The African Review of Physics (2018) **13**: 0009

Proceedings of the first African Conference on Fundamental Physics and Applications 2018, Namibia.
Guest Editors: K. A. Assamagan, M. Backes, D. Charlton, S. Muanza , D. Sahu, and D. Singh

Modern BigData technologies to store and access metadata for the ATLAS experiment

Dario Barberis
University of Genova and INFN Sezione di Genova, Genova, Italy
On behalf of the ATLAS Collaboration

Structured data storage technologies evolve very rapidly in the IT world, driven by BigData projects. LHC experiments, and ATLAS in particular, select and use these technologies to store a wealth of metadata, 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. Our community 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 or metadata 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.

1. Introduction

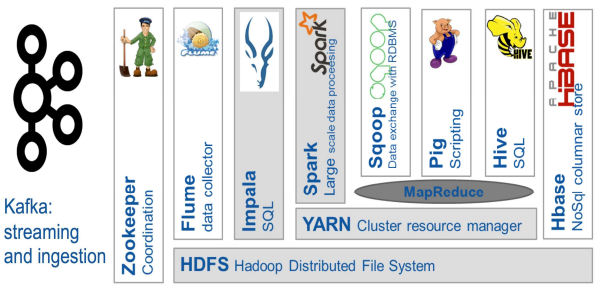
When software developments started for the ATLAS experiment [1] at the Large Hadron Collider (LHC) and all other similar experiments about 20 years ago, the generic word "database" practically referred only to relational databases, with very few exceptions. There were not many options to store large 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 alternatively one could build a new in-house system. The choice was then 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 evidently 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 and manage such diverse information as detector conditions and calibrations, physics dataset metadata, composition of the Collaboration, or location and status of datasets and data processing tasks in the ATLAS distributed computing system. 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 jobs running all over the world was less than obvious already at the time of first LHC operations in 2008. For this reason an interface system (Frontier [4], developed initially by Fermilab for the CDF and CMS Collaborations) had to be adopted and adapted to allow concurrent running of over 300k jobs worldwide. When on the Open Source market new data analytics tools started appearing 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 [5] and the many associated tools and data formats, Cassandra [6], MongoDB [7], etc. They are mostly key-value pair or column-oriented storage systems.

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 [8]. 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 1 shows the many tools provided currently by CERN-IT in the Hadoop ecosystem.



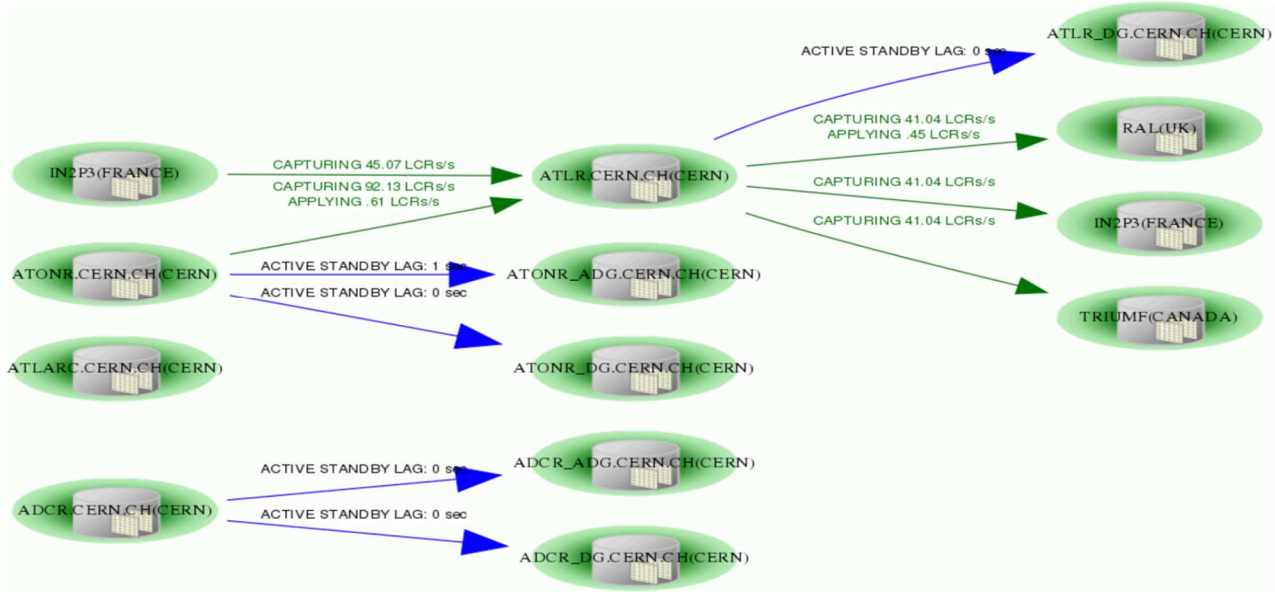


Figure 2: ATLAS Oracle databases and data flow between them.

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

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 [9] 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. The AMI (ATLAS Metadata Interface, [10]) and COMA (COnditions MetadatA, [11]) databases hold and provided all this information in a coherent way.
- Distributed computing data management and processing configurations and 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, [13]) 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, [12]) 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.
  - AGIS (ATLAS Grid Information System, [14]) contains the configurations of the ~140 computing centres that support ATLAS operations around the world and the available resources at each site.

These systems use a combination of quasi-static and rapidly changing information, as ATLAS runs several million jobs/day using on average almost 300k job slots and moves 600 TB/day around the world. Oracle supports these operations very well 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.

The Glance database [15], which holds information on the Collaboration composition, roles, publications and author lists, is also hosted by the Oracle cluster. This is a much smaller application but has very strict access requirements and fits rather well in a relational database like Oracle.

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 ATLAS members 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 Rucio and PanDA servers for access to dataset and production/analysis task information. Figure 2 shows a sketch of the Oracle RACs and the data flow between them, including the replication to the active stand-by instances.

To improve the access speeds and reliability, the conditions data are also distributed to three external computing centres supporting ATLAS, namely IN2P3-CC in Lyon (France), RAL in Oxfordshire (United Kingdom) and TRIUMF in Vancouver (Canada). The Frontier system takes care of directing the queries from running jobs to the most appropriate server and provides failover capabilities in case of network or site problems.

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 that were developed to monitor and optimise the usage of computing resources, including task monitoring, data management accounting and dataset popularity (how often each dataset is accessed, therefore how many replicas are needed).

ElasticSearch [16], a BigData system to store and retrieve large amounts of simply formatted data records, 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 graphically 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

ATLAS and the other LHC experiments record several billions events every year, and produce an equivalent number of simulated events. Event records are stored into files of a few GB size that contain a few thousand events each, depending on the event format. At the time of writing the complete ATLAS data storage amounts to about 150 PB of disk space and an equivalent amount of tapes for data archival; the total amount of LHC data is quickly approaching 1 exabyte, making it the largest non-commercial dataset so far. Considering that each event is processed more than once and is stored in several formats, from raw data to complete reconstruction records and then highly skimmed and compressed analysis formats, it is important to have a fast and reliable tool to keep track of the physical location of each event at each processing stage, and to have the possibility to select events on the basis of a small number of metadata items; something like the index in a library. The ATLAS EventIndex [18] 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

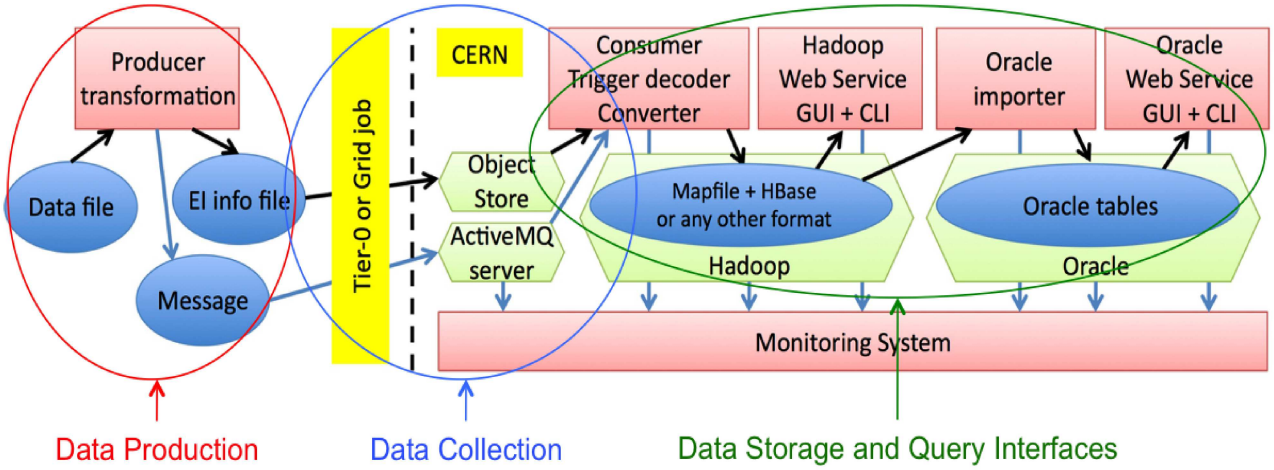


Figure 3: Event Index architecture and data flow.

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 (file identifier and internal pointer) to the events at each processing stage in all permanent files generated by central productions.

The EventIndex 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 has a partitioned architecture following the data flow, sketched in Figure 3. The Data Production component extracts event metadata from files produced on ATLAS resources at CERN or world-wide, the Data Collection system [19] transfers EventIndex information 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 compressed MapFile format [20], with an internal catalogue in HBase [21] (the relational database of the Hadoop system) 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 the servers and the data flow [22].

At the time of writing the Hadoop system stores almost 200 billion event records, using 25 TB for real data and 5 TB for simulated data, plus the auxiliary data (input and transient data and archive). In Oracle we have over 150 billion event records, stored in a table of 2.7 TB with 2.4 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 on the table in 2013, but was not selected because of its poor parallelism that made the performance degrade when data volumes increase; it is more promising now as it shows good performance for event picking but not for all other use cases. The Avro [24] and Parquet [25] data formats have been explored, with tests on full 2015 real data, and look promising, albeit 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. Kudu 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 2019 [27].

3. Evolution of Databases for LHC 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 Rucio 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 always done through an interface server, the user won't actually see the underlying data 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 together by the CMS and ATLAS Collaborations, 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 the actual payload data in BLOBs (Binary Large OBjects). It will contain in its schema only data used for event processing, such as detector calibrations and alignments as function of time.

The functions are thus partitioned: the relational database is used only for payload data identification, but the payload can be anywhere, including files in CVMFS [29], the CERN-developed distributed file system that can be accessed from any computer connected to the Internet. 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 4 shows a scheme of the component architecture and data access paths.

The CREST system for ATLAS is under active development and will be in production for the start of LHC Run3 in 2021. 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.

.

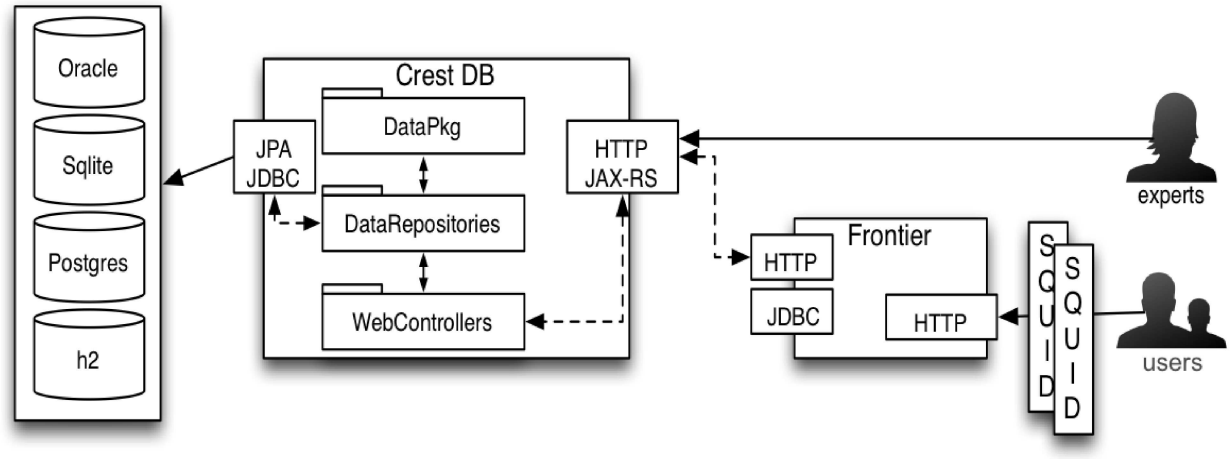


Figure 4: CREST architecture and data access.

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 discouraged; 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.

The CERN Information Technology group decided in 2017 to use the time-series database InfluxDB [30] coupled with Grafana [31], initially for their internal system monitoring and then also for the monitoring of the status of distributed computing sites and experiment distributed computing tools. As they seemed to be 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 5.

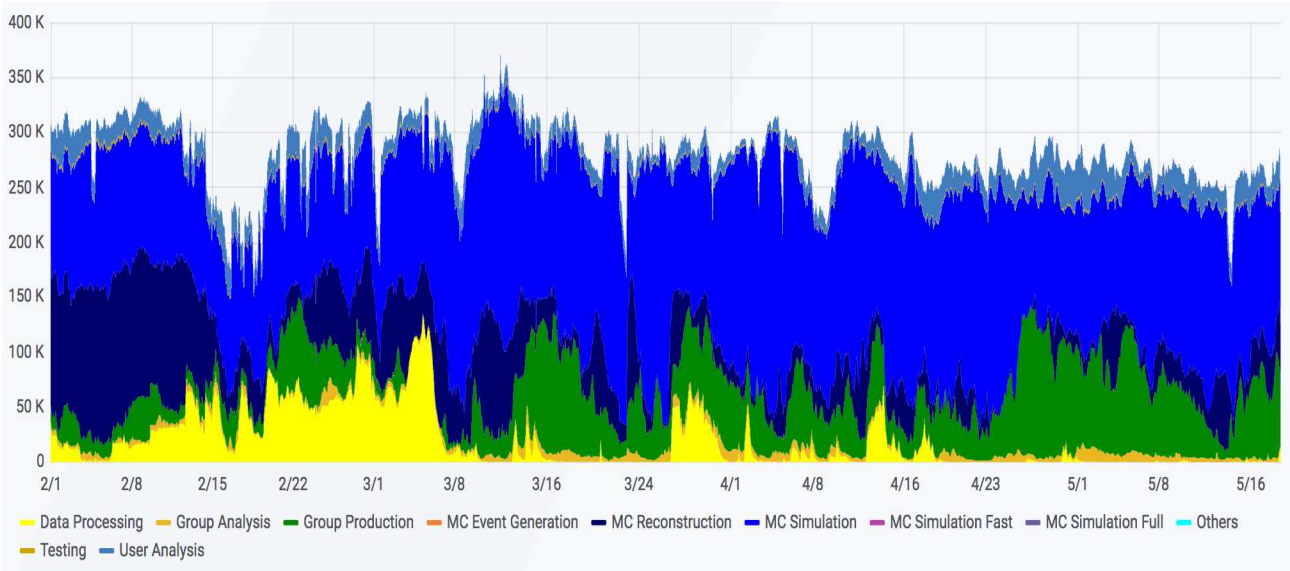


Figure 5: Display with Grafana of the time series of the number of CPU cores used on the Grid by ATLAS jobs between 1st February and 18th May 2018. The colours correspond to different computing activities.

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 the CERN Information Technology department 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] 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

[5] Hadoop and associated tools: http://hadoop.apache.org

[6] Cassandra: http://cassandra.apache.org

[7] MongoDB: https://www.mongodb.com

[8] 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

[9] 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

[10] Albrand S. 2010 The ATLAS metadata interface, *J. Phys. Conf. Ser.* 219 042030, doi:10.1088/1742-6596/219/4/042030

[11] 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

[12] 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

[13] 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

[14] Anisenkov A. *et al.* 2011 ATLAS Grid Information System, *J. Phys.: Conf. Ser.* **331** 072002, doi:10.1088/1742-6596/331/7/072002

[15] 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

[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, doi:10.1134/S1547477116050198

[21] HBase: https://hbase.apache.org

[22] Gallas E J *et al.* 2017 An Oracle-based Event Index for ATLAS, *J. Phys.: Conf. Ser.* 898 042033, doi:10.1088/1742-6596/898/4/042033

[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, *J. Phys.: Conf. Ser.* 898 062020, doi:10.1088/1742-6596/898/6/062020

[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

Received: 11 October, 2018

Accepted: 20 October, 2018