EPJ Web of Conferences **214**, 03020 (2019) https://doi.org/10.1051/epjconf/201921403020
CHEP 2018

Understanding the evolution of conditions data access through Frontier for the ATLAS Experiment

Michal Svatos[1,*], *Alessandro* De Salvo[2,**], *Alastair* Dewhurst[3,***], *Emmanouil* Vamvakopoulos[4,****], *Julio* Lozano Bahilo[5,†], *Nurcan* Ozturk[6,‡], *Javier* Sanchez[5,§], and *Dave* Dykstra[7,¶] , on behalf of the ATLAS Collaboration

[1]Institute of Physics of the CAS, Na Slovance 1999/2, Prague, 18221, Czech Republic
[2]Sapienza Universita e INFN, Roma I
[3]Science and Technology Facilities Council STFC
[4]Centre National de la Recherche Scientifique
[5]Instituto de Física Corpuscular (University of Valencia and CSIC)
[6]University of Texas at Arlington, Department of Physics, Arlington 76019 Texas, USA
[7]Scientific Computing Division, Fermilab, Batavia, IL, USA

Abstract. The ATLAS Distributed Computing system uses the Frontier system to access the Conditions, Trigger, and Geometry database data stored in the Oracle Offline Database at CERN by means of the HTTP protocol. All ATLAS computing sites use Squid web proxies to cache the data, greatly reducing the load on the Frontier servers and the databases. One feature of the Frontier client is that in the event of failure, it retries with different services. While this allows transient errors and scheduled maintenance to happen transparently, it does open the system up to cascading failures if the load is high enough.
Throughout LHC Run 2 there has been an ever increasing demand on the Frontier service. There have been multiple incidents where parts of the service failed due to high load. A significant improvement in the monitoring of the Frontier service was required. The monitoring was needed to identify both problematic tasks, which could then be killed or throttled, and to identify failing site services as the consequence of a cascading failure is much higher. This presentation describes the implementation and features of the monitoring system.

1 Introduction

The ATLAS [1] Distributed Computing (ADC) runs O(100k) grid jobs on more than a hundred grid sites. Each job accesses Conditions, Trigger, and Geometry database data. Some jobs use only a minimal amount of data while some some jobs access a significant amount of

*e-mail: Michal.Svatos@cern.ch
**e-mail: Alessandro.de.Salvo@cern.ch
***e-mail: Alastair.Dewhurst@cern.ch
****e-mail: emmanouil.vamvakopoulos@cc.in2p3.fr
†e-mail: julio.lozano.bahilo@cern.ch
‡e-mail: Nurcan.Ozturk@cern.ch
§e-mail: javier.sanchez@ific.uv.es
¶e-mail: dwd@fnal.gov

those data. If a job cannot access the data, it fails.
During Run 2 of the LHC, the number of database accesses increased. There are several reasons for that: new more complex workloads, increasing number of computing resources, etc. There have been multiple incidents where load generated by ATLAS grid workloads caused parts of the service to fail. If the Frontier system at one site fails, conditions data are searched at another site which can overload and break its Frontier system too. As a result, effort has been put into significantly improving detection and monitoring of problematic tasks.
Frontier servers [2], [3], [4] have both a Squid process for caching and a Tomcat process for converting between http and Oracle protocols. A new real-time monitor in Kibana has been developed to analyse the Tomcat logs.
All Squids are monitored by the Multi Router Traffic Grapher (MRTG) tool, i.e. the MRTG monitors both the Squids on Frontier servers and the Squids at sites. The AWStats tool is also used to monitor the Squids on Frontier servers and backup proxies. Several other monitoring pages based on AWStats and MRTG data are available for shifters and experts.

2 Condition database requests monitoring and alarms

Detailed information concerning each single query to the Conditions database performed by any ATLAS job is recorded in the Frontier server Tomcat log files. This is the basis for a new monitoring system making use of the ELK-stack (Elasticsearch/Logstash/Kibana) [5] services deployed at CERN and the University of Chicago [6]. The workflow of the information follows the schema shown in Figure 1:

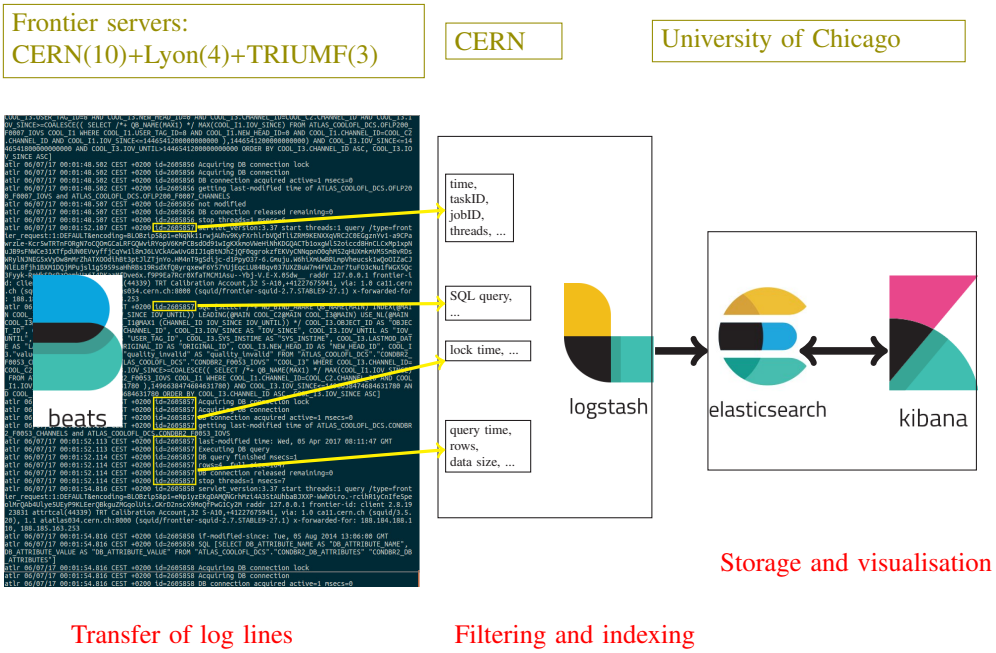

Figure 1. Schema of ELK-based Frontier monitoring

- The log lines from Frontier servers are read and handled by a Filebeat (Beats) service running in each server. It ensures that every single line is transferred sequentially to the

computer where it will be processed. Filebeat allows some flexibility in the way the lines are grouped and even whether uninteresting lines may be skipped. Currently we are monitoring Frontier servers located at the CERN and TRIUMF laboratories and a computing centre at Lyon.

- Relevant information is extracted by Logstash instances running in a single computer at CERN. They use a filter configuration file that handles the information of a single query which appears scattered among several log lines. Queries are identified by a unique ID within a Frontier Tomcat subsystem called servlet. The extracted information includes a timestamp of the query, the number of queries being treated concurrently, machines involved, DN of job owner, SQL query details, times for different processing steps, status of the query (if it failed, how it failed), etc. Additional code within the filter file consults a SQLite database filled with information related with the job and the task it belongs to like the task identification number and the processing type. Those details are also included in tuples of variables that that are built in the configuration file and finally sent for storage as indices to an ElasticSearch database.
- The ElasticSearch DB is a highly scalable platform which allows very quick searches on large amounts of data. Our Frontier servers deal with several millions of queries daily and all their information must be treated almost in real time to allow a performant monitoring system. This infrastructure is located at the University of Chicago and also holds information on the behaviour of many other important ATLAS computing services.
- A Kibana server sharing the ElasticSearch resources at the University of Chicago allows the visualization of the data in diverse ways, like histograms, tables and pie charts. It has a web server frontend to facilitate the interaction with the underlying ElasticSearch DB data. The definition of those visualization objects can be stored and also grouped into Dashboards. Therefore, the most relevant objects can be shown together in a single web page or even linked in several pages for the usage of people monitoring the Frontier server activity. Figure 2 shows a Kibana web page with two relevant blocks: the number of concurrent queries (there is a hard-coded limit on each server above which queries are rejected), and statistics about queries not cached in the Squid that have to be served by the Oracle database. Figure 3 presents distributions of high execution times grouped according to most relevant task identification numbers (top) and Frontier servers (bottom). These are just a couple of examples of pages containing visualization objects that appear in the official ATLAS Frontier Dashboard.

The Kibana Dashboard is complemented with an Alarms and Alerts (A&A) system to inform experts of foreseeable degradations of the performance of the Frontier servers. A set of conditions which, according to our experience, indicate that the servers are close to a state of saturation, were established. When such conditions are fulfilled, the Alarms and Alerts (A&A) system is triggered. This A&A system is implemented on the basis of two components:

1. An electronic mail subscription service implemented with the help of Google Forms. Any ATLAS collaborator can fill a form to subscribe to any of the various mailing lists used to submit specific A&A messages.

2. A Jupyter notebook [7] which is executed periodically by a cron job. It queries the ElasticSearch DB and looks for situations where the maximum number of concurrent queries is above a given limit, when the number of queries that were rejected or disconnected or that failed is above some threshold value, or when there is an abnormally high percentage of queries with high total execution times (above 1 s, whereas usually

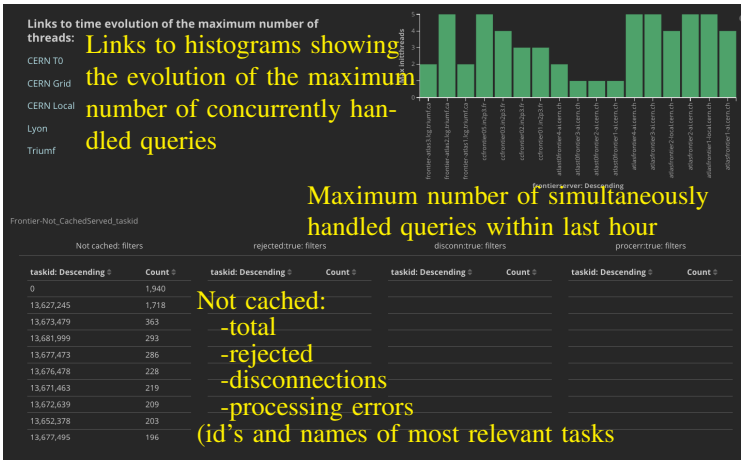

Figure 2. Kibana Dashboard: histogram of concurrent queries (top right) and statistics of not-cached queries (bottom)

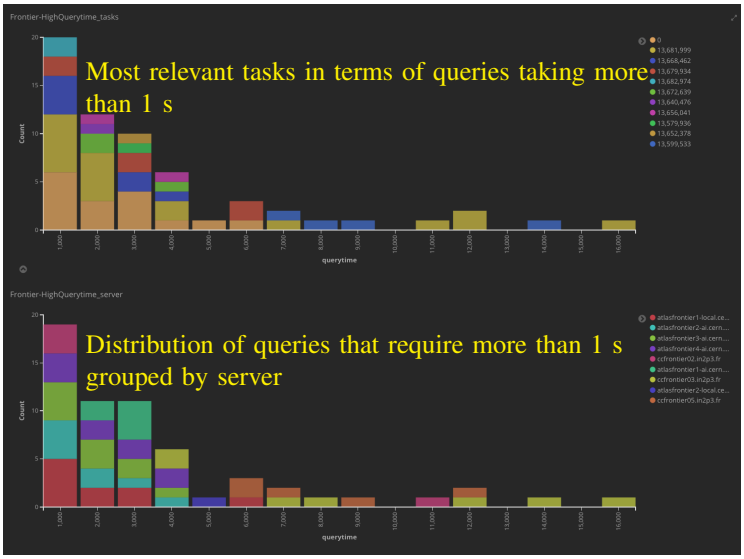

Figure 3. Kibana Dashboard: distribution of time-demanding queries grouped by task ID's (top) and Fontier servers (bottom)

it takes milliseconds). Some relevant numbers and information of specific tasks (identification number and name) is submitted via electronic mail messages to subscribed experts.

3 AWStats

AWStats [8] is a log analyser that generates advanced web, streaming, ftp or mail server statistics. Data from Squids running on each ATLAS Frontier service or backup proxy are

collected and displayed by this tool. The information which AWStats monitoring (Figure 4) provides are: summary of number of unique visitors, number of visits, pages, hits, and bandwidth; plots of those values in various time frames; Top 10 pages, hosts and their countries/domains; visit duration; request size and time; HTTP Status codes; TCP messages of Squid Cache hits.

Figure 4. AWstats monitoring page

4 Maxthreads

Maxthreads monitoring (Figure 5) checks the maximum number of threads used by frontier servlets in Tomcat. The number of threads corresponds to the number of queries that are queued waiting for a relatively small number of database connections. It also monitors client response time and DB query time. In case the number of threads exceeds a predefined threshold, an alarm email is sent to experts. The maxthreads monitoring will probably be decommissioned as the monitoring system described in Section 2 covers its functionality.

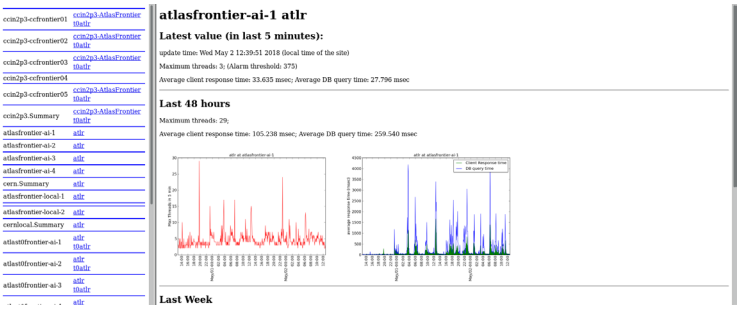

Figure 5. Maxthreads monitoring page

5 Availability in Kibana

The availability of Frontier servlets is also displayed in the Kibana monitoring (Figure 6) of ATLAS central services. The page displays availability as simple green field with one hour granularity. It is created by sending query probes every 5 minutes from the CERN Frontier monitoring machine. Whenever the status changes for two 5-minute periods in a row an email alert is sent. It is regularly checked by shifters.

Figure 6. Kibana monitoring page of Frontier servlets

6 MRTG

The Multi Router Traffic Grapher (MRTG) [9] is a tool to monitor the traffic load on network links. It uses SNMP to read the traffic counters from Squids and creates graphs representing the traffic on the monitored network connection. The output of the tool is HTML pages with those graphs. Plots in several time-windows are available: one day, last 7 days, last five weeks, and last twelve months.

The newly developed ATLAS MRTG monitoring page (Figure 7) is based on the WLCG MRTG page. Site Squids registered in GOCDB/OIM are monitored by the WLCG Squid page. Active ATLAS sites with active Squids (according to AGIS) are picked from them. For sites which do not fit this schema, there is a possibility to be added via an exception file. Plots for picked sites are displayed in form of table. The ATLAS MRTG monitoring page has several views. It can display all site Squids in one page. It has also views based on Tier and Cloud. There is also a per-site view.

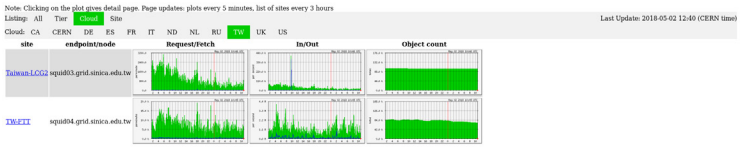

Figure 7. MRTG monitoring page

7 SSB

Site Status Board (SSB) is a WLCG monitoring framework used by ATLAS to monitor various functionalities of sites. The Squid monitoring in the SSB aggregates site status from the ATLAS MRTG page. A site Squid is considered OK if its MRTG page returns its Squid version or (in case it does not return it) if the average number of HTTP requests in the last 30 minutes is above zero. The SSB Squid monitoring (Figure 8) displays OK if all of a site's Squids are OK, down if all of a site's Squids are down or degraded if some of a site's Squids are OK and some are down. It is regularly checked by shifters.

8 Failover monitor

The Failover monitoring page (Figure 9) was adapted from a CMS monitoring page used to see failovers from worker nodes to Frontier server Squids and backup proxies. This was

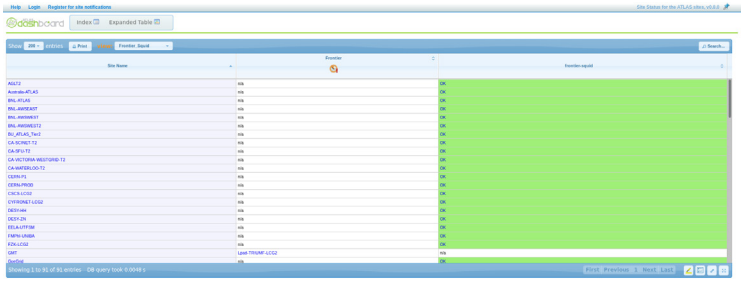

Figure 8. SSB dashboard

setup because direct connections from worker nodes (when a site's Squid is not working) to Frontier server can cause an overload.

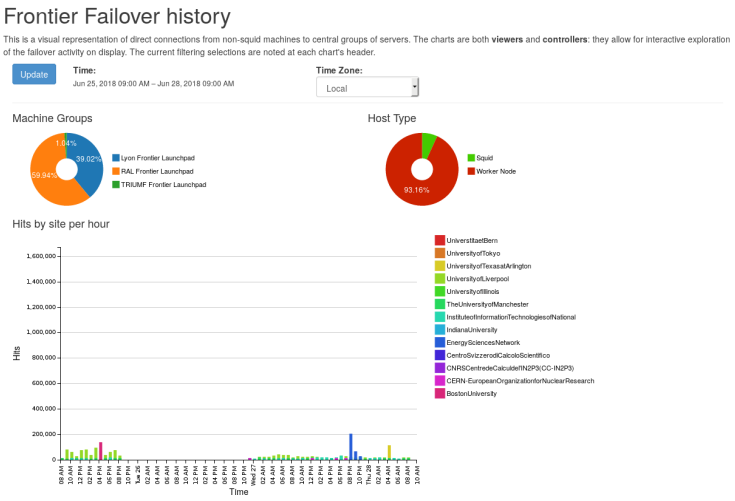

Figure 9. Failover monitoring page

A Python [10] script which prepares data for the monitoring page reads AWStats data of Frontier server Squids and backup proxies. It removes hosts that are known to be site Squids. The number of new hits for each host are calculated by subtracting the number seen during the previous run of the script from the current number. Hits from worker nodes are then summed up for every site and if the sum per site exceeds a predefined threshold, a list of hosts and their properties is exported into a file that is used by the monitoring system. There is also a possibility to send an email alert about such events.
The webpage displays

- pie chart of machine groups (Frontier server Squids and backup proxies)
- pie chart of host type (worker node or Squid)
- plot of hits per site each hour
- table with access details (host, whether it was made by a Squid proxy, timestamp, hits, bandwidth)

- table of email alarms (site, emails, timestamp)

JavaScript [11] functions allow filtering and display of additional information (when hovering on certain parts of tables or plots)

9 Conclusions

Monitoring of the ATLAS Frontier system is constantly evolving. There were monitoring tools available for a long time (AWStats, Maxthreads, Kibana availability info and SSB) which did not allow deeper understanding of the system. Several new tools were developed to improve the situation.
Monitoring of access of Conditions, Trigger, and Geometry database data is in place. An alert system informs experts in case of overload. Now, the incidents are quickly spotted and dealt with before they can destabilize the system for an extended period of time. Since this system became functional, there were no large scale job failures caused by overload of the Frontier system. Using informations from this system will allow deeper analyses of how ATLAS jobs are using conditions data. Such analyses will help to evolve the whole system for Run 3 and beyond.
New ATLAS MRTG monitoring page was developed. Its content is analysed and results are fed into the SSB which is checked by computing shifters. If a squid is failing, jobs on the site are connecting directly to Frontier servers or backup proxies. In case there would be too many of these, the Frontier system could overload. These connections from sites to Frontier servers and backup proxies are displayed by newly developed failover monitor where it can be followed by experts.

Copyright

References

[1] ATLAS Collaboration, JINST **3**, S08003 (2008)
[2] B.J. Blumenfeld, D. Dykstra, L. Lueking, E. Wicklund (CMS), J. Phys. Conf. Ser. **119**, 072007 (2008)
[3] D. Dykstra, L. Lueking, J. Phys. Conf. Ser. **219**, 072034 (2010)
[4] *Frontier page*, http://frontier.cern.ch
[5] *Open source, distributed, restful search engine*, https://github.com/elastic/elasticsearch
[6] *Analytics platform at university of chicago*, http://atlas-kibana.mwt2.org
[7] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay et al., *Jupyter Notebooks – a publishing format for reproducible computational workflows*, in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, edited by F. Loizides, B. Schmidt (IOS Press, 2016), pp. 87 – 90
[8] *Awstats project*, https://www.awstats.org/
[9] *The multi router traffic grapher*, https://oss.oetiker.ch/mrtg/doc/mrtg.en.html
[10] *Python project, "python" [software], version 2.6.9, 2013*, Available from https://www.python.org/download/releases/2.6.9/
[11] *The javascript project*, https://www.javascript.com/