

Yandex



SCHOOL OF DATA ANALYSIS

Event Index for data processing

A. Redkin¹ N. A. Kazeev²³ A. E. Ustyuzhanin²³

¹Yandex ²MIPT ³Yandex School of Data Analysis

Introduction

Event Index

stripping=20 hits



Find

Try wizard or see example: HAS Stripping60ZDD00Beauty2ChemLineDecision AND stripping=201

stripping=20 nRich1Hits

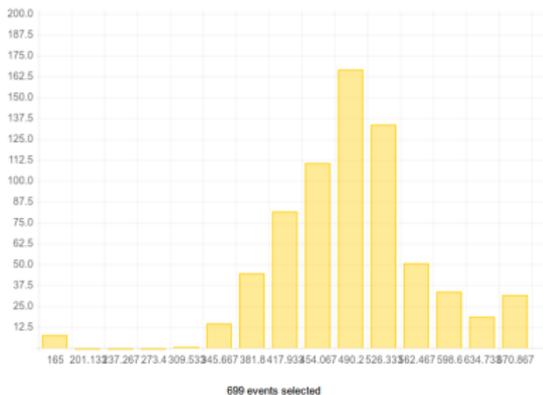
stripping=20 nRich2Hits

stripping=20 nSPDhits



Data Factory

SCHOOL OF DATA ANALYSIS



Axis X

type to find variable

- nMuonTracks
- nSPDHits
- nBackTracks
- nVeloTracks
- nVeloClusters
- nOTClusters
- nPVs
- nRich1Hits
- nTracks

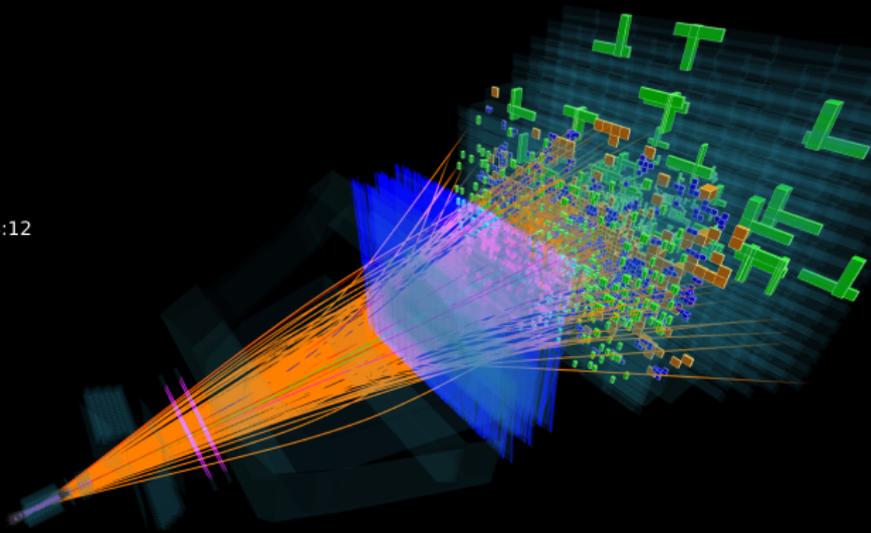
Intervals count

15

15



Event 49402778
Run 126908
Sat, 01 Sep 2012 01:48:12



Query examples

- › stripping=20 AND stream=minibias nPVs=0
- › stripping=20 AND stream!=minibias nPVs=0
- › stripping=20 AND stream=minibias nPVs=0
nDownstreamTracks>10
- › stripping=20 stream=dimuon nMuonTracks=142 nPVs=5
- › stripping=20
StrippingB02D0D0KSDDBeauty2CharmLineDecision

Please note that AND and HAS operators can be implicit. <https://twiki.cern.ch/twiki/bin/view/LHCb/EventIndex>

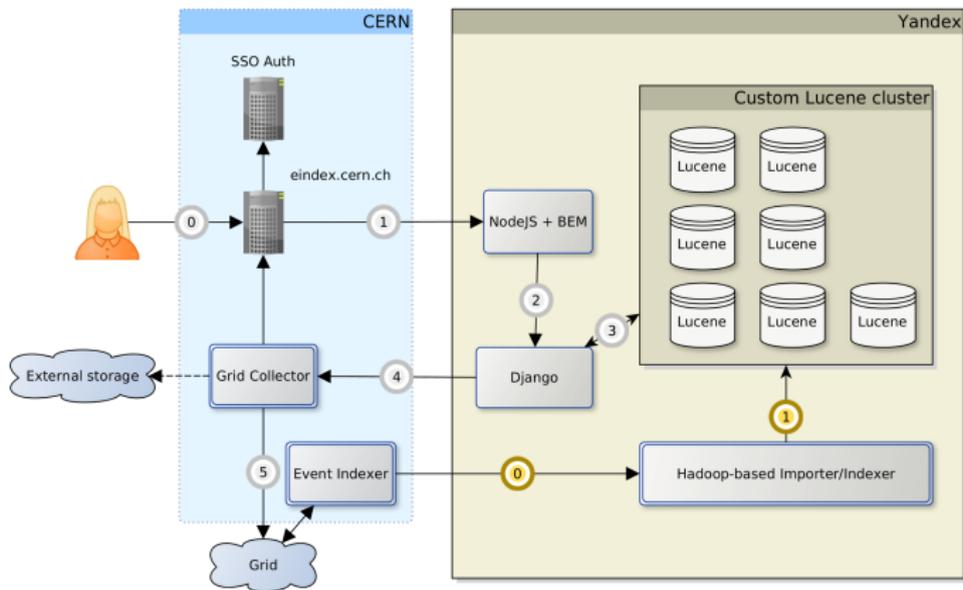
LHCb Event Index

- › Has information about the stripping lines output
- › Has information about the global activity counters (nPVs, nTracks, etc.)
- › Integrated with the LHCb Event Display
- › Integrated with GRID downloader (can fetch small amount of events from the GRID)

https://gitlab.cern.ch/YSDA/grid_collector

Design

Architecture overview



Previous version - HBase

- › Previous version used HBase
- › HBase only has support for a single primary key
- › Had a clever heuristic to avoid full scan in some cases

Performance

- › Less than 90 s. for a histogram over $8 \cdot 10^9$ records for the 7-node LHCb installation.
- › Optimized for bulk indexing. 10^{10} in 3 days on the LHCb installation

Technologies used

- › Apache Lucene for backend
- › MapReduce for indexing
- › Python Django for middleware
- › NodeJS and BEM for frontend
- › GaudiPython and LHCbDIRAC for GRID downloader
- › (CERN-developed SSO) Shibboleth
- › (LHCb-developed Event Display) WebGL

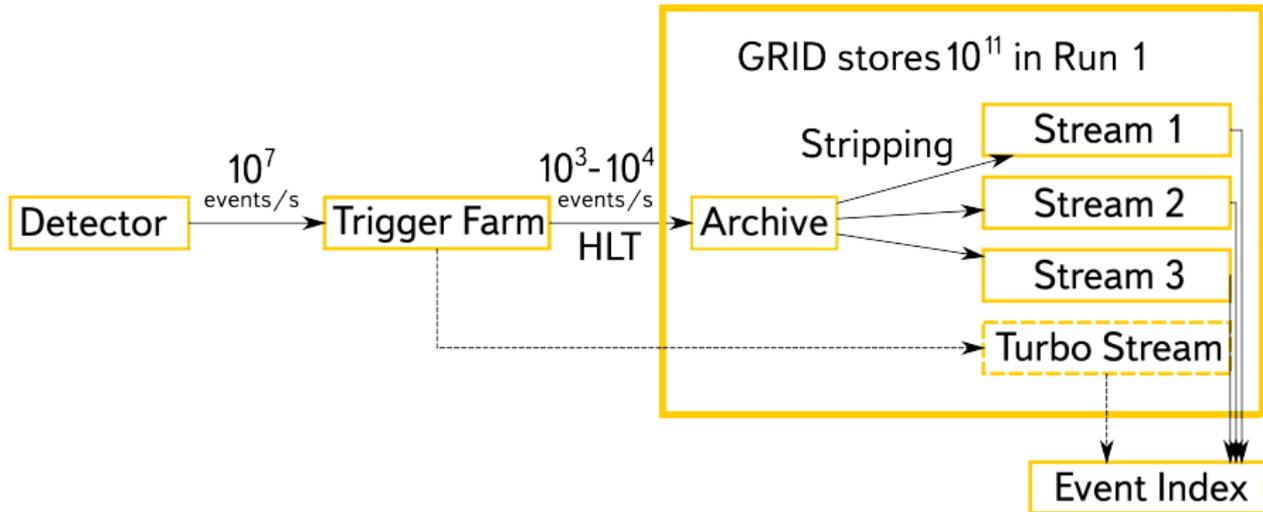
Cluster configuration

- › 7 VM nodes
- › 43 Gb RAM per node
- › 32 CPU cores per node



Ideas: what can be
done with the existing
data?

LHCb processing pipeline



Ideas: what can be done with the existing data?

As a selection tool for
analysis

The problem with the Event Index approach

LHCb software doesn't allow random access to individual events. A job running over Event Index search result will only be marginally faster than a job running over the whole stream - while being a lot more complex.

API access

- › Idea: HTTP API + python library implementing it
- › The same capacity as the web version - search, histograms
- › Could be useful for unique tasks, such as the reconstruction versions comparison in $B_s^0 \rightarrow \mu^+ \mu^-$ (<https://indico.cern.ch/event/337568/session/13/contribution/10>)
- › Now we are implementing a workaround for CERN SSO

Bulk GRID download by runNumber, eventNumber and stream

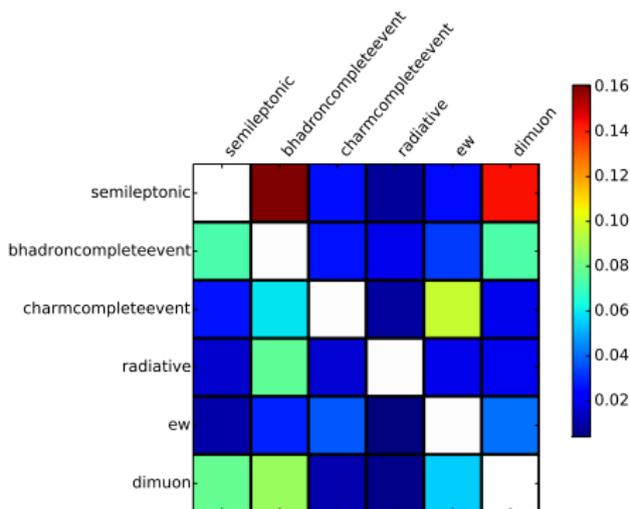
- › Relies on the API to map the desired events to the files and locations in them
- › Uses the GRID collector on the user side to fetch the events

Ideas: what can be done with the existing data?

Streams optimization

Streams optimization

- › Streams are a trade-off between the analyses speed and space lost to the overlap
- › Event Index has access to the event-level stripping lines output and can use that to optimize the distribution of the lines between the streams



Stripping line / trigger optimization - the problem

Correct me if I'm wrong.

- › Line rates are decided by the human expertise
- › Line authors tune their lines to achieve the assigned rate
- › An event passes the selection if any of lines fired
- › An event may pass several lines - to fully utilize the total available bandwidth the individual line rates must be guessed and tuned

Automatic threshold tuning

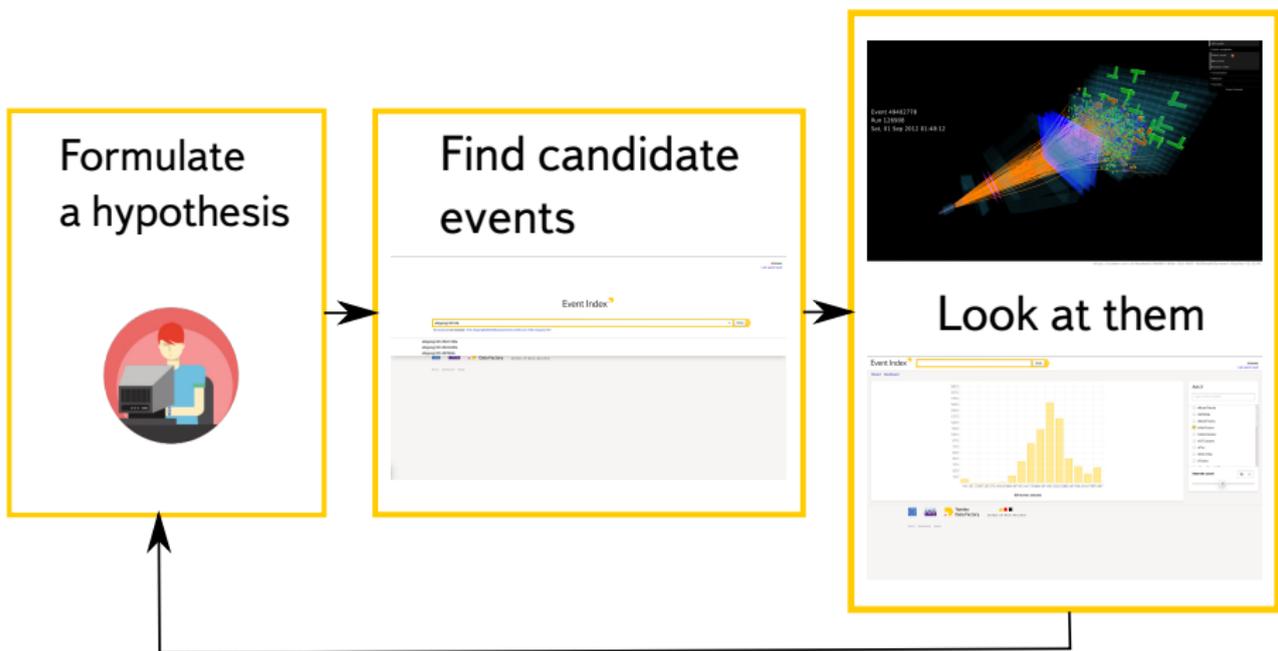
- › Let a line output not a binary decision, but a probability value
- › Run them on a min bias sample
- › Solve a constrained optimization problem

t_i - threshold for the i -th line, r_i - event count for the i -th line, R_i - expert-assigned minimal event count for the i -th line, $e_{i,j}$ - i -th line output for j -th event, L - limit on the total selected events number

$$r_i = \sum_j [e_{i,j} > t_i]; \quad \sum_{j:\exists i:e_{i,j}>t_i} 1 \leq L$$

$$\min_i \frac{r_i}{R_i} \rightarrow \max$$

Exploration



Similar events search

- › A user has some events (for example, MC) and wants to find similar in the data.
- › He runs stripping lines over his events.
- › The system fits a one-class classification on the stripping lines outputs and global activity counters (one class SVM for example) and does the dimensionality reduction.
- › The classifier parameters are turned into an Event Index query

Thanks for your attention

Status:

- › Has the data on stripping lines and global activity counters
- › Can search and aggregate (i.e. histograms)
- › LHCb integration - Event Display and GRID collector

Ideas (and we don't have the manpower for all of them):

- › API & mass GRID download (WIP)
- › Stream optimization
- › Stripping / trigger selection optimization
- › Exploration
- › Similar events search

Contacts

Nikita Kazeev

Artem Redkin



kazeevn@yandex-team



artemredkin@yandex-team.ru

Andrey Ustyuzhanin



anaderi@yandex-team.ru