ATLAS EventIndex in Hadoop

Hadoop User Forum

4 December 2023

Grigori Rybkine

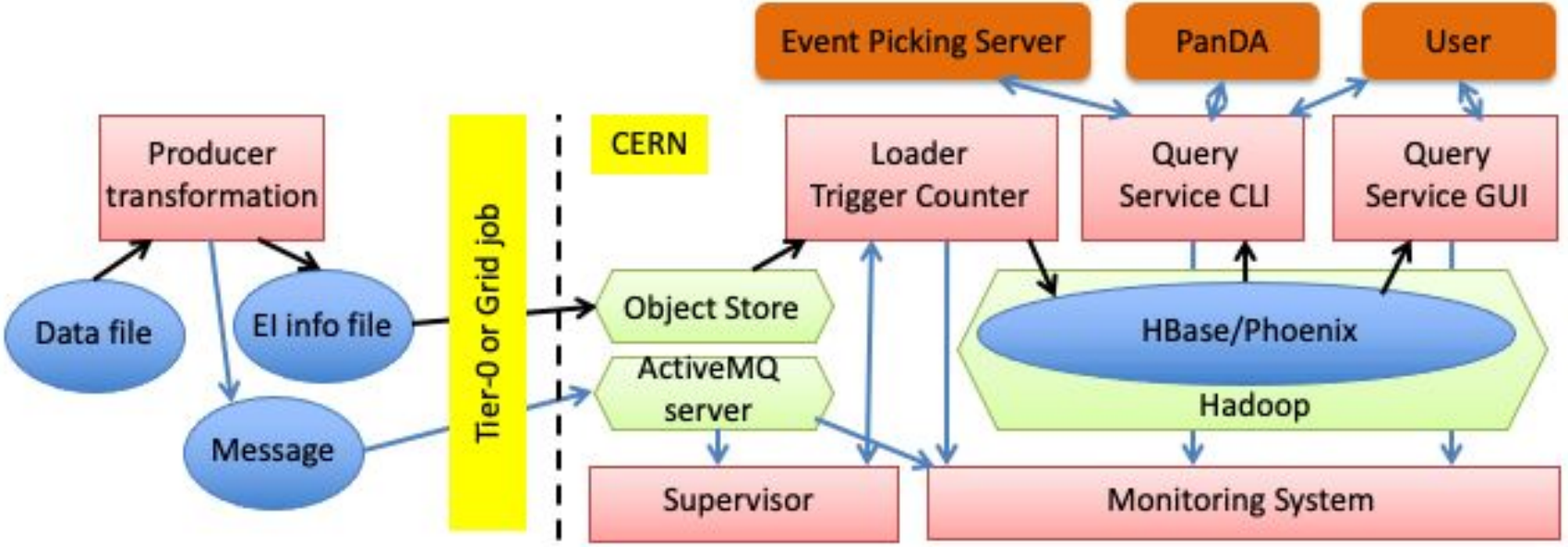
IJCLab Orsay (France)

on behalf of the ATLAS EventIndex team

The ATLAS EventIndex

- The EventIndex is the global catalogue of all ATLAS events
- For each event, each data format and each processing version, it contains:
  - Event identifiers (run and event number, plus other stuff)
  - Location (GUID of the file containing it) and provenance
  - Trigger and other useful metadata
- Main use case is event picking for detailed analysis and/or displays
  - Also production checks and dataset overlap counts
- The first EventIndex version was designed in 2013, implemented in 2014 and started operations in 2015
  - Software tools evolved considerably since then!
- The partitioned architecture allowed the replacement of individual components during the years

EventIndex functional schema and data flow

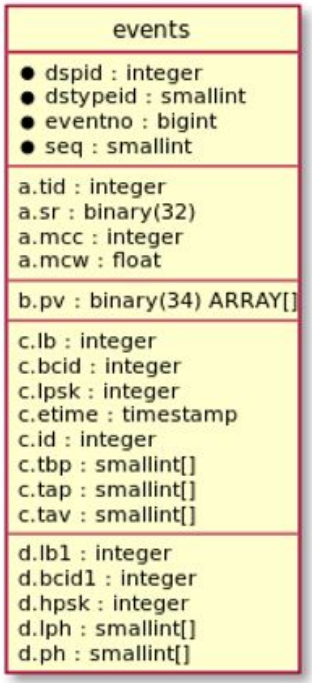


- Grid jobs extract EventIndex data from each data file and send it to CERN
- Data is aggregated per dataset and stored in HBase tables:
  - “Datasets” table with dataset-level information
  - “Events” table with event-level information
  - Auxiliary tables with other information useful to retrieve and decode event records

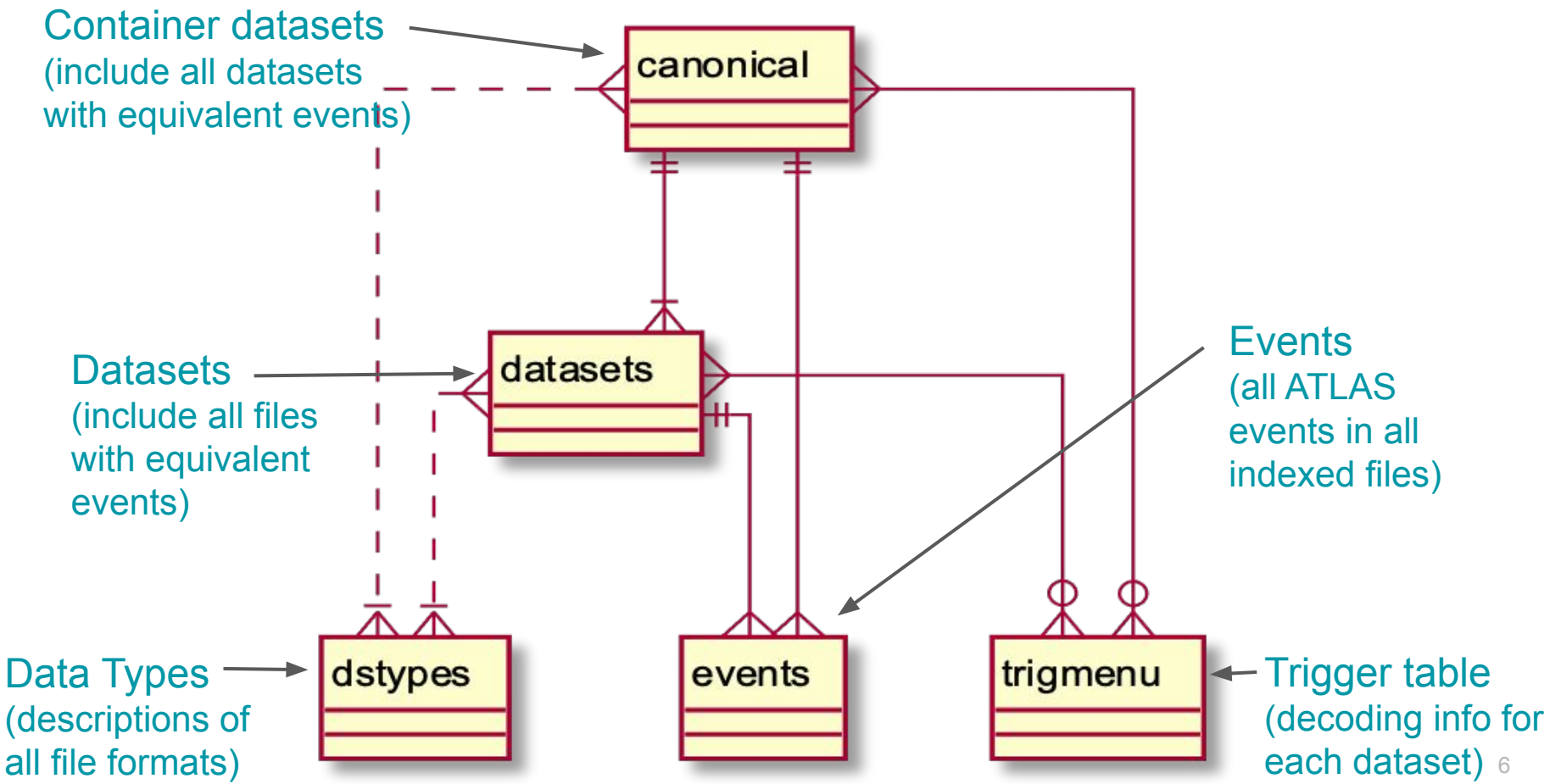
EventIndex in Hadoop: history

- The first implementation in 2014 stored events in HDFS MapFiles, one per dataset
  - HBase was used for the dataset catalogue, and later also to store event records with a subset of the information (no trigger)
  - For read performance reasons, a copy of a subset of events (real data only) with a subset of information (no Trigger) was made to Oracle too
- The core data storage system was reimplemented during 2021 and deployed in 2022 for the start of LHC Run 3
  - HBase for the “datasets” and “events” tables
  - Phoenix interface for SQL queries
  - New client query service CLI implemented for optimal performance
- Old data (Run 1 and Run 2) were transferred to the new structures and new Run 3 data written directly to HBase
  - Simulated data as well!

EventIndex in HBase: tables



EventIndex in HBase: relations



EventIndex in HBase: query service

- Server-client architecture
- Server in Java, making use of Spring Framework
- Database - Phoenix/HBase tables - access via Phoenix JDBC driver
- Provides required performance in terms of both time and number of events queried
- Achieved by heavy use of INNER JOIN in SQL queries
  - SELECT ... FROM EVENTS INNER JOIN (SELECT ... FROM CANONICAL WHERE ...) ON ...
- lhs INNER JOIN rhs
  - rhs will be built as hash table in server cache

EventIndex in HBase: query service configuration

- With default Phoenix configuration, had server-side

  HashJoinCacheNotFoundException
- Had to tune Phoenix configuration, in particular, on the server side
  - phoenix.coprocessor.maxServerCacheTimeToLiveMs

    360000

    Default

    30000

EventIndex in HBase: query service in production

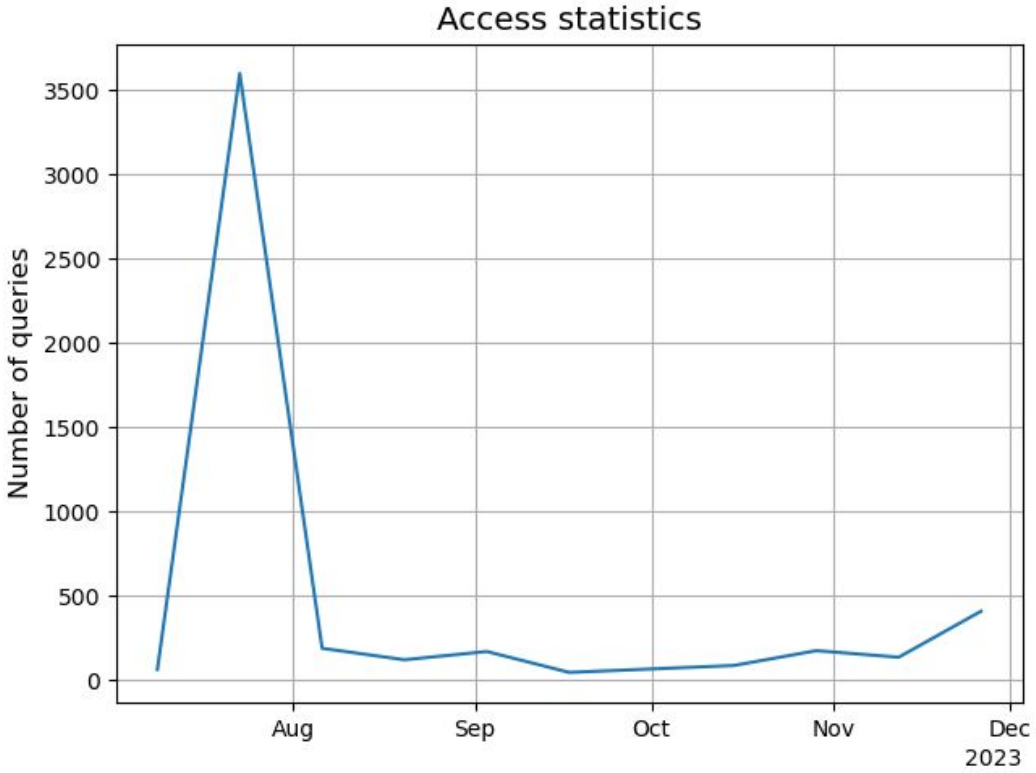
- Putting in production required additional work
  - normally unnecessary since done by Maven/Gradle tools
- Phoenix release deployed on the Hadoop cluster and clients
  - **custom** build of Phoenix-5.0 against **HBase-2.2**
  - artifacts not distributed standard Java way to https://mvnrepository.com or similar repository

EventIndex in HBase: size

- (Container) Datasets: 280k
  - Run 1 data: 2300 — only real data produced at Tier-0 in AOD format
  - Run 2 data: 88400 — includes reprocessing and many derivation formats (DAODs)
  - Run 3 data: 6600 — reduced DAOD formats and no repro yet for 2023 data
  - Run 2 MC: 170k — many small datasets with variants of event generation
  - Run 3 MC: 12k — analysis of 2023 data just starting as stats for real data is low
- Events table entries: 850 billion
  - Largest MC dataset has 2 billion events
  - Largest real data dataset has 1.75 billion events (minimum bias triggers)
- Data volume in HBase: 78 TB (234 TB after 3x replication)

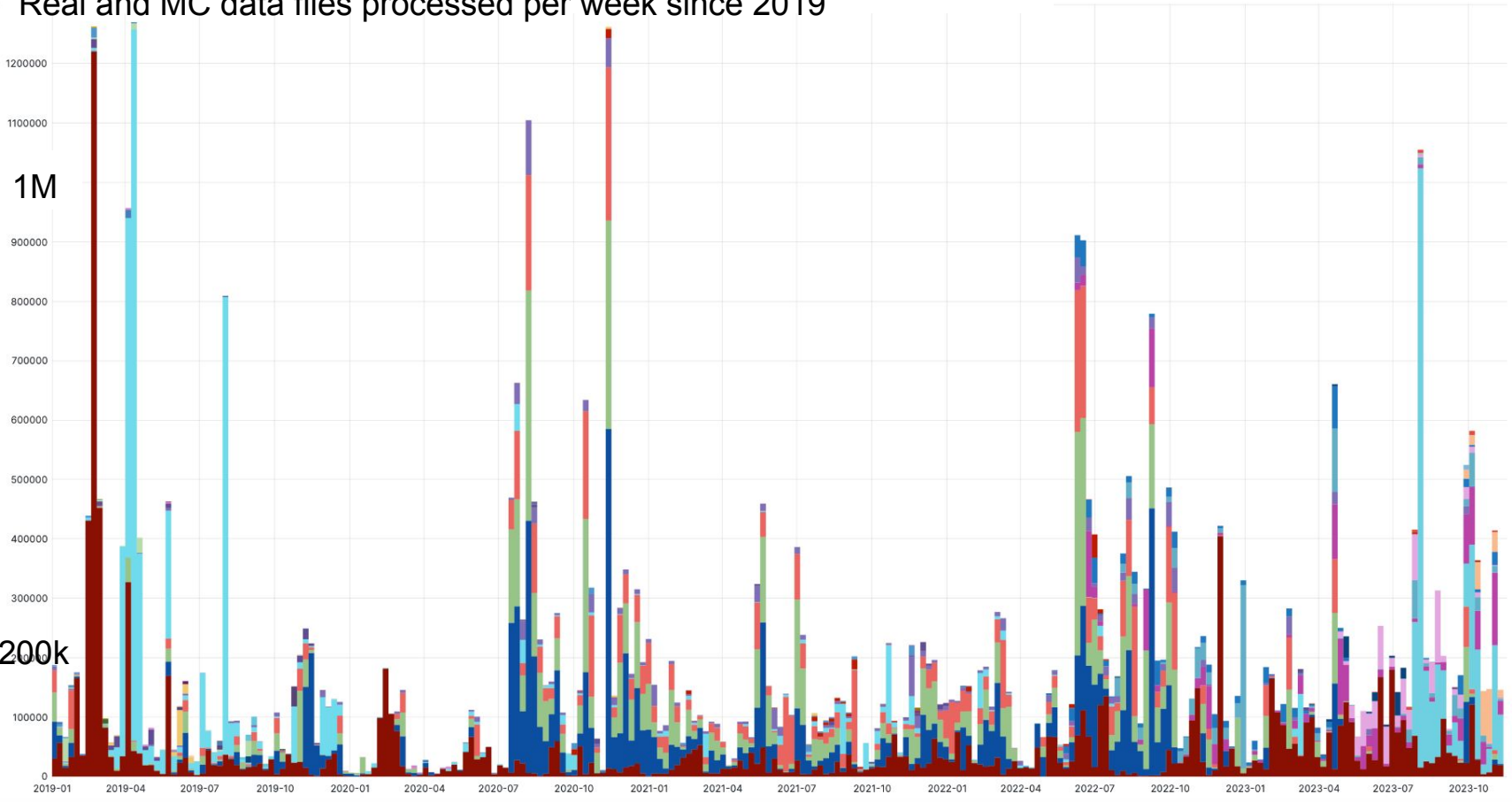
EventIndex in HBase: query rates

Data query rates the last few months



EventIndex in HBase: ingestion rates

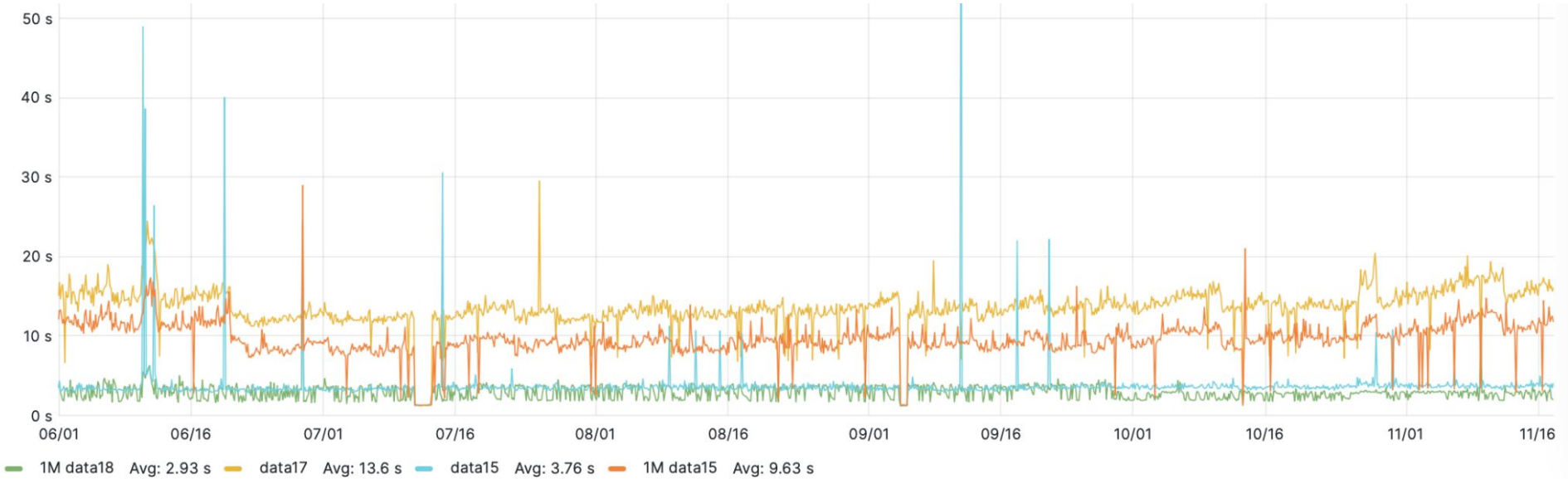
Real and MC data files processed per week since 2019



	total
mc16_13TeV	10669743
data18_13TeV	8406227
data17_13TeV	7946769
data16_13TeV	5640851
mc15_13TeV	5389151
mc23_13p6TeV	2694714
mc20_13TeV	1727690
data15_13TeV	1342005
data22_13p6TeV	1141326
data23_13p6TeV	820315
mc21_13p6TeV	761793
data23_hi	302400
data17_5TeV	219792
data18_hi	170452
mc15_pPb8TeV	142163
mc15_5TeV	120615
data15_hi	111319
data16_hip8TeV	108416
mc15_7TeV	46120
mc16_5TeV	45409
data16_hip5TeV	33262
mc21_14TeV	27924
data15_5TeV	26512
data12_8TeV	10979
mc16_pPb8TeV	10444
data23_5p36TeV	7937
mc15_14TeV	7749
data17_hi	7130
mc16_5p44TeV	6092
mc15_8TeV	5000
data11_hi	4737
valid1	1940
data23_900GeV	1516
data22_900GeV	1262
data22_hi	1101

EventIndex in HBase: performance tests

Event lookup response times for queries of 1000 events from datasets of different sizes between June and November 2023.
Queries generally succeed but there are occasional hiccups. Retries always work.



EventIndex in Hadoop: Outlook

- The system seems to run stably and we don't plan any major change till the end of Run 3 (end of 2025)
  - The priority for the coming months is to adapt all our scripts to run on AL9 instead of CC7
  - We are short of technical personpower and the situation won't improve any time soon
- Some ideas start being discussed targeting Run 4 (x3 data and MC rates)
  - Tests done with exporting and querying data in Parquet (and the Delta format) look promising, but still partial (notably no ingestion tests)
  - Considerable work would be needed for a complete system

Thank you!