MODERN SQL AND NOSQL DATABASE TECHNOLOGIES FOR THE ATLAS EXPERIMENT

D. Barberis[a]
on behalf of the ATLAS Collaboration

Physics Department of the University of Genova and INFN Sezione di Genova,
Via Dodecaneso 33, 16146 Genova, Italy

E-mail: [a] Dario.Barberis@ge.infn.it

Structured data storage technologies evolve very rapidly in the IT world. LHC experiments, and ATLAS in particular, try to select and use these technologies balancing the performance for a given set of use cases with the availability, ease of use and of getting support, and stability of the product. We definitely and definitively moved from the "one fits all" (or "all has to fit into one") paradigm to choosing the best solution for each group of data and for the applications that use these data. This paper describes the solutions in use, or under study, for the ATLAS experiment and their selection process and performance measurements.

Keywords: Scientific Computing, Databases, BigData, Hadoop

1. Introduction

When software developments started for ATLAS [1] and all Large Hadron Collider (LHC) experiments about 20 years ago, the generic word "database" practically referred only to relational databases, with only a few exceptions. There were very few options to store largish amounts of structured data: Oracle [2] was fully supported by CERN-IT including license costs, MySQL [3] was in its early stages, not scaling yet to the expected data volumes and rates but promising rather well, or one could build a new in-house system. So the choice was clear: fit everything into Oracle because of the CERN system-level support, and develop the ATLAS applications to make use of Oracle's tools for performance optimization. ATLAS hired two expert Oracle application developers who obviously helped a lot with application development and optimization.

Having only one underlying technology helped to provide a robust and performant central database service, managed jointly by CERN at the system level and ATLAS at the application level. Many time-critical applications are now hosted by the CERN Oracle infrastructure:

- The conditions database (COOL) [4]
- AMI (ATLAS Metadata Interface) [5] and COMA (Conditions Metadata) [6]
- ProdSys/PanDA (distributed production system) [7]
- Rucio (distributed data management system) [8]
- AGIS (Grid information system) [9]
- Glance (membership, authorship, speakers etc.) [10]

All these applications grew in size and complexity with time and are working quite well for the Collaboration's current usage; Oracle can be very fast if database schemas and queries are well designed and optimized.

On the other hand having only one underlying technology forced some applications that have no need of relational information into fixed schemas that may be not completely optimal; for example time-series measurements produced by DCS (Detector Control System) can be more simply represented by time-value pairs, and their data have to be compressed before storing in Oracle because of their huge sizes. In addition Oracle schemas have to be carefully designed upfront and are then hard to extend or modify, and data access to Oracle databases from Grid jobs was less than obvious and an interface system (Frontier [11]) had to be adopted to allow concurrent running of over 300k jobs. So when data analytics tools started appearing on the Open Source market that can deal with huge amounts of less structured data, ATLAS groups started evaluating them for their needs.

Towards the end of LHC Run1 in 2012 and during the shutdown period in 2013-2014 a number of new structured data storage solutions ("NoSQL Databases") were tested as back-end support systems for new applications, including Hadoop [12] and the many associated tools and data formats, Cassandra [13], MongoDB [14], etc. They are mostly key-value pair or column-oriented storage systems.

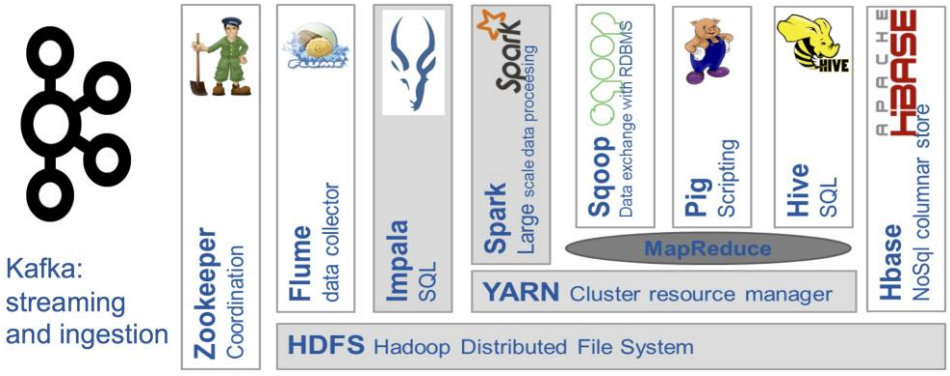

Figure 1. Hadoop ecosystem as provided by CERN-IT at the end of 2017

At the same time the Worldwide LHC Computing Grid (WLCG) Collaboration launched a few study groups on new computing technologies, one of which was the "Database Technical

Evolution Group" (DB TEG), which recommended that CERN deploy and support a Hadoop cluster for new applications, with all associated tools [15]. In fact several Hadoop clusters were set up over the years to avoid destructive interference between different applications, while both system managers and application developers were learning the best practices for application design and optimization. Figure shows the many tools provided currently by CERN-IT in the Hadoop ecosystem.

2. Database usage by ATLAS in LHC Run2

The database systems are used to support ATLAS data processing and analysis, as well as all other collaboration activities. One can identify three major groups of data:

- Conditions data. They are all non-event data that are useful to reconstruct events, such as detector hardware conditions (temperatures, currents, voltages, gas pressures and mixtures, etc), detector read-out conditions, detector calibrations and alignments, and physics calibrations. All conditions data have associated intervals of validity and (for derived data) versions. The COOL database is used for all conditions data.
- Physics metadata. This is information about datasets and data samples, provenance chains of processed data with links to production task configurations, cross-sections and configurations used for simulations, trigger and luminosity information for real and simulated data.
- Distributed computing data management and processing book-keeping. The distributed production/analysis and data management systems produce and need to store a wealth of metadata about the data that are processed and stored:
    - Rucio (Distributed Data Management) has a dataset contents catalogue (list of files, total size, ownership, provenance, lifetime, status etc.) a file catalogue (size, checksum, number of events), a dataset location catalogue (list of replicas for each dataset) and keeps information on the activities of data transfer tools, deletion tools and on storage resource status etc.
    - ProdSys/JEDI/PanDA (Distributed Workload Management) store lists of requested tasks and their input and output datasets, software versions, lists of jobs with status, running locations, lists of processing resources with their status etc.

  Both systems use a combination of quasi-static and rapidly changing information, as ATLAS runs over 1 million jobs/day using on average almost 300k job slots and moves 600 TB/day around the world. Oracle supports very well both systems if the tables and the load don't grow indefinitely; "old" information is automatically copied to an archive Oracle database and removed from the primary one.

2.1 Oracle storage

All this information is stored in three main Oracle RACs (Real Application Clusters), respectively for ATLAS online, offline, and distributed computing applications, plus an archive database, all with active stand-by replicas and back-ups. Selected users and processes have write access; all users have read access. Read access normally goes through front-end web services as direct access to Oracle from many processes could overload the servers: Frontier for access to conditions data from production and analysis jobs, the AMI and COMA front-end servers for access to metadata, and DDM and PanDA servers for access to dataset and production/analysis task information. Figure shows a sketch of the Oracle RACs and the data flow between them, including the replication to the active stand-by instances and the distribution of conditions data to IN2P3-CC in Lyon (France), RAL in Oxfordshire (United Kingdom) and TRIUMF in Vancouver (Canada).

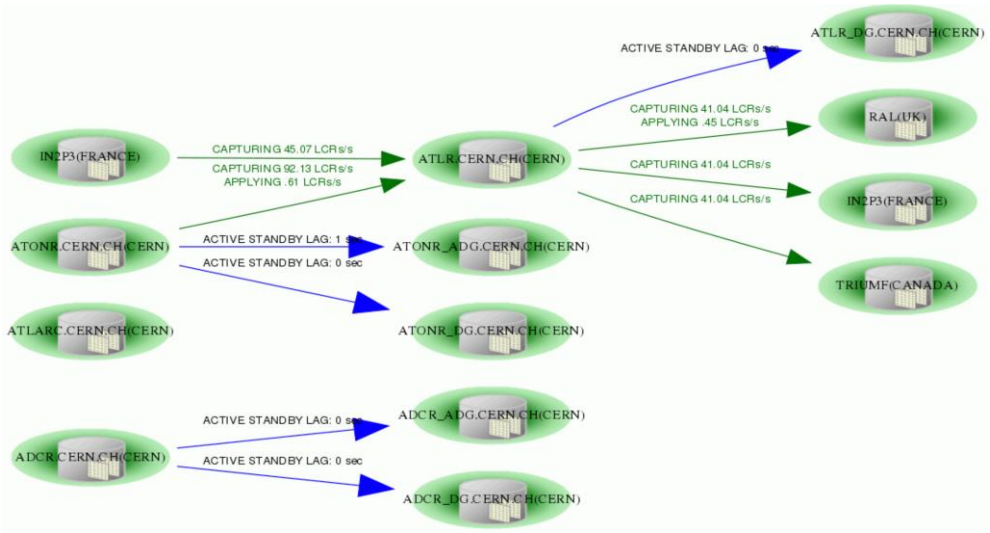

Figure 2. ATLAS Oracle databases and data flow between them

2.2 NoSQL storage

The main distributed computing applications (Rucio and ProdSys/PanDA) have a very high transaction rate and the Oracle database is very efficient in dealing with this large information flow. Applications such as monitoring and accounting, that only read from the database, are instead better suited for different storage systems, with needed data extracted from Oracle and formatted appropriately for the expected queries. Tasks to extract the relevant information from Oracle and store it in Hadoop run continuously and provide input to several other tools, including dataset popularity, task monitoring and data management accounting.

ElasticSearch [16] became popular in the last couple of years as a "quick" way to search information, and it is now used by several distributed computing analytics applications. The ElasticSearch storage needs filling with data extracted from logfiles or databases, and then interactive tools can be used to generate plots that are displayed with Kibana [17]. It is very useful for monitoring and to find out what is going on in case of unexpected failures, correlating information from different sources; for example, if a Frontier server becomes unresponsive, we can look up which jobs or tasks caused that, where they ran (or are running) and correlate it with the PanDA status of that site. As the ElasticSearch performance gets degraded if the amount of accumulated data becomes large and the hardware is not sufficient for the data size and the tasks to be performed, careful provisioning is needed (like for any other computing system!).

2.3 The first ATLAS NoSQL tool: EventIndex

The ATLAS EventIndex [18] is the first application that was entirely developed having in mind the usage of modern structured storage systems as back-end instead of a traditional relational database. The design started in late 2012 and the system was in production at the start of LHC Run2 in Spring 2015. The EventIndex is a system designed to be a complete catalogue of ATLAS events, with all real and simulated data. Its main use cases are event picking (give me this event in that format and processing version), counting and selecting events based on trigger decisions, production completeness and consistency checks (data corruption, missing and/or duplicated events) and trigger chain and derivation overlap counting. It contains event identifiers (run and event numbers, trigger stream, luminosity block, bunch crossing number), trigger decisions and references (GUID plus internal pointer) to the events at each processing stage in all permanent files generated by central productions.

The EventIndex has a partitioned architecture, following the data flow, sketched in Figure . The Data Production component extracts event metadata from files produced at Tier-0 or on the Grid, the Data Collection system [19] transfers EventIndex information from jobs to the central servers at CERN, the Data Storage units provide permanent storage for EventIndex data and fast access for the most common queries, plus finite-time response for complex queries. The full information is stored in Hadoop in MapFile format [20], with an internal catalogue in HBase [21] and also a copy to

HBase for event look-up; reduced information (only real data, no trigger) is copied to Oracle for faster queries [22]. A monitoring system keeps track of the health of servers and the data flow [23].

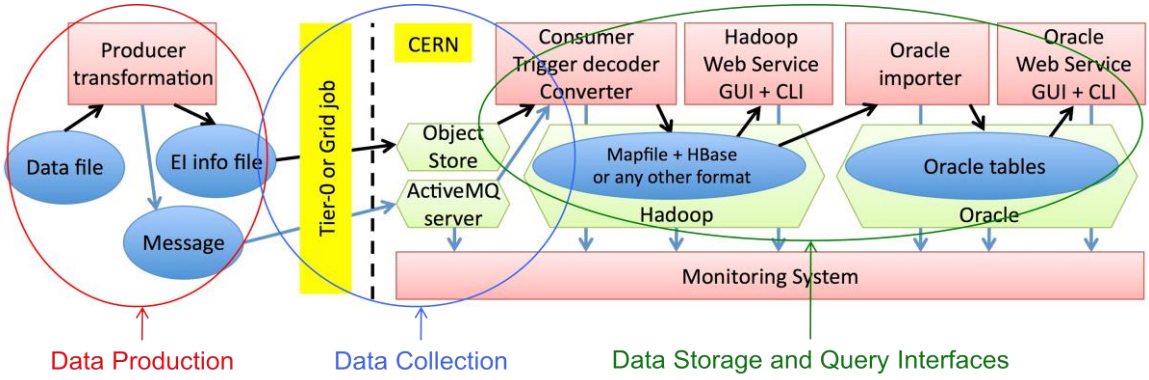

Figure 3. EventIndex architecture and data flow

At the time of writing the Hadoop system stores 120 TB of real data and 36 TB of simulated data, plus 154 TB of other data (input and transient data and archive). In Oracle we have over 100 billion event records, stored in a table of 2.2 TB with 2 TB of index space.

An active R&D programme to explore different, and possibly better performing, data store formats in Hadoop was started in 2016. The "Pure HBase" approach (database organized in columns of key-value pairs) was one of the original options in 2013, but did not work in 2015 because of the then poor performance of the CERN lxhadoop cluster (problem solved at the end of 2015); it is more promising now as it shows good performance for event picking. The Avro [24] and Parquet [25] data formats have been explored, with tests on full 2015 real data, and look promising (for different reasons). Kudu [26] is a new technology in the Hadoop ecosystem, implementing a new column-oriented storage layer that complements HDFS and HBase. It appears to be more flexible to address a wider variety of use cases, in particular as it is addressable also through SQL queries, placing it midway between Oracle and the NoSQL world; tests are continuing this year in view of a possible use in production in 2018 [27].

3. Evolution of Databases for Run3

The continued usage of Oracle is fine for the time being but we were warned by CERN that the license conditions may change in the future, so some kind of diversification may be needed. Some types of data and metadata fit naturally into the relational database model, but other data much less, for example the large amounts of useful but static data on DDM datasets for accounting, or information on completed PanDA production and analysis tasks, event metadata and so on.

As long as access to the data is done through an interface server, the user won't actually see the underlying storage technology. In this way it is possible to keep only the "live" data in Oracle and move the rest to different technologies. This also means that at some point in the future we could change technology for the SQL database without too much trouble.

3.1 A new Conditions Data Service for Run3

CREST [28] is a new architecture for conditions data services for HEP experiments, developed initially by CMS and ATLAS, and now considered by a number of other experiments. It is based on the relational schema simplification introduced by CMS for Run2, with data identified by type, interval of validity and version, and payload data in BLOBs. It will contain in its schema only data used for event processing (no dump of raw information). The functions are partitioned: the relational database is used only for payload data identification, but the payload can be anywhere, including files in CVMFS [29]. A web server (with an internal cache) is used for interactions with the relational database and data input, search and retrieval, and Frontier servers and Squids provide access from Grid jobs and local caches. Figure shows a scheme of the component architecture and data access paths.

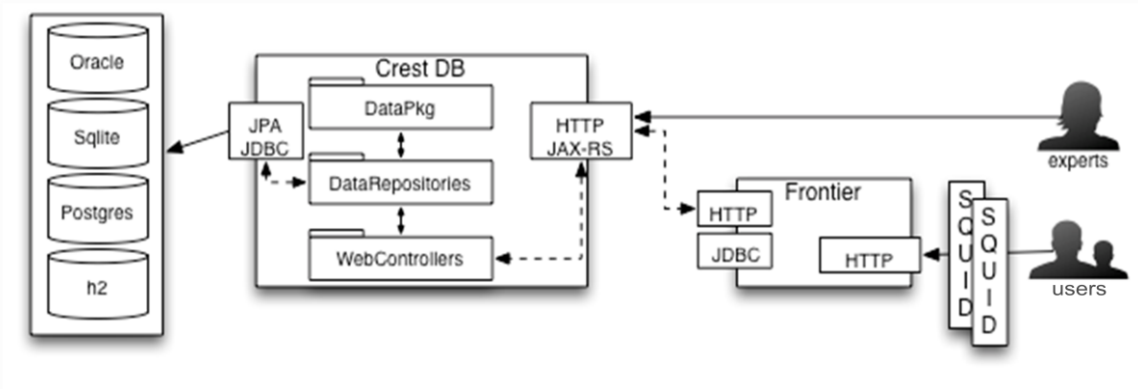

Figure 4. CREST architecture and data access

The CREST system for ATLAS is under active development and will be in production for the start of LHC Run3. By that time all existing conditions for Run1 and Run2 will have to be transferred to the new system, to allow processing and analysis of all ATLAS data with the most recent software suites.

3.2 Time series databases

Time series are for example streams of DCS (Detector Control System) data, where for each data type a raw data record consists of a time stamp and one or a few values. This information is currently stored in Oracle using COOL, after averaging over short time periods, or storing new values only when sufficiently different from previous ones. Data sizes can become enormous compared to other data types, so much that direct use of this information in reconstruction jobs is not a good idea; it is much better to store this information in a system that is designed for time series and has useful tools for averaging over predefined time intervals, threshold detection, and an integrated display of the values as a function of time.

CERN-IT decided to use InfluxDB [30] coupled with Grafana [31] initially for their internal system monitoring and then also for the monitoring of WLCG site status and experiment distributed computing tools. As they seem happy with it, ATLAS started evaluations in the online and offline context, including displaying the time series with Grafana. An example of data extracted from the PanDA database in Oracle, stored as time series in InfluxDB and displayed with Grafana is shown in Figure .

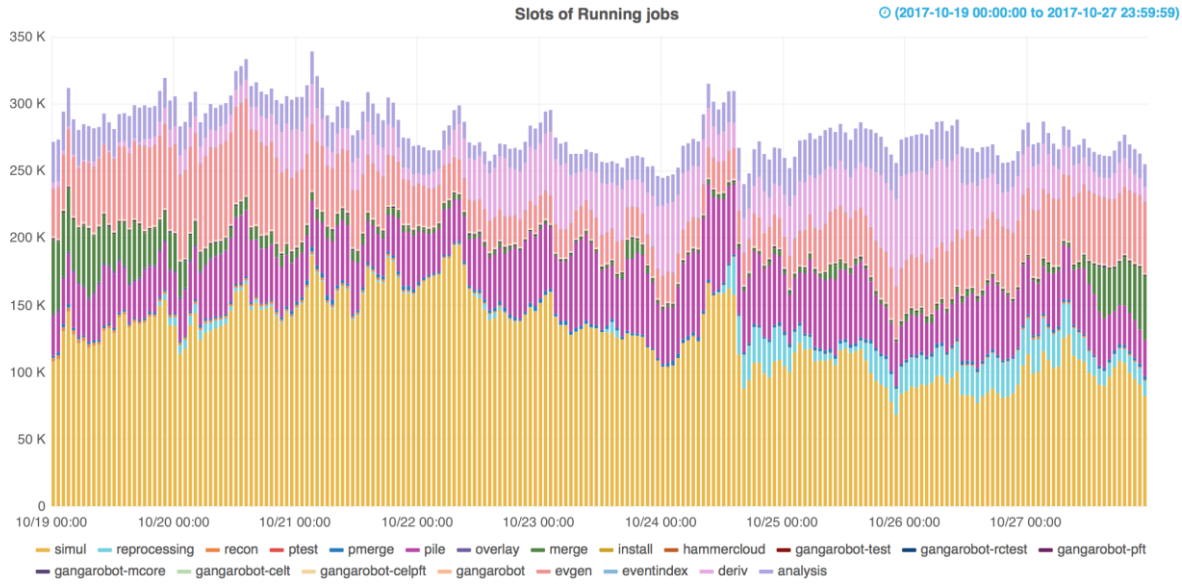

Figure 5. Display with Grafana of the time series of the number of CPU cores used on the Grid by ATLAS jobs between 19 and 27 October 2017

3.3 Metadata for Run3 and beyond

A new effort to revise and harmonize metadata information and its storage and retrieval tools started this year in ATLAS: the DCC (Data Characterization and Curation) project. It has three complementary approaches, respectively on the overall architecture, a top-down approach for dataset related metadata and dataset discovery based on the Data Knowledge Base, and a bottom-up approach to event metadata based on the concepts of the Event WhiteBoard and Virtual Datasets.

The Event WhiteBoard (EWB) project has been launched recently. It is an evolution of the EventIndex concept, but with an event-oriented architecture, whereas the EventIndex has a dataset-oriented internal storage organization. It will have one and only one logical record per event, containing event identification and immutable information (trigger, luminosity block etc.), but then for each processing step involving each event it will have the link to the algorithm producing it (processing task configuration), pointers to outputs and flags for offline selections (derivations). An important new feature of the EWB is the possibility for automatic processes and single users to annotate single event records, adding key-value pairs (or similar formats) that can then be interpreted and used for automatic selections or other actions. Figure shows the EWB component architecture and the flow of data into and out of it.

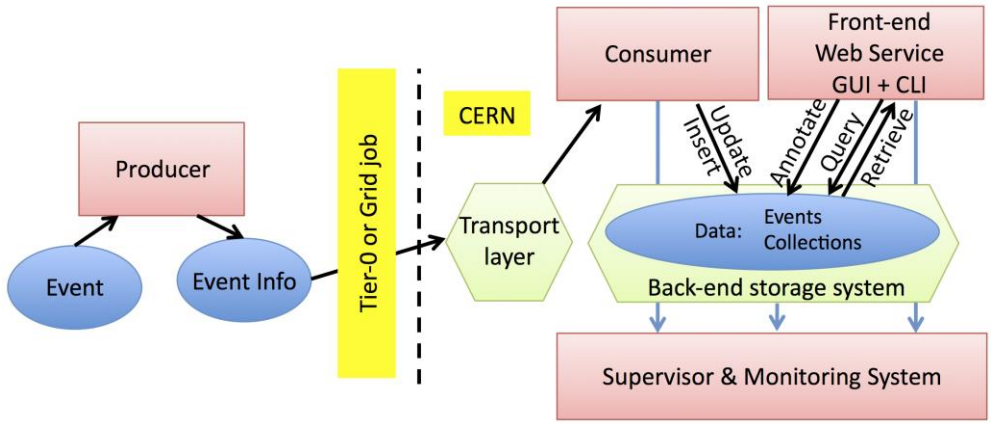

Figure 6. Event WhiteBoard architecture and data flow

The problem is not intrinsically difficult but the EWB will have to support 10 billion new real and 35 billion simulated events per year for Run2, a factor of 3 more for Run3 and another factor of 3 more for Run4. Work on the technology selection is starting now, with the aim to have a prototype working at the Run2 scale by the end of 2018 that will be promising to scale as required for Run3, and the new EWB in operation during 2020.

Virtual Datasets (VDS) are not a new idea but with the new EWB technology it should be possible to implement them. A VDS is a list of events that satisfy a number of conditions as contained in the EWB. For example, the derivation step after reconstruction now writes out O(100) streams with selected events; even if the events are "slimmed", the amount of required disk space is large. With VDSs, it will be enough to flag the selected events in the EWB, saving lots of storage, and user analysis jobs will then read only those events.

4. Conclusions

ATLAS is always following technology developments in the database and structured data storage fields. The lifetime of ATLAS computing tools and infrastructure is much longer than the active lifetime of many open source products, and this fact poses very strong constraints on product selection. In any case we need to continue the R&D programs to make the best use of new upcoming computing technologies, without neglecting ongoing operations of course.

Continued collaboration with CERN-IT is essential for providing well-performing and robust services to the Collaboration.

The tool that is invisible to most users is the one that works without problems all the time!

References

[1] ATLAS Collaboration 2008 The ATLAS Experiment at the CERN Large Hadron Collider, *JINST* **3** S08003 doi:10.1088/1748-0221/3/08/S08003
[2] Oracle: https://www.oracle.com
[3] MySQL: https://www.mysql.com
[4] Valassi, A. *et al.* 2008 COOL, LCG Conditions Database for the LHC Experiments: Development and Deployment Status, CERN-IT-Note-2008-019 and NSS 2008 Proceedings of the Medical Imaging Conference, Dresden, Germany
[5] Albrand S. 2010 The ATLAS metadata interface, *J. Phys. Conf. Ser.* 219 042030, doi:10.1088/1742-6596/219/4/042030
[6] Gallas E J *et al* 2014 Utility of collecting metadata to manage a large scale conditions database in ATLAS, *J. Phys.: Conf. Ser.* 513 042020, doi:10.1088/1742-6596/513/4/042020
[7] Maeno T *et al.* 2014 Evolution of the ATLAS PanDA workload management system for exascale computational science, *J. Phys. Conf. Ser.* 513 032062 doi:10.1088/1742-6596/513/3/032062
[8] Garonne V *et al.* 2014 Rucio – The next generation of large scale distributed system for ATLAS Data Management, *J. Phys. Conf. Ser.* 513 042021 doi:10.1088/1742-6596/513/4/042021
[9] Anisenkov A. *et al*. 2011 ATLAS Grid Information System, *J. Phys.: Conf. Ser.* 331 072002, doi:10.1088/1742-6596/331/7/072002
[10] Grael F F *et al.* 2011 Glance Information System for ATLAS Management, *J. Phys.: Conf. Ser.* **331** 082004, doi:10.1088/1742-6596/331/8/082004
[11] Barberis D *et al* 2012 Evolution of grid-wide access to database resident information in ATLAS using Frontier, *J. Phys.: Conf. Ser.* 396 052025, doi:10.1088/1742-6596/396/5/052025
[12] Hadoop and associated tools: http://hadoop.apache.org
[13] Cassandra: http://cassandra.apache.org
[14] MongoDB: https://www.mongodb.com
[15] Barberis D *et al.* 2012 Report of the WLCG Database Technical Evolution Group, CERN report https://espace.cern.ch/WLCG-document-repository/Technical_Documents/ Technical Evolution Strategy/TEG Reports - April 2012/DB_TEG_report_2.0.pdf
[16] ElasticSearch: https://www.elastic.co
[17] Kibana: https://www.elastic.co/products/kibana
[18] Barberis D *et al* 2015 The ATLAS EventIndex: architecture, design choices, deployment and first operation experience, *J. Phys.: Conf. Ser.* 664 042003, doi:10.1088/1742-6596/664/4/042003
[19] Sánchez J *et al* 2015 Distributed Data Collection for the ATLAS EventIndex, *J. Phys.: Conf. Ser.* 664 042046, doi:10.1088/1742-6596/664/4/042046
[20] Favareto A *et al.* 2016 Use of the Hadoop structured storage tools for the ATLAS EventIndex event catalogue, *Phys. Part. Nuclei Lett.* 13: 621, https://doi.org/10.1134/S1547477116050198
[21] HBase: https://hbase.apache.org
[22] Gallas E J *et al.* 2017 An Oracle-based Event Index for ATLAS, http://cds.cern.ch/record/2252389/files/ATLAS-COM-CONF-2017-011.pdf, to be published in *Proceedings of CHEP 2016*, San Francisco, USA, October 2016
[23] Barberis D *et al*. 2016 ATLAS EventIndex monitoring system using the Kibana analytics and visualization platform, *J. Phys.: Conf. Ser.* 762 012004, doi:10.1088/1742-6596/762/1/012004
[24] Avro: https://avro.apache.org
[25] Parquet: http://parquet.apache.org
[26] Kudu: http://kudu.apache.org
[27] Baranowski Z *et al*. 2017 A study of data representation in Hadoop to optimise data storage and search performance for the ATLAS EventIndex, http://cds.cern.ch/record/2244442/files/ATL-SOFT-PROC-2017-043.pdf, to be published in *Proceedings of CHEP 2016*, San Francisco, USA, October 2016
[28] Barberis D *et al*. 2015 Designing a future Conditions Database based on LHC experience, *J. Phys.: Conf. Ser. 664* 042015, doi:10.1088/1742-6596/664/4/042015
[29] CVMFS: https://cernvm.cern.ch/portal/filesystem
[30] InfluxDB: https://www.influxdata.com
[31] Grafana: https://grafana.com