Evaluation of the ATLAS model for remote access to database resident information for LHC Run 3

Elizabeth J Gallas[1,*] and *Gancho* Dimitrov[2,**]

[1]University of Oxford, Denys Wilkinson Bldg, Keble Rd, Oxford OX1 3RH, UK
[2]CERN, 1211 Geneva 23, Switzerland

Abstract. The ATLAS model for remote access to database resident information relies upon a limited set of dedicated and distributed Oracle database repositories complemented with the deployment of Frontier system infrastructure on the WLCG (Worldwide LHC Computing Grid). ATLAS clients with network access can get the database information they need dynamically by submitting requests to a Squid proxy cache server in the Frontier network which provides results from its cache or passes new requests along the network to launchpads co-located at one of the Oracle sites (the master Oracle database at CERN or one of the Tier 1 Oracle database replicas). Since the beginning of LHC Run 1, the system has evolved in terms of client, Squid, and launchpad optimizations but the distribution model has remained fundamentally unchanged. On the whole, the system has been broadly successful in providing data to clients with relatively few disruptions even while site databases were down due to overall redundancy. At the same time, its quantitative performance characteristics, such as the global throughput of the system, the load distribution between sites, and the constituent interactions that make up the whole, were largely unknown. But more recently, information has been collected from launchpad and Squid logs into an Elasticsearch repository which has enabled a wide variety of studies of various aspects of the system. This contribution[***] will describe dedicated studies of the data collected in Elasticsearch over the previous year to evaluate the efficacy of the distribution model. Specifically, we will quantify any advantages that the redundancy of the system offers as well as related aspects such as the geographical dependence of wait times seen by clients in getting a response to its requests. These studies are essential so that during LS2 (the long shutdown between LHC Run 2 and Run 3), we can adapt the system in preparation for the expected increase in the system load in the ramp up to Run 3 operations.

1 Introduction

Modern particle physics experiments store data in a combination of file systems and database systems. ATLAS [1] is no different in this respect, but the scale of the data stored and processed is exceptionally high so the majority of this processing, by necessity, takes place across a world-wide computing grid. These data processing tasks require information from centralized databases at CERN such as configuration and control data, detector geometry, as well as,

[*] e-mail: Elizabeth.Gallas@physics.ox.ac.uk

[**] e-mail: Gancho.Dimitrov@cern.ch

for example, alignment derivatives and subsystem calibrations each of which is specialized to the calculations being performed and customized for the specific events being processed. Another case for database schema replication arises in the area of metadata.

This presentation describes the evolution of ATLAS database replication over the last 10 years (since the beginning of Run 1) in order to provide task-customized data from centralized database systems to event processing and other clients across the grid. We review the evolution of technologies over this period as well as the refinements in the data distribution infrastructure as the experiment gained operational experience over time.

In Section 2, we describe database-resident data the subset of that data which is needed by either processes across the grid or by applications at multiple sites. Section 3 revisits the initial (pre-Run 1) database distribution model and the evolution of that model as ATLAS entered into the LHC Run 1 data taking era. The further evolution of the system over Run 1 and Run 2 are recounted in Section 4. In Section 5 we lay out our future plans to address areas needing improvement. We conclude in Section 6 with a summary of the lessons learned, current challenges, and outlook for the future.

2 ATLAS Databases and replication use cases

Database information is used extensively in ATLAS at every stage of data taking and processing as well as for system monitoring and data discovery. Most of this data is recorded centrally in master Oracle [2] repositories at CERN and queried directly by application servers deployed within the CERN network. This section describes database-resident information needed at more than one site on the grid.

The main classes of data needed at sites outside of the CERN network are the configuration, conditions, calibration, and alignment data are needed by grid sites for event-wise processing, sub-system specific analysis, and monitoring. These data are stored in the ATLAS Trigger, Geometry, and Conditions database schemas. These schemas are based on the LCG COOL and CORAL [3] database infrastructure which includes API access methods to query for the specific data needed via C++ or python programs. Most of the tasks requiring database information at grid sites are running Athena [4] (the ATLAS event processing software framework) to process event data. There may also be tasks on the grid from subsystem expert processes which are using COOL or CORAL methods outside of Athena to retrieve this data for non-event processing or for monitoring purposes. These needs were anticipated before Run 1 data taking commenced, and the ATLAS Computing model provided for the replication of these specific schemas to grid sites hosting Oracle database infrastructure.

A feature of the COOL conditions database is that the data may include references to external files (e.g. data stored in a file system rather than in the database itself). The distribution model must therefore include the provision of those files whenever a client requests these database rows.

Another class of database-resident data which has proven useful to replicate to multiple sites is the AMI metadata repository for ATLAS [5]. Exceptionally, the master copy of this repository is resident outside CERN at the IN2P3 facility in France since it was developed and implemented by our colleagues in Grenoble, France. Since the data collected in the AMI database is useful beyond the AMI application itself, the AMI data is replicated to CERN to satisfy the needs of CERN-based systems directly as well as provide a fail-over for AMI services in the case of network or infrastructure degradation at IN2P3.

3 Initial model and evolution for Run 1

The pre-Run 1 database distribution implementation is shown in Figure 1. The left hand side shows that data is consolidated in the CERN central offline Oracle database (ATLR) including data from the online database (ATONR) as well as a replica of the AMI database from the IN2P3 site. Pre-Run 1 event processing tasks executed locally at CERN using direct Oracle database access of the ATLR CERN database showed very good performance and scalability in requesting the needed Trigger, Geometry, and Conditions data. But knowing that the number of tasks on the grid would grow over time, we worried about over-dependence on the CERN database and networks in terms of load and availability as tasks on the grid ramped up. Therefore, these specific schemas were replicated from the central ATLAS database at CERN to all 10 ATLAS Tier 1 sites as shown on the right side of Figure 1. In principle, making the data available at many sites across the grid on a geographic basis reduces the average network distance from worker nodes to database servers as well as allows requests to fail-over to another site in the case of individual Tier 1 downtime or degradation. In addition, COOL performance was optimized with SQL hints, and IOV[1] alignment was built into database access methods in Athena to take advantage of Oracle caching.

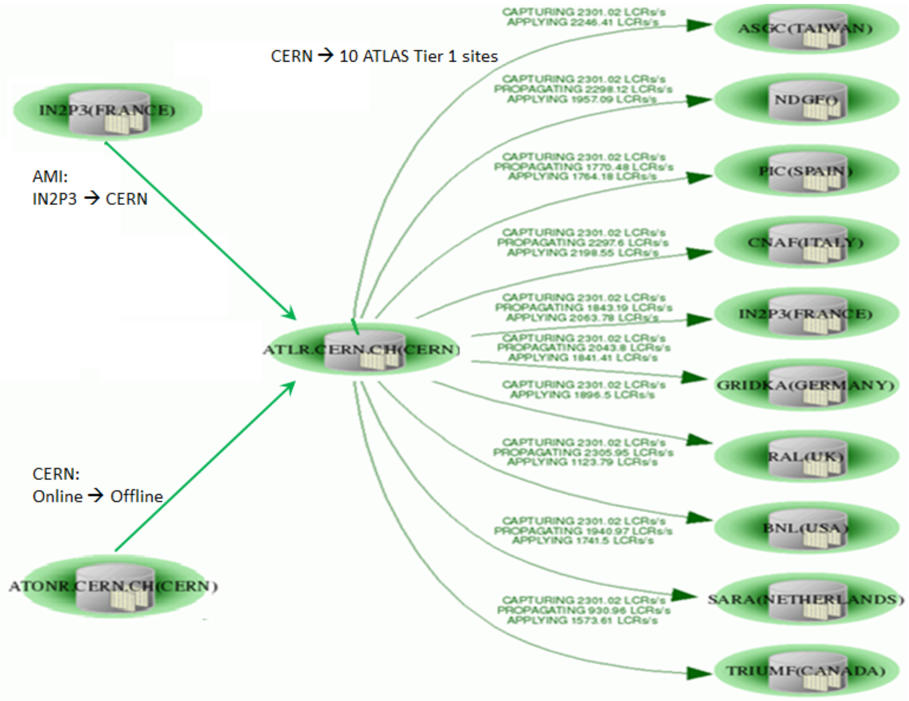

Figure 1. A snapshot of the ATLAS Database replication between Oracle RACs during 2008 (pre-Run 1). Arrows indicate the direction of Oracle Streams replication from source(s) to destination(s). The left side of the figure shows that data is consolidated in the central ATLAS Oracle offline database (ATLR) including data from the ATLAS online database (ATONR) as well as a replica of the AMI database from the IN2P3 site. Data in selected schemas in the central offline database are replicated to the 10 ATLAS Tier 1 sites. The Oracle Streams technology kept the data in sync within minutes of any changes on ATLR.

[1] IOV, or 'Interval of Validity' is the temporal range over which the condition or state is valid.

3.1 Run 1 database access from the grid

With this model deployed, the flaws in remote access quickly became apparent: Grid processing tasks on worker nodes outside of CERN and Tier 1 sites observed much slower response to its database requests while the databases themselves showed minimal load. The problem was found to be proportional to distance along the WAN (wide area network) compounded by Oracle communication protocols needing back and forth communication between client and server beyond a simple request-response interaction. During the same period, we had already been in the process of evaluating Frontier technology [6] which is a caching system which distributes data from data sources (in our case, the Oracle databases) to distributed clients (in our case, tasks on the grid) using straight-forward HTTP protocols. Using a prototype Frontier deployment at one of the Tier 1 sites, our colleagues in Tokyo ran dedicated tests on their local grid site comparing database access times using the three available access methods:

1. Data accessed from local SQLite DB files (Tokyo, Japan): DB access time 52 seconds.
2. Data access via Frontier (at BNL, New York, USA): DB access time 343 seconds.
3. Direct query of an Oracle replica (IN2P3, France): DB access time 18000 seconds !

This result is shown graphically in Figure 2 (note that the vertical scale is logarithmic). The quickest database access was, as expected, via local files at the grid site: this local file access test was included only to provide a benchmark and cross check (creating and distributing such files globally is not viable model for all processing needs). Despite the fact that the distance from Tokyo to New York (11,000 km) is about the same as the distance from Tokyo to Lyon (10,000 km) and that the former network path is across the Pacific Ocean, the data access via Frontier was over 50 times faster than direct Oracle access over the WAN. These tests confirmed our suspicions: while Oracle response time is excellent on local area networks (LAN), it is very slow over wide area networks (WAN). Further tests demonstrated that even over LANs, average access times are better via Frontier than via direct Oracle queries due to the Frontier Squid server caching.

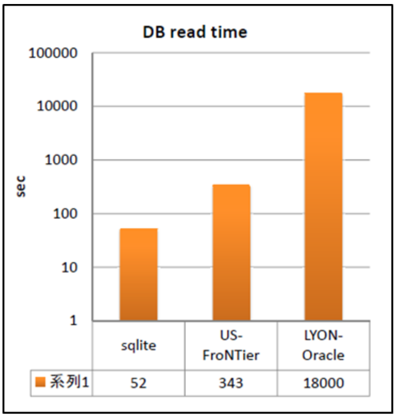

Figure 2. A graphical representation of the database access times observed from event processing tasks on local Tokyo worker nodes to three database sources: a local SQLite file, a Frontier deployment in New York (USA), and direct database access in Lyon (France). Data access was over 50 times faster via Frontier than direct Oracle access over a WAN.

The decision was therefore taken to complement every Oracle site hosting a replica with a requisite of Frontier server infrastructure to negotiate client requests to the partner database. This infrastructure was put in place in time for Run 1 operations to good effect. In addition to reducing average client wait times for data, Frontier also provides a buffer to queue database access requests which further insulates the database from spikes in load.

4 Evolution through Run 1 and Run 2

4.1 Evolution of schema replication

From the beginning of Run 1 and through Run 2, the system performed well overall. We found over time that distributing data to all 10 Tier 1s resulted in excess capacity primarily as a result of the Frontier deployment. Therefore, we gradually consolidated these services to just three sites. In 2015 (during LS1: the long shutdown between Run 1 and Run 2) the replication mechanism between sites was updated to use Oracle Golden Gate (Oracle Streams technology was deprecated). Figure 3 shows the ATLAS database distribution as of the date of this conference, with seven (of the ten) Tier 1 sites having phased out their Oracle replicas.

The remaining three sites focused on further refinement of the Frontier deployment. A further consolidation is anticipated soon, leaving 2 site replicas remaining:

- RAL (Oxfordshire, United Kingdom): currently phasing out Oracle support by 2020,
- TRIUMF (Vancouver, Canada): a geographically strategic replica in North America, and
- IN2P3 (Lyon, France): a useful fail-over principally for CERN in times of high load in addition to inversely providing the replication of the its AMI database to CERN.

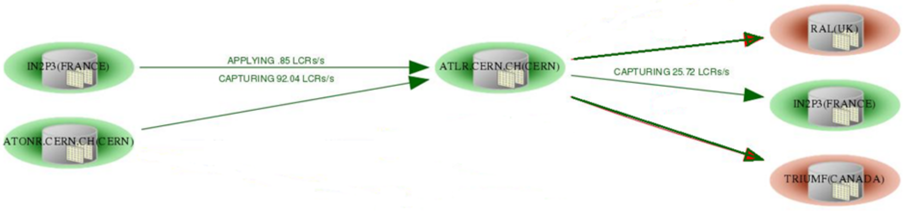

Figure 3. A snapshot of the ATLAS Database replication between Oracle RACs in 2019. The left hand side is identical to that in Figure 1 (2009 deployment). The needs of global event processing tasks are now served by the CERN central offline database and 3 ATLAS Tier 1 database replicas.

A major development in recent years is the deployment of an Elasticsearch [7] (ES) repository at the University of Chicago [8] to collect information from Frontier site logs. The Frontier deployment as well as collection of monitoring data into ES is shown schematically in Figure 4. These logs contain, in principle, every request to the database as well as include requests which were satisfied by the cache at the Frontier sites. This repository enables us to monitor the system globally, identify transactions which require more resources than others, evaluate cache efficiency in various contexts, as well as study the details of individual requests. An interesting global metric is the overall load carried by each of the Oracle sites. With the RAL site being phased out recently, the load re-balanced toward the remaining sites: CERN satisfied 55% of requests while IN2P3 and TRIUMF carried 26% and 17% of the load, respectively.

Generally, for normal event processing workflows, the system provides smooth and efficient retrieval of the data requested by tasks. But on occasion during Run 2, we started to observe service degradation and failures within the Frontier system. Upon investigation, the problematic periods coincided with the execution of special workflows, so a monitoring application was set up to investigate the database demands of these tasks as described in another presentation in this conference [9] which found that a main issue affecting system performance is the cacheability of requests. While in some cases it has been possible to refine the requests of these tasks, in other cases, we need to modify the underlying schema implementation because some COOL storage options are inherently not cache compatible.

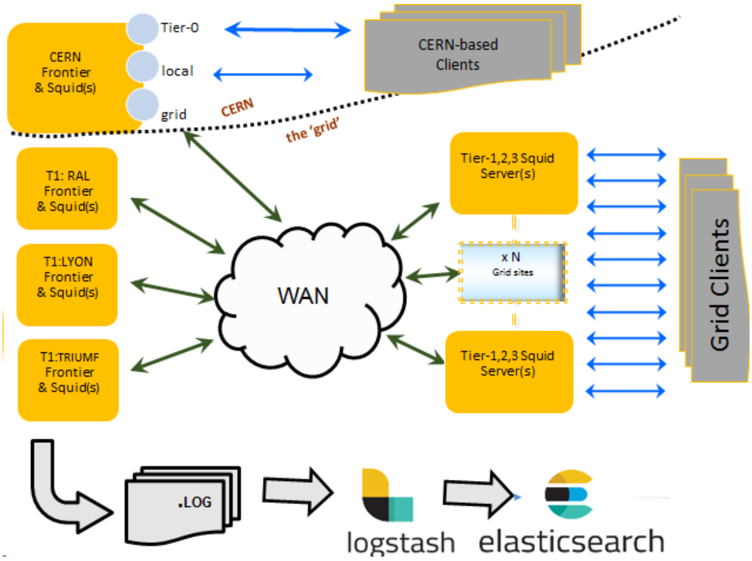
CERN Frontier & Squid(s)
Tier-0
local
grid
CERN-based Clients
CERN
the 'grid'
T1: RAL Frontier & Squid(s)
T1:LYON Frontier & Squid(s)
T1:TRIUMF Frontier & Squid(s)
WAN
Tier-1,2,3 Squid Server(s)
x N Grid sites
Tier-1,2,3 Squid Server(s)
Grid Clients
.LOG
logstash
elasticsearch

Figure 4. A schematic of global ATLAS Frontier deployment to serve the database needs of processing tasks across the world-wide ATLAS computing grid. Data from Frontier site log files are collected in an Elasticsearch repository which enables monitoring globally as well as studies of individual transactions.

While the primary area of development relates to improving the cacheability of requests, we are also studying other factors influencing system performance such as understanding the effects of requested data volume and row counts, studying queries which are more database-resource intensive, and optimizing the geographic distribution of our Oracle replicas and Frontier Squid servers to best serve our clients across the globe. These studies will influence our strategy for future distributed database deployment to ensure we provision the system overall to handle the higher request rates anticipated in Run 3.

4.2 Evolution of conditions file distribution

We have 2 categories of file-based data related to databases with global distribution requirements:

1. **Externally referenced conditions files:**
   The LCG COOL database infrastructure allows in-line database records to contain references to external files identified by GUID (Globally Unique Identifiers) which uniquely point to a particular conditions file. This option has been utilized by many subsystems for data which is stored in files rather than in the conditions database itself due to their size or needs in other contexts.

2. **DB Release files**:
   DB Release [10] files are an aggregation of both SQLite database replicas packaged along with the associated conditions files (above) referenced by the database. These files allow event processing jobs to execute without network access such as on HPCs.

Both types of files have been a component of ATLAS database infrastructure through Run 1 and Run 2 and have permanent registration within the DDM/Rucio [11] system (ATLAS distributed file and dataset management).

While Rucio is capable of providing requested files dynamically, this mechanism is too slow for running jobs in real time. Originally (during Run 1), conditions files were distributed to sites via POOL file registration: Sites were required to host a 'hotdisk' containing all conditions files along with a local file catalog (LFC) so that these files could be quickly retrieved as needed by local jobs. DB Release files were distributed to sites running the job (MC only) but this eventually became unsustainable due to the cumulative volume of evolving MC conditions.

Since Run 1, these file systems have been made available via CVMFS [12] which has been a big improvement over maintaining up-to-date file systems and LFCs at all sites. In addition, DB Release production has been streamlined to contain only the latest 'best knowledge' MC conditions data which reduced the data to a manageable volume on worker nodes.

4.3 AMI replication to CERN

The AMI application has a long history in ATLAS, celebrating its 10 year anniversary in CHEP 2013 [13]. Since its inception, it has continued to evolve with the latest technologies and has been extended in various ways to accommodate new use cases. It is now a mature ecosystem which has proven to be resilient, robust, scalable, and adaptable. While its schema has evolved, its replication from the IN2P3 site to CERN (shown on the left hand side of Figure 1) has remained unchanged through both Run 1 and Run 2. Over this period, this data at CERN grew in popularity, being used by other metadata repositories [14, 15] at CERN as well as other systems. The replica additionally provides a fail-over for AMI services in the case of network or infrastructure degradation at IN2P3.

5 Outlook and Future Direction

With the main bottleneck (cache efficiency) being in the conditions database area (COOL), late in Run 2 we met with an expert panel to review the system to agree on plans for Run 3 operations and Run 4 evolution: It was concluded that ATLAS must continue to use COOL for Run 3 generally (because COOL is currently so deeply embedded in ATLAS software and workflows) but take action to modify areas of problematic COOL storage by some subsystems. For Run 4, we aim to move towards modernizing our conditions infrastructure to use a new system called CREST [16] which is based on a data storage model which will improve the efficiency of caching and features RESTful access to the data (which simplifies code on the client side).

6 Summary and Conclusion

We have described the evolution of the distribution of database-related data and associated files in the ATLAS experiment since the beginning of LHC Run 1. The deployment of Frontier and CVMFS resulted in considerable improvements for clients in terms of data availability, throughput and efficiency and for Oracle sites in terms of load leveling and security. These changes also enabled a consolidation of services at sites which optimizes global resources and minimizes maintenance.

The collection of Frontier log data into an Elasticsearch repository has enabled a rich variety of studies of various aspects of the system which were previously not possible to assess quantitatively. As a result we have a much better picture of the system globally (i.e. overall throughput, load distribution between sites, etc.), are able to analyze the requests of problematic workflows, and investigate any constituent interactions that seem problematic.

While the analysis is ongoing, we have gathered sufficient evidence showing the main areas to be improved: Use a data storage model which is compliant with caching and implement RESTful access to that data. Unfortunately, the existing COOL implementation does not easily conform to this modern paradigm, i.e. we cannot start from scratch. Rather, improvements must also meet the additional challenge of converting existing data into the new storage model along with the associated changes in the software in order to maintain our ability to process and analyze legacy data.

The basic need for distributed database-resident data and associated files remains unchanged and will only increase in the future. Just as in the previous LHC runs, the storage and distribution of database related services will continue to evolve to meet the needs of the ever-growing needs of clients across the globe.

References

[1] ATLAS Collaboration, *"The ATLAS Experiment at the CERN Large Hadron Collider"* JINST **3** S08003 (2008)

[2] Oracle Database, http://www.oracle.com

[3] R Trentadue, et.al., *"LCG Persistency Framework (CORAL, COOL, POOL): status and outlook in 2012"* J. Phys.: Conf. Series **396** 053067 (2012)

[4] P Calafiura, et.al., *"Running ATLAS workloads within massively parallel distributed applications using Athena multi-process framework (AthenaMP)"* J. Phys.: Conf. Series **664** 072050 (2015)

[5] J Fulachier, J Odier, and F Lambert, *"ATLAS Metadata Interface (AMI), a generic metadata framework"* J. Phys.: Conf. Series **898** 062001 (2017)

[6] D Barberis, et.al., *"Evolution of grid-wide access to database resident information in ATLAS using Frontier"* J. Phys.: Conf. Series **3**96 052025 (2012)

[7] Elasticsearch: https://www.elastic.co/

[8] M Svatos, et.al. *"Understanding the evolution of conditions data access through Frontier for the ATLAS Experiment"* EPJ Web Conf. **214**, 03020 (2019)

[9] A Formica, N Ozturk, M Si Amer, J Lozano Bahilo, E J Gallas, and I Vukotic, *"An Information Aggregation and Analytics System for ATLAS Frontier"* EPJ Web Conf. for CHEP'19 (to be published)

[10] L Rinaldi, E J Gallas, and A Formica, *"Optimizing access to conditions data in ATLAS event data processing"* EPJ Web Conf. **214** 04051 (2019)

[11] V Garonne, et.al., *"Rucio – The next generation of large scale distributed system for ATLAS data management"* J. Phys.: Conf. Ser. **513** 042021 (2014)

[12] P Buncic, et.al., *"CernVM - a virtual software appliance for LHC applications"*, J. Phys.: Conf. Ser. **219** 042003 (2010)

[13] J Fulachier, O Aidel, S Albrand and F Lambert *"Looking back on 10 years of the ATLAS Metadata Interface"* J. Phys.: Conf. Ser. **513** 042019 (2014)

[14] E J Gallas, et.al., *"Conditions and configuration metadata for the ATLAS experiment"* J. Phys.: Conf. Ser. **396** 052033 (2012)

[15] D Barberis, et.al., *"The ATLAS EventIndex: architecture, design choices, deployment and first operation experience"* J. Phys.: Conf. Ser. **664** 042003 (2014)

[16] L Rinaldi, A Formica, E J Gallas, N Ozturk, and S Roe, *"Conditions evolution of an experiment in mid-life, without the crisis (in ATLAS)"* EPJ Web Conf. **214** 04052 (2019)