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

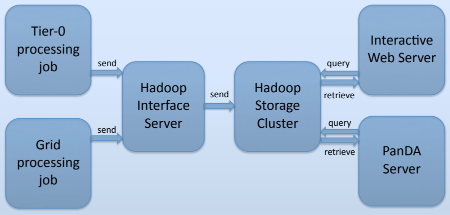
CHEP2013 – Computing in High Energy and Nuclear Physics 2013, 14-18 October 2013, Amsterdam, Netherlands

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] 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

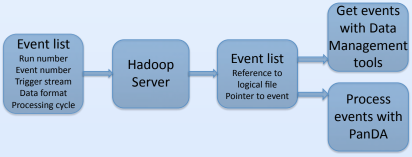
EventIndex: an event-level metadata catalogue for all ATLAS events

- Modern scientific experiments collect large amounts of data that need cataloguing according to different points of view to meet multiple use cases and search criteria.
  - ATLAS produced 2 billion real events and 4 billion simulated events in 2011 and the same in 2012.
- A database that contains the reference to the file that includes every event at every stage of processing is necessary to retrieve selected events from data storage systems.
- Using NoSQL technologies we can store information for each event in a single logical record.
  - The EventIndex record is created upon recording of the event from the online system:
    - Event number, run number, time stamp, luminosity block number, trigger that selected the event, and the identifier of the file that contains the event in RAW format
  - Each reconstruction campaign produces new versions of every event, in different formats, and adds information to the EventIndex record including the identifiers of all files containing it

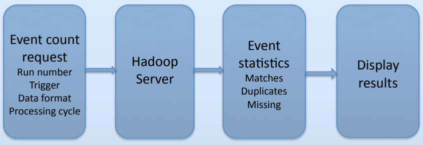


EventIndex use cases

- Event picking
  - Give me the reference (pointer) to "this" event in "that" format for a given processing cycle



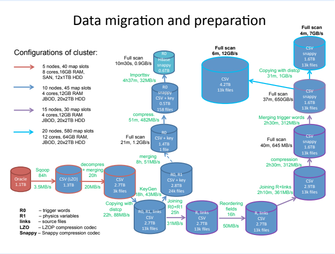
- Production consistency checks
  - Technical checks that processing cycles are complete (event counts match)

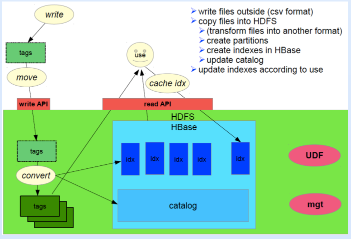


- Event service
  - Give me the references for this list of events, to be distributed individually to processes running on (for example) HPC and/or cloud clusters

Tests of storage technologies

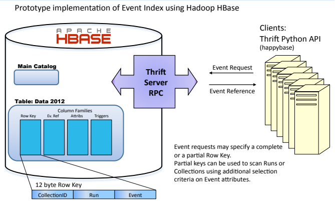
- ATLAS developed for many years an event metadata catalogue, the TAGDB, based on Oracle technologies [1,2].
- LHC Run-2 in 2015-2017 will produce several billion raw events/year and about the same number of simulated events. A metadata database for this amount of data will need to scale to more than 100 TB of payload information and have matching processing power.
- Out of the many NoSQL [3] structured storage solutions, Hadoop [4] and its many associated tools looks like the most promising solution.
- ATLAS tested during 2013 several storage formats based on Hadoop using clusters provided by CERN-IT and importing a 1-TB dataset from the TAGDB (all Tier-0 processing of 2011):
  - TAG events and links tables were imported from the Oracle database to the Hadoop cluster in CSV format.
  - The sequence of comma-separated columns is exactly the same as in Oracle.
  - Every collection was then imported into the HBase test table. The table has 2 column families, each containing 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".
  - 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.


Data migration and preparation



- write files outside (csv format)
- copy files into HDFS
  - (transform files into another format)
  - create partitions
  - create indexes in HBase
  - update catalog
- update indexes according to use

- A prototype application was 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).
  - 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 data 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

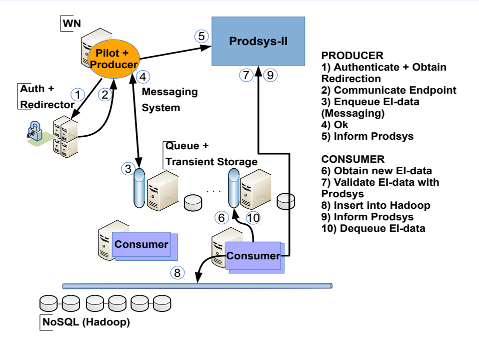

Prototype implementation of Event Index using Hadoop HBase

Event requests may specify a complete or a partial Row Key.
Partial keys can be used to scan Runs or Collections using additional selection criteria on Event attributes.

- Preliminary performance benchmarking shows 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. An efficient collection catalog should be created to keep track of the existing collections and indices and their status.

Tests of data collection

- Event metadata for the EventIndex are produced by jobs running on Tier-0 and the Grid.
- The data collection system must be extremely robust to assure completeness of the data catalogue.
- A few estimated rates (from May 2013):
  - 20 Hz of production jobs, each producing up to 50 MB of event metadata
  - Average event processing rate (all stages) was 3.5 kHz
  - During data-taking periods this rate would double (Tier-0 processing)
  - Trigger rates will also increase to 1 kHz
  - ∴ The system must be ready to deal with 80 Hz of file records containing over 30 kHz of event records to insert into the Hadoop back-end (plus contingency)



PRODUCER
1) Authenticate + Obtain Redirection
2) Communicate Endpoint
3) Enqueue EI-data (Messaging)
4) Ok
5) Inform Prodsys

CONSUMER
6) Obtain new EI-data
7) Validate EI-data with Prodsys
8) Insert into Hadoop
9) Inform Prodsys
10) Dequeue EI-data

- High-level architecture:
  - Producers collect metadata from running jobs (within the PanDA pilot or Tier-0 framework)
  - Information is transmitted to a server at CERN through a resilient messaging system
  - If the job completes correctly, metadata are loaded to the Hadoop server by Consumers

Project timescales

- Jan-Sept 2013: tests of data formats, schemas, performance of upload, search and retrieve data on a reduced dataset (1 TB)
- Oct-Dec 2013: implementation of the chosen solution on the CERN Hadoop cluster; adaptation or development of external services; upload of all existing data
- Jan-Jun 2014: commissioning of the new system; performance optimization
- Jul-Dec 2014: commissioning with new cosmic-ray data; discontinuation of Oracle TAGDB

References

[1] J. Cranshaw et al., "Event selection services in ATLAS", prepared for 17th International Conference on Computing in High Energy and Nuclear Physics (CHEP 09), Prague, Czech Republic, 21-27 Mar 2009. Published in J.Phys.Conf.Ser. 219:042007,2010.

[2] The ATLAS Collaboration (W. Ehrenfeld et al.), "Using TAGs to speed up the ATLAS analysis process", Published in J.Phys.Conf.Ser.331:032007,2011.

[3] NoSQL databases: see http://nosql-database.org and http://en.wikipedia.org/wiki/NoSQL

[4] Hadoop: see http://hadoop.apache.org, http://hbase.apache.org and http://hive.apache.org