Journal of Physics: Conference Series **513** (2014) 042002 doi:10.1088/1742-6596/513/4/042002

The ATLAS EventIndex: an event catalogue for experiments collecting large amounts of data

D. Barberis[1*], J. Cranshaw[2], G. Dimitrov[3], A. Favareto[1], Á. Fernández Casaní[4], S. González de la Hoz[4], J. Hřivnáč[5], D. Malon[2], M. Nowak[6], J. Salt Cairols[4], J. Sánchez[4], R. Sorokoletov[7], Q. Zhang[2], on behalf of the ATLAS Collaboration

[1] Università di Genova and INFN, Genova, Italy
[2] Argonne National Laboratory, Argonne, IL, United States
[3] CERN, Geneva, Switzerland
[4] Instituto de Fisica Corpuscular (IFIC), University of Valencia and CSIC, Valencia, Spain
[5] LAL, Université Paris-Sud and CNRS/IN2P3, Orsay, France
[6] Brookhaven National Laboratory, Upton, NY, United States
[7] University of Texas at Arlington, Arlington, TX, United States

*Corresponding author: Dario.Barberis@cern.ch

Abstract. Modern scientific experiments collect vast amounts of data that must be catalogued to meet multiple use cases and search criteria. In particular, high-energy physics experiments currently in operation produce several billion events per year. A database with the references to the files including each event in every stage of processing is necessary in order to retrieve the selected events from data storage systems. The ATLAS EventIndex project is studying the best way to store the necessary information using modern data storage technologies (Hadoop, HBase etc.) that allow saving in memory key-value pairs and select the best tools to support this application from the point of view of performance, robustness and ease of use. This paper describes the initial design and performance tests and the project evolution towards deployment and operation during 2014.

1. Introduction

The ATLAS experiment has a catalogue of all real and simulated events in a database (TAGDB). The TAGDB is implemented in Oracle, as that was the only proven technology that could hold the impressive amount of expected data when this project started in the early 2000s before the start of LHC operations [1-4]. The main purposes of this database are for the selection of events on the basis of physical variables, for the identification of files that contain a list of events already selected by other means, and to check that the same event does not mistakenly appear in more than one file.

In TAGDB each event is recorded several times, once for each reconstruction cycle, and contains three main data blocks: information identifying the event (event and run number and conditions satisfied by the online trigger), information allowing the location of the event data in files that contain it (GUIDs (Global Unique IDentifiers) of the files with the event in RAW (original), ESD (complete result of the reconstruction) and AOD (compactified result of reconstruction)), and a few physical

variables of the event, such as the number of reconstructed particles, their identification and properties, which may be useful to select events for specific analyses.

The TAGDB, with all related services and user interfaces, works acceptably well for the retrieval of event information for limited numbers of events and for the technical checks described above; the event selection use case turned out not to be interesting for physics analysis. Nevertheless the enormous amount of data (1 kB per record, which turns into several tens of TB in Oracle for all data collected by ATLAS so far) makes the implementation in Oracle particularly labour-intensive and expensive [5]. The number of simulated and real events amounted to about 8 billion in 2012 and will increase even faster with the resumption of the operation of the LHC accelerator in 2015 after the planned improvements in terms of energy and luminosity [6].

As it was necessary to redesign the overall system, we tried to avoid the intrinsic schema limitations of relational database systems that hindered development of the TAGDB in the past and considered the data storage technologies that were developed in the academic and commercial computing world in recent years [8]. The technologies of "NoSQL" databases are used by large commercial information companies and are well adapted to this type of applications [9]. At the same time, one can separate the parts of the project that can be of interest to other generic scientific applications from the parts closely related to ATLAS, and create a system that will be potentially useful for several other scientific and technical applications.

2. The EventIndex project

This project consists in the development and deployment of a catalogue of events for experiments with large amounts of data, such as those currently taking data with the LHC accelerator at CERN. The ultimate goal is to have available for HEP experiments, and make available to other users, a consistent set of software packages that allow fast and efficient searching of events of interest among the billions of events recorded in millions of files scattered in a world-wide distributed computing system.

The project starts from the knowledge accumulated over the years with the development and operation of the ATLAS TAGDB [1-7]. The work has been divided into several stages:

- Studies of the patterns of data storage in terms of functionality, scalability and sustainability of the various types of NoSQL databases currently available; local tests of functionality and distributed test of scalability; system architecture design of both the database and the auxiliary services to enter the data and facilitate the data mining and extraction.
- Implementation of the catalogue in the selected technology; population of the database with Run1 ATLAS events; complete functional tests of the scalability of databases containing billions of records; comparison of performance with the ATLAS TAGDB. Adaptation and consolidation of services, including the automatic loading of data from each phase of reconstruction and event selection.
- Definition of public interfaces to load, search and retrieve the data; packaging and release of the produced software so that it can be used by other scientific communities.

The most innovative part of this project is the adaptation and application of the NoSQL technology for the cataloguing of data of a large experiment. ATLAS alone accumulated in 2011-2012 two billion real and six billion simulated events per year. If the production of events were uniform in time, this would correspond to a rate of creation of the records in TAGDB of about 100 Hz. In reality, each event is processed several times, so that in TAGDB the number of records is considerably larger.

Each reconstruction campaign produces new versions of every event, in different formats (for ATLAS, ESD, AOD and several versions of ROOT n-tuples). For each reconstruction campaign and for each format one can generate a key-value pair (where the value is always the navigational information, *i.e.* the identifier of the file that contains the event in question and the internal structured pointer) and add it to the original record. The result is that an event will always correspond to one and only one logical record in the database that contains the entire processing history of the event.

Two main applications must be developed simultaneously to the back-end database: the infrastructure to populate the database and the tools to search for the records and extract the identifiers of the files that contain the events of interest. Because of the large amount of data being continuously processed, care must be taken in the development and optimization of these applications. The whole transmission chain of this information must be optimized in order to sustain rates of the order of 100 kHz. These aspects of performance and scalability are extremely important to have a product that will be useful to the scientific community.

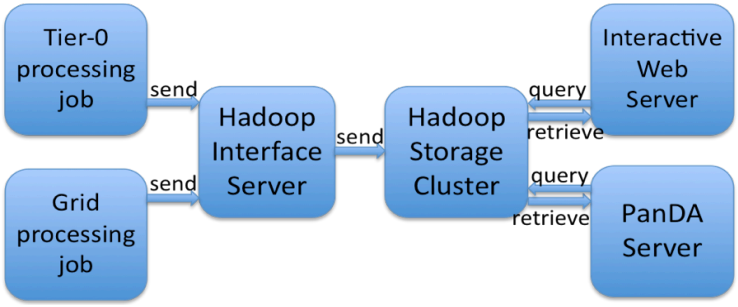

Figure 1: Block diagram of the EventIndex data flow. Information is produced by jobs running on the Tier-0 or on the Grid, inserted into the database, and retrieved by interactive and batch processes.

The major use cases the EventIndex project is developed for are:

- Event picking: give me the reference (pointer) to "this" event in "that" format for a given processing cycle.
- Production consistency checks: technical checks that processing cycles are complete (event counts match).
- Event service: give me the references for this list of events, to be distributed individually to processes running on (for example) HPC and/or cloud clusters.

3. Core architecture

NoSQL technologies offer several advantages over Oracle for applications like the EventIndex, as they scale naturally with the amount of data, there are many external tools to optimize data storage and data searches, they use clusters of low-cost Linux servers, including disks, and finally they belong to the world of open source software, so there is no license for the installation and use of these products.

Using NoSQL technologies it is possible to store information for each event in a single logical record. The record is created upon recording of the event from the online system, with initially only limited information, such as event number, run number, time stamp, luminosity block number, which trigger selected the event, and the identifier of the file that contains the event in RAW format. ATLAS uses the GUID (Global Unique IDentifiers) as identifiers of the files but any other system can be used, provided it is capable of uniquely associating identifiers and files. This information can be stored as key-value pairs in NoSQL systems, together with the internal file navigation information.

As CERN-IT is providing a prototype Hadoop [9] service, and Hadoop and the many associated tools seem to be a good solution for the EventIndex, we started testing the performance of solutions based on plain HDFS (the Hadoop file system) and/or HBase [10] (the Hadoop database system), using different data formats. For these initial tests we imported 1 TB of data from the TAGDB, which correspond to the first processing of the full real data collected in 2011.

Hadoop can be used to store database-like information in a very flexible way. There is no need to define a schema up front as long as data are stored as key-value pairs; keys can be changed if needed at any point in time (however we should design the storage with respect to performance and access patterns). HBase can be used to structure the data store and search/retrieve the data. The only important choice to be done up front is the separation of "immutable" quantities (such as event and run

number, luminosity block, trigger stream and pattern) from the quantities that depend on the processing cycle or step (references to files containing the given event in different formats, any physics variables). These two classes of information can (and should) be stored separately in different "column families".

For the initial tests, TAG events and links tables were imported from the Oracle database to the Hadoop cluster in CSV format. The sequence of the comma-separated columns is exactly the same as in Oracle; additionally, the tables with lists of collections and descriptions of collection fields were imported in the same place. Every collection was then imported into the HBase test table. The table has two column families, each of which contains one value in text format that includes comma-separated fields as in the source CSV files. The row key is a simple string with format "event number - run number - table number", where the table number is the number from the TAG data table name. This combination is unique for every event; the event number was placed in the first position of the row key because the event number selectivity is much higher than for the other fields.

A prototype application has been developed to convert data stored in Hadoop in CSV format into formats more suitable for querying. Several file formats are supported (from simple text format to full binary map format). Data can be also partitioned vertically (per attribute group) or horizontally (per collection of tags). Transformed data can be then further processed by creating new indexes (including inverted indexes) for faster access and by adding new attributes to existing collections. The created collections can be then analysed in several ways:

- using Hadoop maps (with natural indices kept in memory) to get access to concrete data;
- using additional indices to get already pre-selected data;
- using Map/Reduce jobs to search for data satisfying certain query;
- using full scan reads to perform detailed analyses and selections.

The preliminary performance benchmarking indicates so far that the most efficient way would be to store data in Hadoop map files and additional indices, with data in simple compressed text file on disk and indices kept in memory. Some of the indices would be created in the later stages, storing effectively results of query jobs and serving as a kind of cache. An efficient collection catalog should be created to keep track of the existing collections and indices, and their status. Figure 2 shows a schematic view of this possible architecture.

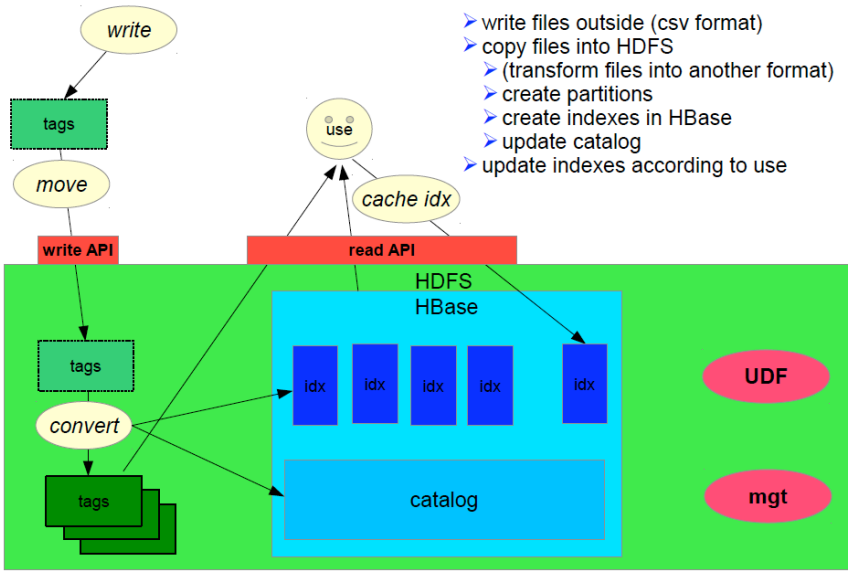

Figure 2: Schematic view of data organization in Hadoop/HBase.

4. Data collection and upload

Data to be stored in the EventIndex are produced by all production jobs that run on Tier-0 or the Grid. For every permanent output file a snippet of information, containing the file unique identifier and for each event the relevant attributes, has to be sent to the central server at CERN. The estimated rate (in May 2013, during the LHC shutdown) would be ~20 Hz of file records containing ~7 kHz of event records. During data-taking periods these numbers should be doubled; as both the data-taking rate and the Grid processing power are expected to increase by perhaps a factor two by 2015, the system should be ready to accept over 80 Hz of file records and insert into the back-end over 30 kHz of event records (plus contingency).

The architecture prototype is based on producers of the EventIndex on the worker nodes, and asynchronous consumers that effectively insert the data in the Hadoop database. Producers are running in the pilot of the worker nodes; the data is transmitted with a messaging system (open source options are studied, such as ActiveMQ [11]) to the queue in one of the endpoints. We can potentially have replication of the message queues in other sites, to permit duplication and high availability. The consumers are asynchronous, continuously consuming data. The task of the consumers is to get new data, and check it is valid with the production system. The final step is to inform the production system of the successful insertion, and remove data from the queue. Figure 3 shows a schematic view of the data collection system.

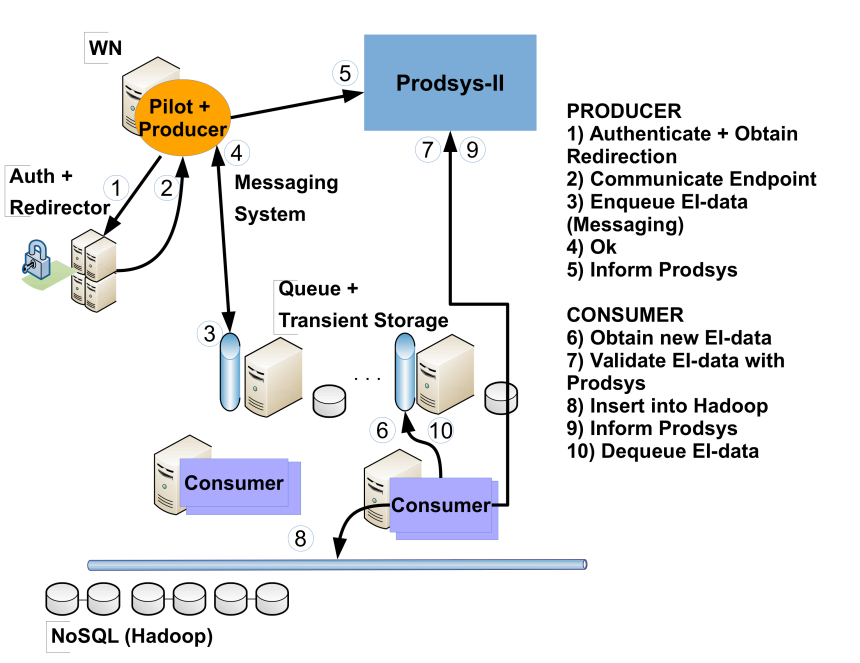

Figure 3: Schematic view of the data collection system.

5. Outlook

The EventIndex project is on track to provide ATLAS with a global event catalogue in advance of the resumption of LHC operations in 2015. The global architecture is defined and prototyping work is in progress; preliminary results are encouraging. Next steps consist in the full implementation of the final prototype, loaded with Run1 data, by the end of 2013 and the completion of deployment and operation infrastructure during 2014.

References

[1] Cranshaw J *et al*. 2008 Building a scalable event-level metadata service for ATLAS *J. Phys. Conf. Ser.* **119**:072012
[2] Cranshaw J *et al*. 2008 Integration of the ATLAS tag database with data management and analysis components *J. Phys. Conf. Ser.* **119**:042008
[3] Cranshaw J *et al*. 2008 A data skimming service for locally resident analysis data *J. Phys. Conf. Ser.* **119**:072011
[4] Mambelli M *et al*. 2010 Job optimization in ATLAS TAG-based distributed analysis *J. Phys. Conf. Ser.* **219**:072042
[5] Viegas F *et al*. 2010 The ATLAS TAGS database distribution and management: operational challenges of a multi-terabyte distributed database *J. Phys. Conf. Ser.* **219**:072058
[6] Cranshaw J *et al*. 2010 Event selection services in ATLAS *J. Phys. Conf. Ser.* **219**:042007
[7] The ATLAS Collaboration (Ehrenfeld W *et al*.) 2011 Using TAGs to speed up the ATLAS analysis process *J. Phys. Conf. Ser.* **331**:032007
[8] Barberis D *et al*. 2014 The future of event-level information repositories, indexing, and selection in ATLAS, these proceedings.
[9] Hadoop: http://hadoop.apache.org
[10] HBase: http://hbase.apache.org
[11] ActiveMQ: http://activemq.apache.org