IT Information Technology Department

A PROTOTYPE FOR THE EVOLUTION OF ATLAS EVENTINDEX BASED ON APACHE KUDU STORAGE

Z. Baranowski[1], L. Canali[1], A. Fernandez Casani[2], E. Gallas[3], C. Garcia Montoro[2], S. Gonzalez de la Hoz[2], J. Hrivnac[4], F. Prokoshin[5], G. Rybkin[4], J. Salt[2], J. Sanchez[2], D. Barberis[6]
on behalf of the ATLAS Collaboration

[1]CERN, Geneva, Switzerland; [2]Insitut de Fisica Corpuscular, Valencia Spain; [3]University of Oxford, Oxford, UK; [4]LAL, Université Paris-Sud and CNRS/IN2P3, Orsay, France; [5] Universidad Tecnica Federico Santa Maria, Chile [6]Università di Genova and INFN, Genova, Italy

ABOUT THE ATLAS EVENTINDEX

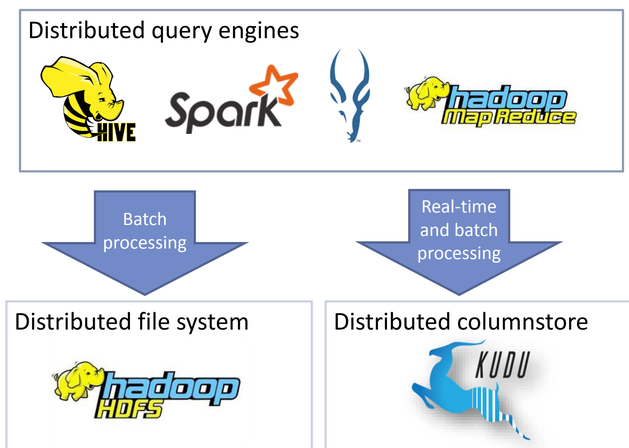
- The ATLAS EventIndex is a **catalogue** of all real and simulated **events** produced by the experiment at all processing stages [1].
- The system contains **hundreds of billions** of event records (180 billions of records as of June 2018), each consisting of ~1000 bytes.
- The goal of the ATLAS EventIndex is to allow **fast** and **efficient selection** of events of interest, based on various criteria, and provide references that point to those events in millions of files scattered in a world-wide distributed computing system.

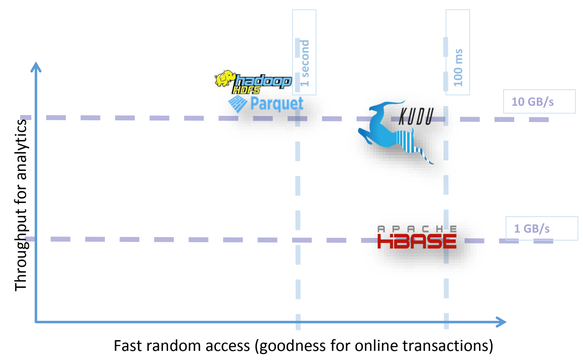
MOTIVATION FOR EVENTINDEX EVOLUTION

- Current EventIndex was designed in 2012-2013 using best BigData technology available at that time (Hadoop), implemented in 2014 using MapFiles and HBase, in operation since 2015 with satisfactory results [2].
- Use cases **extended** in the meantime from event picking and production completeness checks to trigger overlap studies, duplicate event detection and derivation streams (offline triggers) overlaps.
- Fast data querying based on traditional relational database involving a subset of information for real events only [3]
- Also event **rate increased** steadily throughout Run 2.
- BigData **technologies advanced** in the meantime and now we have the choice between many different products and options.
- Studies of **new data formats** and/or new storage technologies [4] concluded that Kudu is the most promising technology for the next few years. Hence this prototype.

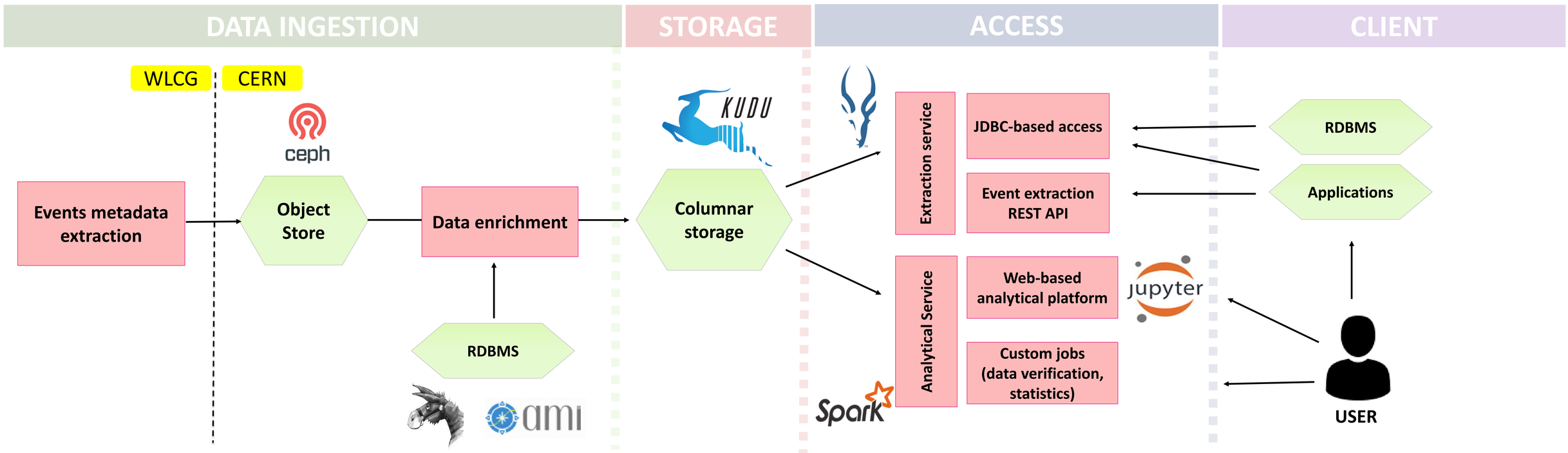
APACHE KUDU

- Next generation scalable and distributed table-based storage designed for HTAP systems – **Hybrid Transactional and Analytical Processing** [5]
- Unlike Hadoop Distributed File System (HDFS), Kudu provides indexing and columnar data organization natively – this is to establish a good compromise between random **data lookups** and **analytics** performance
- Organization of the data in sharded tables with named columns, types and a primary index makes Kudu very attractive for systems with relational data models that needs to scale-out
- Apache Kudu is supported by top open-source frameworks for parallel data processing and computation including
  - Apache Spark, Apache Impala, Apache Hive, MapReduce,…





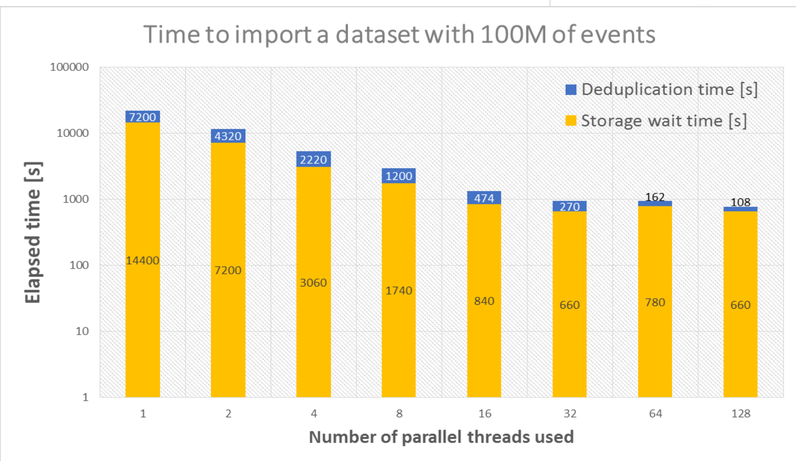
CONCEPT OF THE NEW ATLAS EVENT INDEX PLATFORM



MEASURED PERFORMANCE WITH APACHE KUDU STORAGE

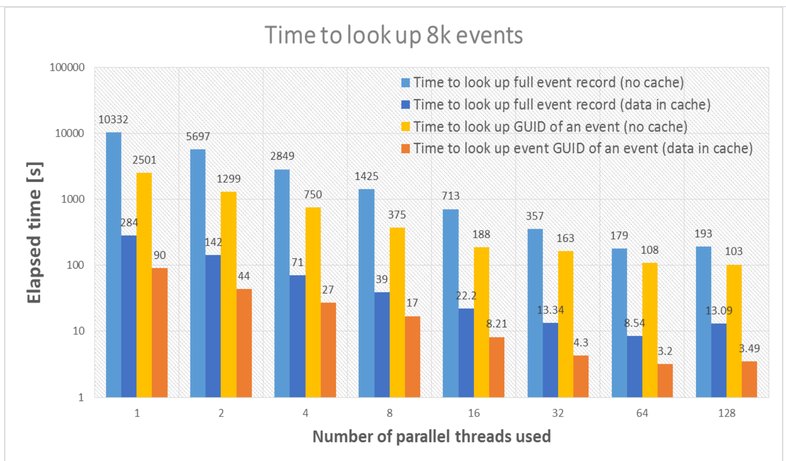
DATA INGESTION

Hardware specification: cluster of 12 machines with 2 x 8 cores @2.60GHz, 64GB RAM, 48 SAS drives



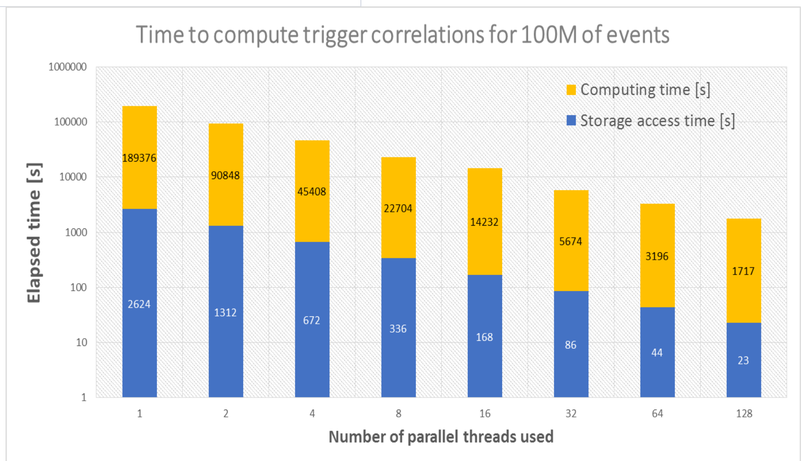
- Data loading tests were performed with **Apache Spark** 2.2.1 using **real** data from the current production system
- Before loading a dataset to Apache Kudu, duplicated events are filtered out and stored in a dedicated table
- Measured average writing speed was **5kHz** per thread, max overall writing speed to a Kudu cluster was **120kHz – this is ~10x more than what is needed today**

ACCESS



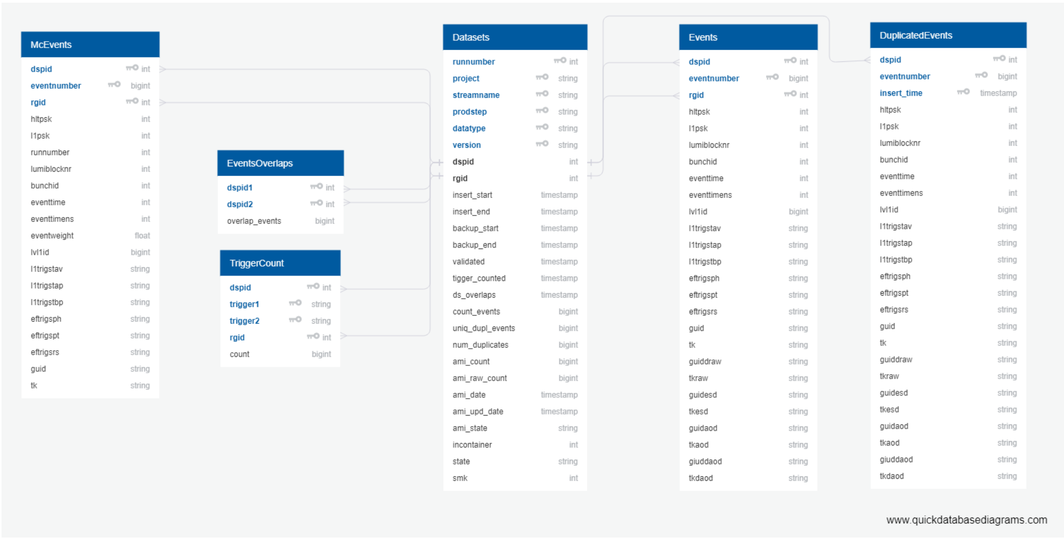
- On the plot above, we present the time to look up eight thousands of random events records (full record or just a GUID attribute) from Apache Kudu with a simple client written in **Python**
- The results for each type of a lookup were grouped into two cases; a pessimistic one (no cache used) and an optimistic one (all data where lookup from Kudu cache)
- **Average event lookup time below 1s and ability to handle more then 400 requests per second fully satisfies the system needs**

ANALYTICS



- Data analytics tests were performed with **Apache Spark** 2.2.1 reading Atlas EventIndex from data stored in Apache Kudu
- In the test case, we count occurrences of all combinations of trigger bits pairs within a dataset of 100M events
- The trigger count computation on a Spark cluster takes the majority of the wall time (52 hours), when data retrieval from Kudu is just a small fraction (< 2%) of it (45 minutes). A single scanner thread could deliver 40k of records per second
- Scalable data scan performance in combination with modern data processors (Spark, Impala) opens the system to new use cases on a filed of data exploration and analytics (like counting trigger correlations)

DATA SCHEMA PROTOTYPE IN APACHE KUDU



REFERENCES

[1] Barberis D et al. 2014 The ATLAS EventIndex: an event catalogue for experiments collecting large amounts of data, J. Phys. Conf. Ser. 513 042002

[2] Barberis D et al. 2015 The ATLAS EventIndex: architecture, design choices, deployment and first operation experience, J. Phys. Conf. Ser. 664 042003

[3] Gallas E et al. 2016 an Oracle-based Event Index for ATLAS

[4] Z. Baranowski et al 2017 J. Phys.: Conf. Ser. 898 062020

[5] Lipcon T et al., 2015 Kudu: Storage for fast analytics on fast data.

CHEP 2018 Sofia, Bulgaria