

A STUDY OF DATA REPRESENTATION IN HADOOP TO OPTIMIZE DATA STORAGE AND SEARCH PERFORMANCE FOR THE ATLAS EVENTINDEX

Z. Baranowski¹, D. Barberis², L. Canali¹, J. Hrivnac³,
R. Toebbicke¹

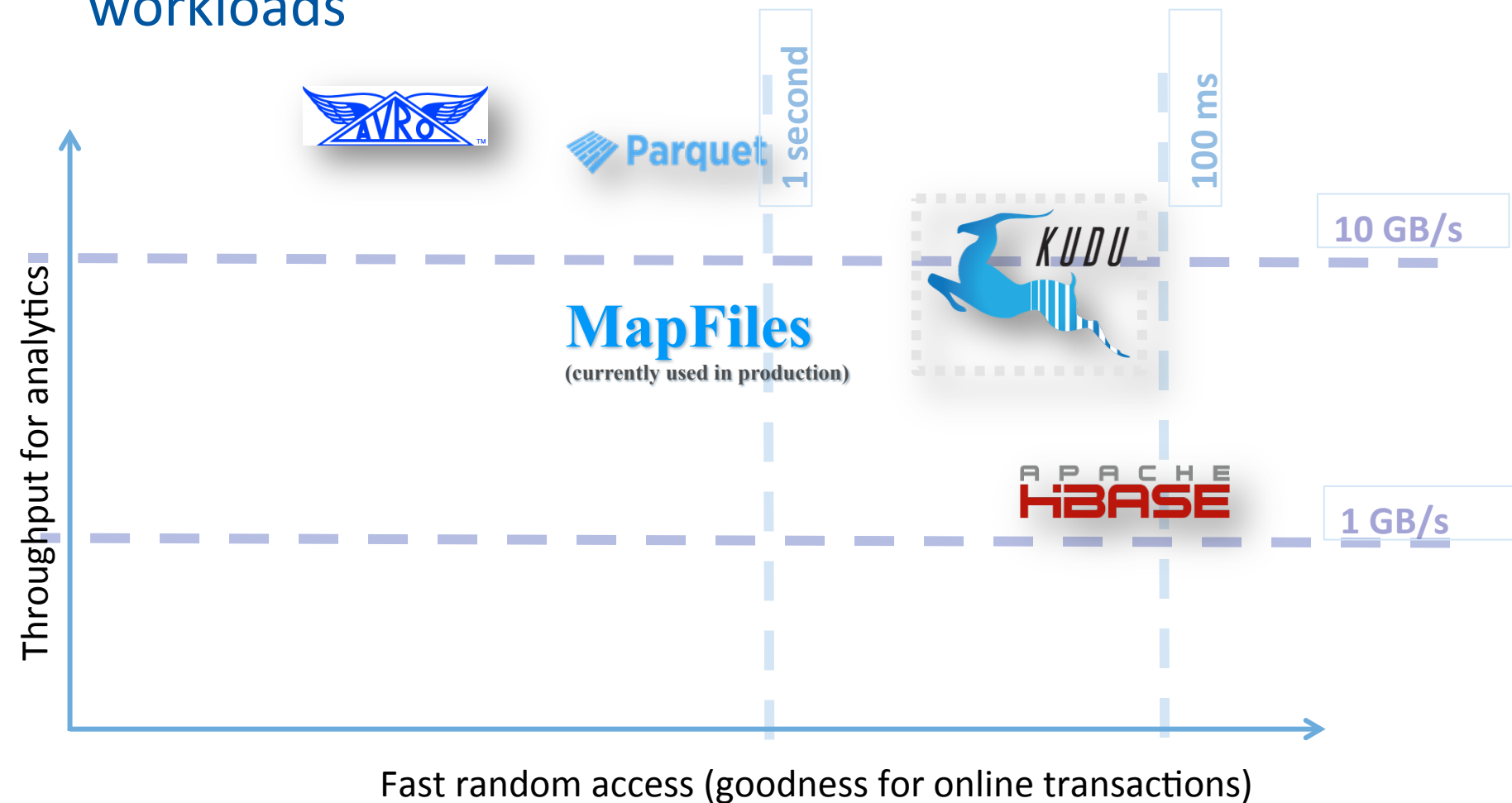
on behalf of the ATLAS Collaboration

¹CERN, Geneva, Switzerland; ²Università di Genova and INFN, Genova, Italy; ³LAL, Université Paris-Sud and CNRS/IN2P3, Orsay, France

Objective

- The ATLAS EventIndex is a **catalogue** of all real and simulated **events** produced by the experiment at all processing stages (6e10 records as of September 2016).
- 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.
- This work reports on tests using several popular data formats and storage solutions including **Map Files, Apache Parquet, Apache Avro, Apache Kudu** in order to improve the performance of storing and searching data within the ATLAS EventIndex system.

Overview of the performance measured with the technologies tested for analytic and random lookup workloads



Summary

Performed evaluation of alternative approaches for storing and accessing data, revealed new opportunities for **improving** Atlas EventIndex system on:

- **Storage efficiency** – with Parquet or Kudu and Snappy compression the total volume of the data can be reduced by a factor 10
- **Data ingestion speed** – all tested solutions provide faster ingestion rate (between x2 and x50) than the current data format used in production
- **Random data access time** – using Apache HBase or Apache Kudu, typical random data lookup is below 1s.
- **Data analytics** – with Apache Parquet or Apache Kudu it is possible to perform very fast and scalable (typically 300k of records per second per core on a cluster node) data aggregation, filtering and reporting.
- **Support of data mutation** – Apache HBase and Apache Kudu can modify records (schema and values) in place.

According to the tests, columnar stores like **Apache Parquet** and **Apache Kudu** achieve the best compromise between **fast data ingestion**, fast random data **look-up** and scalable data **analytics**.