets that are produced remotely at Tier-1 sites will be transferred to CERN and uploaded centrally to make the process more centralized and easier to monitor.

The Tier-0 System has the workflow information necessary to signal that the last file for a given run has been uploaded, and will transmit that signal to the database. This will make the data consistent as a unit, and will trigger the process of indexing and getting the data ready to query.

This upload process has been tested on different sets of data and releases, and has been evolving to become compatible with the required 200Hz rate, scalable and manageable.

3.3. Run /Luminosity metadata

Additional metadata at a level of granularity higher than that of individual events is also needed if event-level selections via the TAG database are to be useful—one may need detector status information, for example, to restrict the domain from which events are selected. Such information is being extracted from COOL, which ATLAS uses for conditions, calibrations, detector status, and quality information, and is put into an Oracle schema. This schema will have to be replicated into the databases that hold TAG data. Some of the voluntary TAG database sites will be at the Tier-2 level, which means that the Oracle Streams technology cannot be used. This is due to the fact that streaming data from CERN is only supported to the 10 ATLAS Tier-1s. However, as this metadata has a small size, a standard nightly export of the data should be enough for replication. Materialized views are another mechanism which may be useful to replicate and enhance the performance of this conditions metadata.

4. Getting TAG data ready for querying

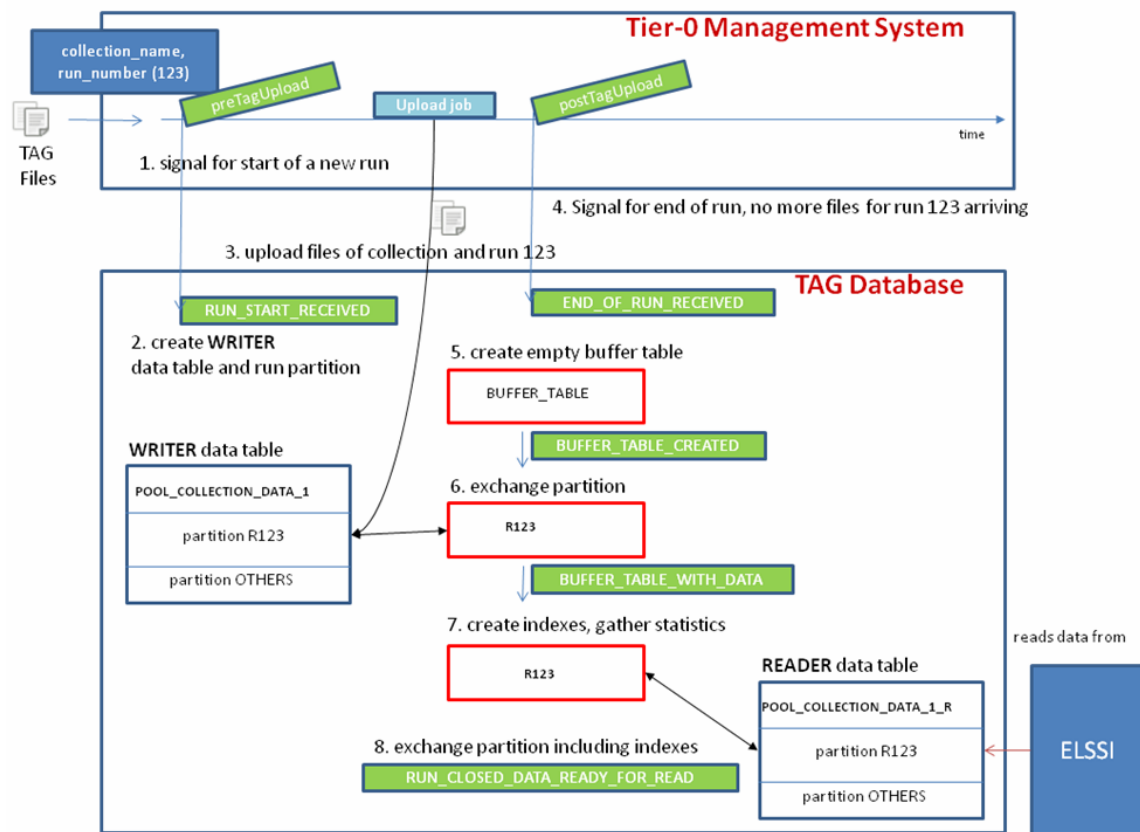


Figure 2. Workflow of TAG data from upload to querying

4.1. Oracle-Specific extensions to the POOL Collection model

The POOL Collection model is built to be persistent-technology independent. As such it can store data in ROOT files or in Relational Databases, and using the CORAL framework, keeps its autonomy from the database technology used.

The price of this independence is that it cannot use technology specific improvements and features designed for scalability such as partitioning and bitmap indexing options. To get around this limitation, the developers of the POOL Relational Collection Model put in place a call to database stored procedure for the Oracle specific case, so that before the collection is filled, the tables are partitioned by Run Number, the “transactional unit” in the ATLAS Computing model. This procedure is called POOL_COLLECTION_INIT, written in PL/SQL and its code is available in the ATLAS CVS repository [7].

4.2. The all-column indexing approach

For the querying of such a massive amount of data to be effective, tests performed using mock data [8] showed that indexes must absolutely be used for filtering the events and getting statistics out of the results. Since the user may query by any attribute, so all columns must be indexed.

Indexing creates a very large overhead during data uploading, and tests determined that it was not possible with current hardware to keep the tables indexed while uploading the data, and still achieving the 200Hz requirement. Also, some indexing options, such as bitmap, were out of the question for parallelization of uploads.

To overcome these constraints, a model was put in place that let the data be uploaded in one table first, and then, on a signal from the Tier-0 system, that the last file for a given run has been uploaded, to index all the columns and put the data to a read-only table.

4.3. Naming conventions as further partitioning of the database

The data to be uploaded is partitioned on run number at the relational table level. However the data is further broken down by the upstream production of TAG files, by physics streams. This narrows down the size of the collections making the tables smaller. Also, the naming conventions that come from the dataset name are used to separate the data into dataset name, release tags pairs, that makes it look like a large set of collections for different natures of data. This is also very useful to make queries on the database fall on smaller tables, instead of relying on huge collections that are difficult to manage.

4.4. The Writer and Reader collection workflow

The Tier-0 system is able to submit a procedure call that signals the arrival of the first TAG file for a given run, and the arrival of the last file of that run. These two events are used for signaling when the data is ready to be indexed and put into a reading form for the physics user to query.

What happens in practice (shown in figure 2):

- When the start signal is given, the writing collection is prepared, a new partition is created and the upload can proceed from there.
- When the end of run signal is given, the full partition is then put into a buffer table, fully indexed, statistics gathered, and the table is then converted into a partition on the reader collection. When this procedure is finished, the whole run is available for reading, and the tools that access that collection will see the data.

These procedures are coded in PL/SQL inside the database and put into the ATLAS Offline Software CVS Repository [7].

4.5. The ELSSI website querying workflow

Physicists are encouraged to use a website called Event Level Selection Service Interface, as their main portal into selecting events.

This website is optimized for accessing the database, uses the Run/Lumiblock metadata and puts together a whole user-friendly workflow, that leads the physicist from an overview of data into an actual Extraction and Skimming service to get the AOD or ESD from the user's selection trigger and/or physics attribute criteria. The working model starts with selecting collections and restricting run numbers, counting events with applying filters and when all conditions are satisfied, and the number of events is reasonable, to extract the result into a TAG ROOT file, accessible from AFS[9].

Using this model of access, together with the fully indexed partitioned collections, we can guarantee a scalable and manageable system that can use wisely the database resources to many Terabytes of data.

5. Controlling database usage

5.1. Tier-0 monitoring

Status and progress of TAG uploading at the Tier-0 can be monitored from its comprehensive web-based shifters' interface [10]. This both gives an overview of the overall situation, in terms of global overview plots, and a detailed view at task and job level for all the individual TAG upload tasks (to CERN and external sites) running at the Tier-0, as mentioned in section 2.2.

It is among the tasks of the Tier-0 shifter to watch the progress of TAG uploading and do troubleshooting in case of problems. With the Tier-0 and the database dashboard monitoring (described in the next section, 4.2) he/she has all information at hand for diagnosing the problems and taking appropriate actions, which range from notifying (Tier-0, ATLAS Database, IT) experts-on-call via pausing/resuming affected Tier-0 upload tasks to submitting bug reports on individual job failures through a dedicated Savannah portal [11].

5.2. Relational Collection monitoring

In order to allow shifters and database experts to monitor the upload not only from the Tier-0 process point of view as described in the above paragraph, but also regarding database procedures, a relational collection monitoring has been put in place. The dashboard [10] gives the shifter an overview of the status of TAG uploads at the participating sites.

Various scenarios have been identified as being problematic, such as the existence of database locks, buffer tables, non-partitioned tables or unusable indexes. If any of those exist, the database procedures described in 3.4 may have failed and investigation is needed. Additionally, a drill-down for each site allows the follow-up of the upload status for each collection, with the possibility to display run details and log files indicating on which part of the workflow depicted in figure 2 a problem has occurred. A related detailed troubleshooting guide is also available to the shifters.

5.3. Monitoring Oracle database performance and scalability

The Oracle databases that are used for uploading TAG data, at CERN and remote sites, are generally included in the Oracle Enterprise Manager Grid Control monitoring system [12], and so have the same level of service and overview as all of the production databases in ATLAS. This tool, together with the systems administrations tools from the various sites, such as Ganglia [13] and Nagios [14], alert and give info about the resource usage of the databases.

At present, in the Tier-1 sites that have TAG databases, such as BNL and TRIUMF, and also CERN, use the same cluster that hosts the Conditions Data, so some competition for resources may occur in these sites. The option of separating the TAG database into its own cluster is available at the sites, and will be put in action if needed.

5.3.1. Throttling upload jobs

Job submission at the Tier-0 in general takes place in a throttled, but still configurable and adjustable, manner. This is necessary for controlling the load on the various external Tier-0 clients, like the batch system, CASTOR, and several databases (including the TAG ones). For TAG uploading, both the global job submission rate and the individual task types (i.e., each upload process to CERN and the external sites, respectively, as described in section 2.2.) are adjustable. The latter allows limiting the number of running upload jobs per task type by the maximum concurrent sessions limit of the respective site.

5.3.2. Controlling querying access

Reading the TAG database is the most resource-intensive and difficult to predict activity in the database. In theory, any user with the reader password can use the PoolCollection utilities widely available, or even connect directly to the database with any tool, and get the full database back in a file, if she wants to. This scenario must be avoided at all costs, and mechanisms put in place to make the risk lower.

There are three main services that are the preferred way to query the database: ELSSI website, Extract service and Skimming service. All of these are HTTP webservices, and as such, can control the access patterns to the databases, and additionally protect the reader passwords whenever possible.

The way to mitigate the risk of excessive load from the database point of view, is to define different profiles for different reader accounts, limiting the number of concurrent sessions, and if needed, even the number of calls or CPU spent in the database. The DBAs also have the power to kill sessions and monitor rogue queries that might go against the fair-share of resources. This will have to be an iterative process to determine what the best values are for each site, as resources are very different.

6. Managing and archiving the TAG relational data

6.1. Identification of read-only data and reprocessing passes

As soon as all the TAG files for a given run and physics stream have been uploaded, the data is made read-only, using the workflow described in 3.4. This means that if another set of files is inserting in the same run and physics stream, this data belongs to a new reprocessing pass.

It has been agreed in the storage capacity planning of the CERN ATLAS databases to keep only two reprocessing passes per year, for the TAGS database. This means that the data must be removed from the CERN databases, and possibly stored in a different type of database, with storage designed for archiving, not only TAGS but other application data as well. At the remote sites, this decision is left to its own hardware management policies.

6.2. How the schema model allows the archiving of data

The dividing of the collections by physics streams and release tag makes the TAG database schema look like a set of tables that are very well ordered in time, and have self-contained data. Moreover, each table is partitioned by run number, so each partition is a database segment of its own. The Oracle database allows that these segments be put in different tablespaces, for ease of management. As TAG data is never updated, we can be certain that a segment that has been marked as read-only is able to be archived, if we wish so.

This ease of data management is perfect for devising policies and mechanisms of data archival. One popular way to archive is to move the tablespaces from one Oracle database to another. Oracle allows this movement in a very transparent and straightforward manner using its “Transportable Tablespace” technology. At CERN, an archival system for relational data is being put in place using Oracle databases, pared down from its availability features, but with large inexpensive storage systems. This service will hold data that no longer needs to be “live” and heavily accessed, but should not be moved to tape, so it is ready for querying if some validation needs to be done against old data, using the same tools and structure.

7. Future Work

At the time of publication of this paper the upload technology described in Sections 3 and 4 is quite robust and actively used for uploading commissioning and Monte Carlo data at CERN and other 2 remote sites. Many improvements and fine tuning have made the process streamlined and automation has been put in place, to insure a quick TAG file to ELSSI turnover.

The aspects focused on Sections 5 and 6, controlling database usage and archiving of data, are a work in progress, where all the conditions for use are in place, but has not been extensively tested in production cases.

In addition there are two major steps to a truly distributed TAGS database, where work is being started and a plan put in place to be achieved in the medium term of one year:

- Central catalogue of the data distributed across sites. This component will make the ELSSI interface knowledgeable to where the data is located, and direct the user and internal configuration to the nearest extraction and skimming service. This requires filling up of the catalogue at upload time, and at refresh of the TAG Metadata
- Distributed Service Architecture. ELSSI web presentation, extraction, skimming, catalogue and TAG Metadata can be (and some already are) broken down into web services that are

hosted at different points and institutions across the world, regardless of the location of the TAG data. A catalogue of these services can be kept and used to decide the optimal composition of a full TAG querying service.

8. Conclusions

With less than a year to go for the start-up of the LHC, with real data to analyze, the ATLAS TAG database project team is confident that the requirements and operational challenges put forth by the ATLAS computing model will be met. Much effort has been put into making this a reality: application deployment is progressing at a fast rate, the technology is very well known by now and properly tested and all pieces are being put in their place in preparations for the data taking activities.

9. References

- [1] ATLAS Computing Group 2005 *Computing Technical Design Report - TDR* (CERN LHCC-2005-022 ISBN 92-9083-250-9)
- [2] Battaglia A, Beck H P, Dobson M, Gadomski S, Kordas K and Vandelli W 2007 The data-logging system of the trigger and data acquisition for the ATLAS experiment at CERN *Nuclear Science Symp. Conf. Record (Honolulu, Oct 2007)* vol 1 pp 527-532
- [3] Assamagan K, Barberis D et al. 2004 Final report of the ATLAS AOD/ESD Definition Task Force *ATLAS Notes Detectors and Experimental Techniques Software* (CERN ATL-COM-SOFT-2004-008)
- [4] Assamagan, K A et al. 2006 *Report of the Event Tag Review and Recommendation Group* / (CERN ATL-SOFT-PUB-2006-002)
- [5] Govi G, Chytrcek R, Duellmann D, Papadopoulos I and Xie Z 2006 POOL Developments for Object Persistency into Relational Databases *15th Int. Conf. on Computing In High Energy and Nuclear Physics(Mumbai)* (Mumbai: Macmillan) vol 1 pp.379-382
- [6] Brun R and Rademachers 1997 ROOT: An object oriented data analysis framework. *Nucl. Instrum. Meth. A* 389 pp 81–86.
- [7] CERN ATLAS Software 2009 *ATLAS Offline Software Repository* (<http://ATLAS-sw.cern.ch/cgi-bin/viewcvs-ATLAS.cgi/offline/Database/AthenaPOOL/POOLCollectionTools/sql/CollectionManagement/>)
- [8] Goossens L. 2007 *TAGS Scalability and Performance Testing –Preliminary results from the 1B TAG test* ATLAS Computing Workshop Munich (<http://indico.cern.ch/getFile.py/access?contribId=109&sessionId=8&resId=1&materialId=slides&confId=5060>)
- [9] CERN IT Service 2009 *AFS - The Andrew File System* (<http://consult.cern.ch/service/afs/>)
- [10] CERN ATLAS Software 2009 *ATLAS Tier-0 Monitoring Dashboard Website* (<https://ATLAS.web.cern.ch/ATLAS/tzero/prod1/monitoring/>).
- [11] CERN ATLAS Software 2009 *ATLAS Tier-0 Savannah Bug Reporting Portal* (<https://savannah.cern.ch/projects/ATLASpoint1t0/>).
- [12] ORACLE Corporation 2009 *Oracle Enterprise Manager* (<http://www.oracle.com/technology/products/oem/index.html>)
- [13] SourceForge.NET 2009 *Ganglia Monitoring System* (<http://sourceforge.net/projects/ganglia/>)
- [14] Nagios Enterprises 2009 *Nagios Open Source Monitoring System* (<http://www.nagios.org/>)