Evolution of grid-wide access to database resident information in ATLAS using Frontier

D Barberis[1], F Bujor[2], J de Stefano[3], A L Dewhurst[4], D Dykstra[5], D Front[6], E Gallas[7], C F Gamboa[3], F Luehring[8] and R Walker[9] on behalf of the ATLAS collaboration

[1] INFN Sezione di Genova and Dipartimento di Fisica, Universitá di Genova, Genova, Italy
[2] CERN, Geneva, Switzerland
[3] Physics Department, Brookhaven National Laboratory, Upton NY, United States of America
[4] Particle Physics Department, Rutherford Appleton Laboratory, Didcot, United Kingdom
[5] Computing Division, Fermilab, Batavia, IL, USA
[6] Department of Particle Physics, The Weizmann Institute of Science, Rehovot, Israel
[7] Department of Physics, Oxford University, Oxford, United Kingdom
[8] Department of Physics, Indiana University, Bloomington IN, United States of America
[9] Fakultät für Physik, Ludwig-Maximilians-Universität München, München, Germany

E-mail: alastair.dewhurst@cern.ch

Abstract. The ATLAS experiment deployed Frontier technology world-wide during the initial year of LHC collision data taking to enable user analysis jobs running on the World-wide LHC Computing Grid to access database resident data. Since that time, the deployment model has evolved to optimize resources, improve performance, and streamline maintenance of Frontier and related infrastructure. In this presentation we focus on the specific changes in the deployment and improvements undertaken such as the optimization of cache and launchpad location, the use of RPMs for more uniform deployment of underlying Frontier related components, improvements in monitoring, optimization of fail-over, and an increasing use of a centrally managed database containing site specific information (for configuration of services and monitoring). In addition, analysis of Frontier logs has allowed us a deeper understanding of problematic queries and understanding of use cases. Use of the system has grown beyond just user analysis and subsystem specific tasks such as calibration and alignment, extending into production processing areas such as initial reconstruction and trigger reprocessing. With a more robust and tuned system, we are better equipped to satisfy the still growing number of diverse clients and the demands of increasingly sophisticated processing and analysis.

ATL-SOFT-PROC-2012-058
20 June 2012

1. Introduction

For ATLAS, conditions data is defined as event independent time varying information. It is stored separately from the collision data but needs to be accessed in the same jobs that process data. There are two elements to conditions data access; access to the COOL[1] relational database itself and access to conditions data POOL files which are used to store some large calibration objects (eg., calorimeter calibration data or Inner Detector alignment). When POOL files are used, the COOL database just contains a reference to the appropriate POOL file. The relational database data can be read from an Oracle server, or from an SQLite replica file.

For Grid jobs, the POOL files are distributed to sites and stored on their storage elements through FTS as a collection of files known as a DBrelease. For Monte Carlo and some types of production work, where the type of conditions data is known before hand this can be made into an SQLite file and distributed with the software release. However for analysis jobs and some other types of production work, access to the conditions data stored in the Oracle database is needed.

The ATLAS experiment deployed Frontier technology world-wide during the the initial year of LHC collision data taking to enable user analysis jobs running on the World-wide LHC Computing Grid to access this database resident information more efficiently.

The protocol for Frontier is http-based and uses a RESTful architecture which is excellent for caching and scales well [2]. The Frontier system uses the standard web caching tool squid to cache the http objects at every site. It is ideal for applications where there are large numbers of widely distributed clients that read basically the same data at close to the same time, in much the same way that popular websites are read by many clients.

The 2011 LHC run was extremely successful in terms of the amount of data collected compared to what was expected. Figure 1 shows how the conditions database (known as 3D) has grown throughout the 2011 data taking run. It also shows how the number of operations per second (reads and writes) has changed throughout the year. The size of the 3D database has grown at a constant rate during 2011 data taking and this is expected to continue into 2012. The number of operations per second has many spikes in it but the overall trend shows a higher rate in the second half of 2011.

2. Frontier and Squid Deployment

During a data taking run conditions data is written into a central Oracle database at CERN. This data is then streamed to several Tier 1 sites. These sites also provide a Frontier service. Originally ATLAS setup replicas of the Oracle conditions databases at ten Tier 1 sites: BNL, CNAF, FZK, IN2P3, PIC, RAL, TRIUMF, ASGC, NDGF, SARA. When the Frontier service was deployed in 2008-2009, it was realised that fewer Oracle/Frontier sites are needed and now 5 Tier-1 sites (BNL, FZK/KIT, IN2P3, RAL and TRIUMF) provide access to conditions data for all ATLAS sites.

The recommended site setup for a Tier 1 is to have two Frontier launchpads in a load balanced pair. This is primarily for resilience and allows the site to perform scheduled maintenance on one Frontier launchpad without affecting the availability of the service.

Every site that runs ATLAS analysis jobs has been asked to provide a squid for their worker nodes to use before connecting to a Frontier server. For larger sites (> 500 job slots), the recommendation is to have at least two squids in a load balanced configuration. This is for both load and resilience reasons.

It has been observed that the there is very little performance loss for jobs if they are having to connect to a distant Frontier server as long as the squid they are connecting to is close. This is for two reasons, firstly the amount of data that can be cached is very large and secondly the queries that can't be cached are often smaller in size. CMS relies on this by having their conditions database only at CERN and using site squids around the world to provide a fast service.

Figure 2 is a plot showing the hit ratio in terms of bytes cached for requests going through a typical site squid. In this instance the average is almost 90%. The exact amount of caching will depend on the size of the farm using the squid and the type of jobs ran there.

3. Tier 0 usage

For 2011, Tier 0 processing directly accessed the 3D database for its conditions data. However the load on this database was frequently too large. Accessing the 3D database through Frontier

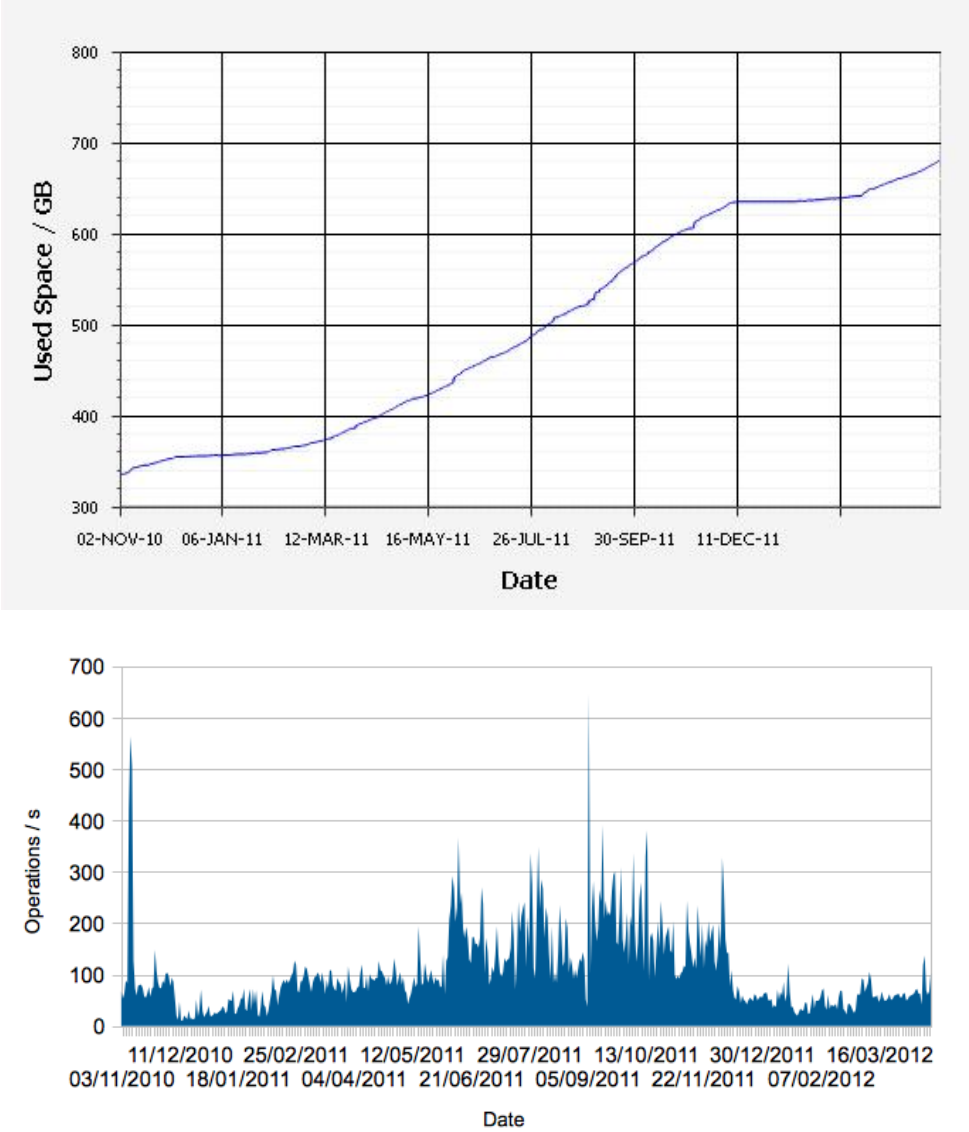

Figure 1. (Top) Figure showing how the size of the 3D database has grown during during 2011. (Bottom) Figure showing the number of operations per second averaged per day on the RAL 3D database during 2011.

was an obvious way to partly reduce the load. However there was a complication: One of the reasons why Frontier hadn't been used earlier was because of concerns about the stale information being cached. Through testing of the ATLAS code, it was demonstrated that there was no difference between the results with and without Frontier; however, it was decided that the cache refresh time should be 5 minutes.

A stress test was conducted to see how effective Frontier would be using this setup. Jobs were submitted at twice the normal rate of 80 per minute and their length reduced to processing just 10 events each. In a typical ATLAS job the conditions data is required at the start of the job, therefore this setup maximised the load.

Figure 3 shows the number of hits to the one of the CERN squids and Frontier launchpads during the stress test. The squid was in production which is why there are hits throughout

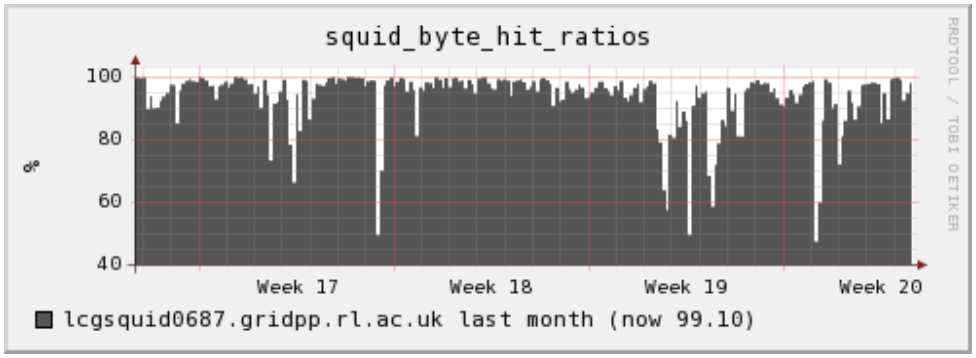

Figure 2. Figure showing the squid hit ratio in terms of bytes (rather than hits) over the course of a month. This plot was taken from the RAL squids.

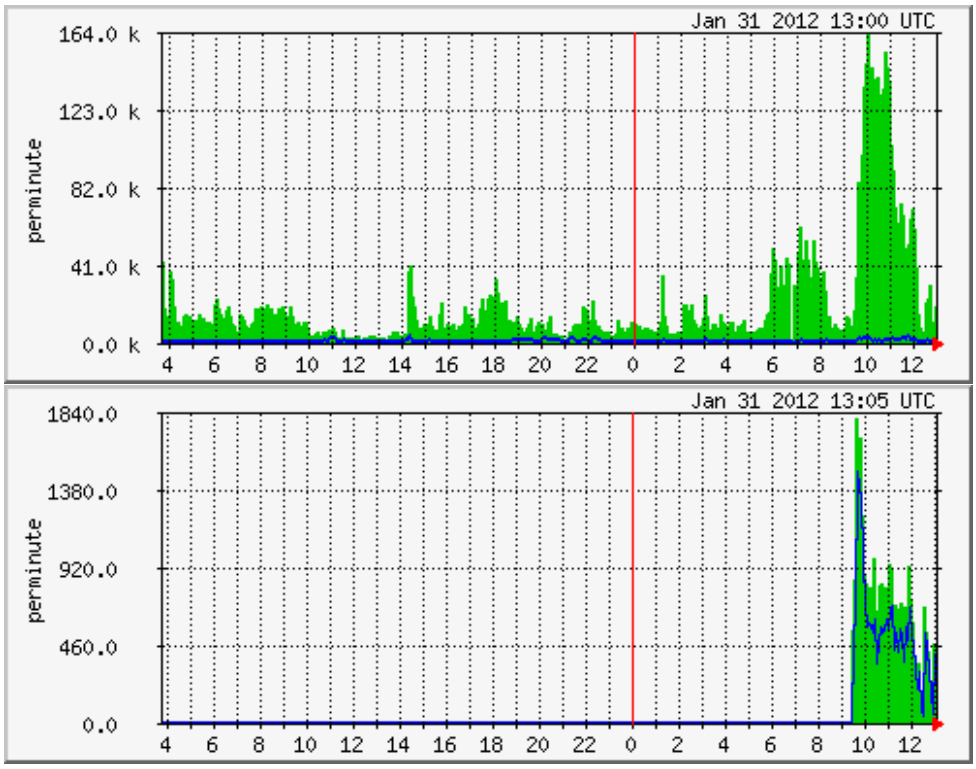

Figure 3. Figure showing the number of hits on the Squid (top) and Frontier launchpad (bottom) during the Tier 0 load test

the monitoring period. The Frontier launchpad was setup specifically for Tier 0 usage and hits to this are just from the stress test. The caching reduced the number of queries to the Oracle database by a factor of 100.

4. AGIS and site configuration

One of the key features of Frontier is the ability to use fail-over squids and Frontier launchpads if necessary. If a job has a problem connecting to its first choice squid or Frontier launchpad then it will try alternatives. ATLAS requested that if the site squid fails then Frontier queries

should fail-over to a nearby site squid. This way performance is maintained despite any failures.

Initially sites were configured using "Tiersofatlas". This is a simple python file with a line for each site's configuration. As mentioned earlier, the recommended site configuration is to have a pair of squids in a round robin alias. Therefore in addition to the alias ATLAS needs to know the individual squids behind the alias so they can use these as failover and monitor them. This was not possible with Tiersofatlas. In order to keep track of site configuration (including Frontier) ATLAS has developed the ATLAS Grid Information System (AGIS)[3].

The overall purpose of AGIS is to store and to deploy static and semi-static information about services, resources, configuration parameters and topology of the whole ATLAS computing grid. AGIS retrieves different types of information from the diverse sources coming from the different sub-grids.

Within AGIS, Frontier is defined in terms of services and configurations. The machines running the launchpads and the squids are defined in services and each site has a configuration which tell it which services to use:

- Frontier Service - For each Tier 1 running a Frontier launchpad a Frontier Service is defined.
- Frontier Configuration - Every site has a Frontier configuration made up of a primary and backup Frontier Service.
- Squid Service - Every site is assumed to have a squid service.
- Squid Configuration - Every site has a Squid configuration made up of a list of backup Squid services. By default the site squid is assumed to be the primary squid.

Each squid service has a flag to describe if it should be monitored or not. When the Frontier configuration is changed in AGIS a job will run and configure the site with a primary and backup Frontier and Squid servers. AGIS can also be used by monitoring tools to find out which nodes need to be monitored.

5. Monitoring

Distributed with the Frontier RPMs is the AWstats package [4]. This performs detailed analysis of the log files produced by the Frontier launchpad. It is useful in debugging problems. There is also SLS monitoring of the servers.

At the time of writing there are 98 squids in production for ATLAS at grid sites around the world. In order to monitor them correctly, ATLAS use two different tools. These are:

- **MRTG monitoring** of every squid allows ATLAS to monitor the number and type of requests coming in. This tool provides the clearest evidence that a squid is working and ATLAS computing shifters are required to check this regularly.
- **SUM Test** This is a simple job run at each site. This takes the Frontier and Squid configuration for each site and tries each combination of primary and backup to make sure they are working. As well as being an independent check that the squid is working this also allows ATLAS to monitor that the squid ACL's are correctly configured.

To properly monitor the services, ATLAS needs to know if there are any scheduled interventions planned. In order to do this, the squid service needs to be declared in somewhere like the GOCDB or OIM. At the start of 2012 these features were added and ATLAS is now testing implementing this in its automatic blacklisting procedure. ATLAS is also requesting that Frontier/Squid be recognised as an official service in WLCG.

For ATLAS shifters all the monitoring information is displayed in a single page which they are requested to check 3 times a day and report problems directly to sites.

6. Conclusions

Since LHC data taking started the Frontier has provided an effective means to provide conditions data and protect the Oracle databases from too much use. So successful has this been that the main problems in the evolution of Frontier use have been managing the large number of squids now deployed across the grid. This has been achieved through the use of AGIS to store the configuration details and through effective monitoring backed up by shifters actively checking sites.

ATLAS is confident that Frontier will provide the solution for accessing conditions data throughout the 2012 running and beyond.

References

[1] S. A. Roe, on behalf of the ATLAS experiment "A RESTful Web service interface to the ATLAS COOL database ," J. Phys. Conf. Ser. **219** (2010) 042021.

[2] D. Dykstra, "Scaling HEP to Web size with RESTful protocols: The frontier example," J. Phys. Conf. Ser. **331** (2011) 042008.

[3] `https://twiki.cern.ch/twiki/bin/view/Atlas/AtlasGridInformationSystem`

[4] `http://awstats.sourceforge.net/`