The ATLAS EventIndex

architecture, design choices, deployment and first operation experience

Dario Barberis

Genoa University/INFN

On behalf of the ATLAS Collaboration

D. Barberis (1), S.E. Cárdenas Zárate (2), J. Cranshaw (3), A. Favareto (1), Á. Fernández Casaní (4),
E. Gallas (5), C. Glasman (6), S. González de la Hoz (4), J. Hřivnáč (7), D. Malon (3),
F. Prokoshin (2), J. Salt Cairols (4), J. Sánchez (4), R. Többicke (8), R. Yuan (7)

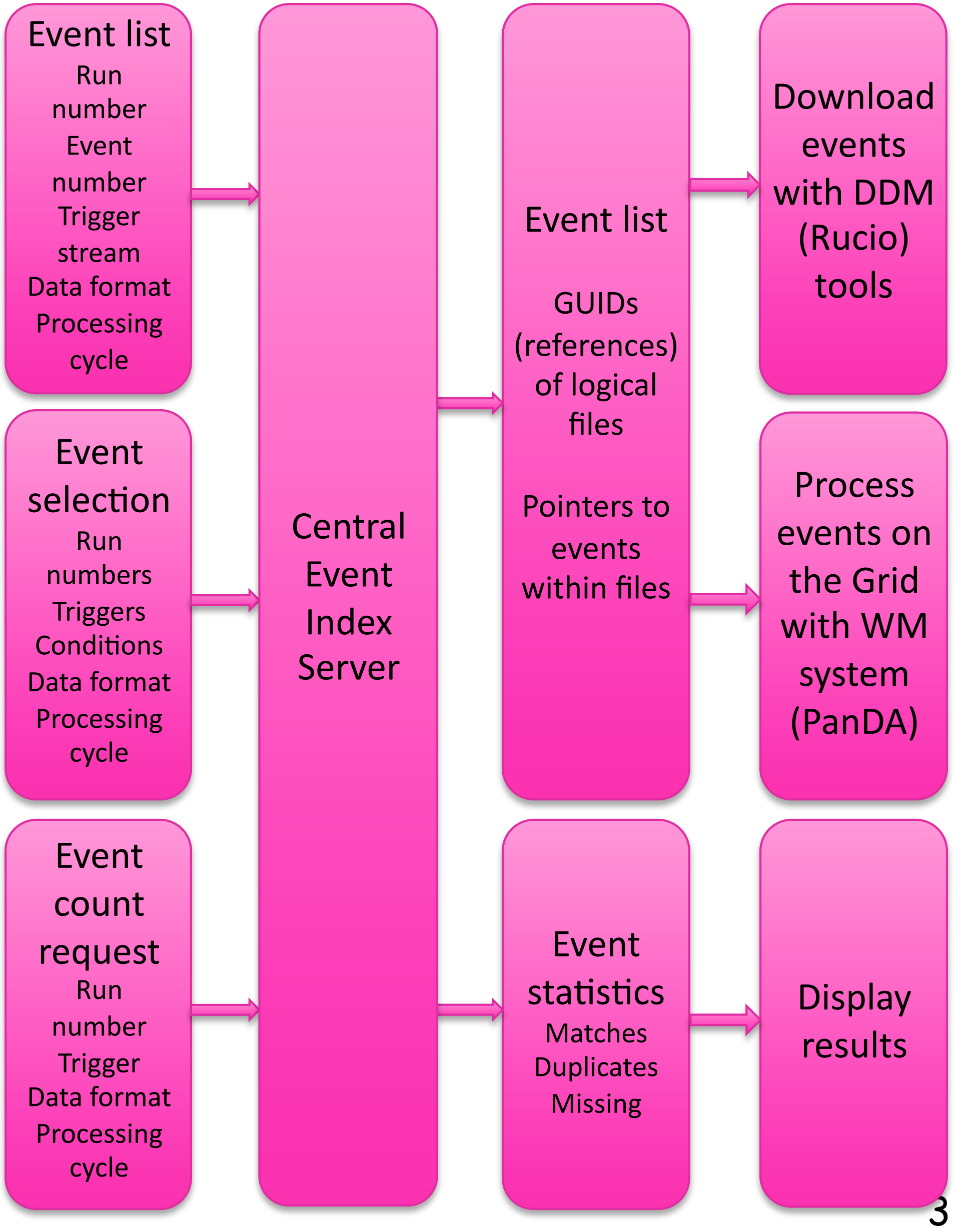
(1) Università di Genova and INFN, Genova, Italy — (2) Universidad Técnica Federico Santa Maria, Valparaíso, Chile —
(3) Argonne National Laboratory, Argonne, IL, United States —
(4) Instituto de Física Corpuscular (IFIC), University of Valencia and CSIC, Valencia, Spain —
(5) University of Oxford, Oxford, UK — (6) Universidad Autónoma de Madrid, Madrid, Spain —
(7) LAL, Université Paris-Sud and CNRS/IN2P3, Orsay, France — (8) CERN, Geneva, Switzerland

What is the ATLAS EventIndex?

- A system designed to be a complete catalogue of ATLAS events
  - All events, real and simulated data
  - All processing stages
- Contents
  - Event identifiers (run and event numbers, trigger stream, luminosity block etc.)
  - Trigger patterns
  - References (pointers) to the events at each processing stage (RAW, ESD, (x)AOD, NTUP) in all permanent files on storage generated by the ATLAS Production System (central productions)
- Size and constraints
  - ATLAS collects a few billion real events each year of data taking and generates more than twice that number of simulated events
  - ~350 B/event ➔ 2 TB of raw information (6 TB after internal replication in Hadoop) in the EventIndex only for LHC Run 1
    - The trigger rate for Run 2 is more than twice that for Run 1
    - Simulated data not counted yet
  - Needs clever storage structure with smart search and retrieve tools

Use cases

- Event picking
  - Give me the reference (pointer) to "this" event in "that" format for a given processing cycle
- Event service
  - Give me the references for this list of events (to be distributed to HPC or cloud clusters for processing)
  - Technically the same as event picking
  - More info in the talk by T. Wenaus (contribution #183)
- Trigger checks and event skimming
  - Count, or give me the list of, events passing "this" selection and their references
- Production consistency checks
  - Technical checks that processing cycles are complete



Event list
Run number
Event number
Trigger stream
Data format
Processing cycle

Event selection
Run numbers
Triggers
Conditions
Data format
Processing cycle

Event count request
Run number
Trigger
Data format
Processing cycle

Central Event Index Server

Event list

GUIDs (references) of logical files

Pointers to events within files

Event statistics
Matches
Duplicates
Missing

Download events with DDM (Rucio) tools

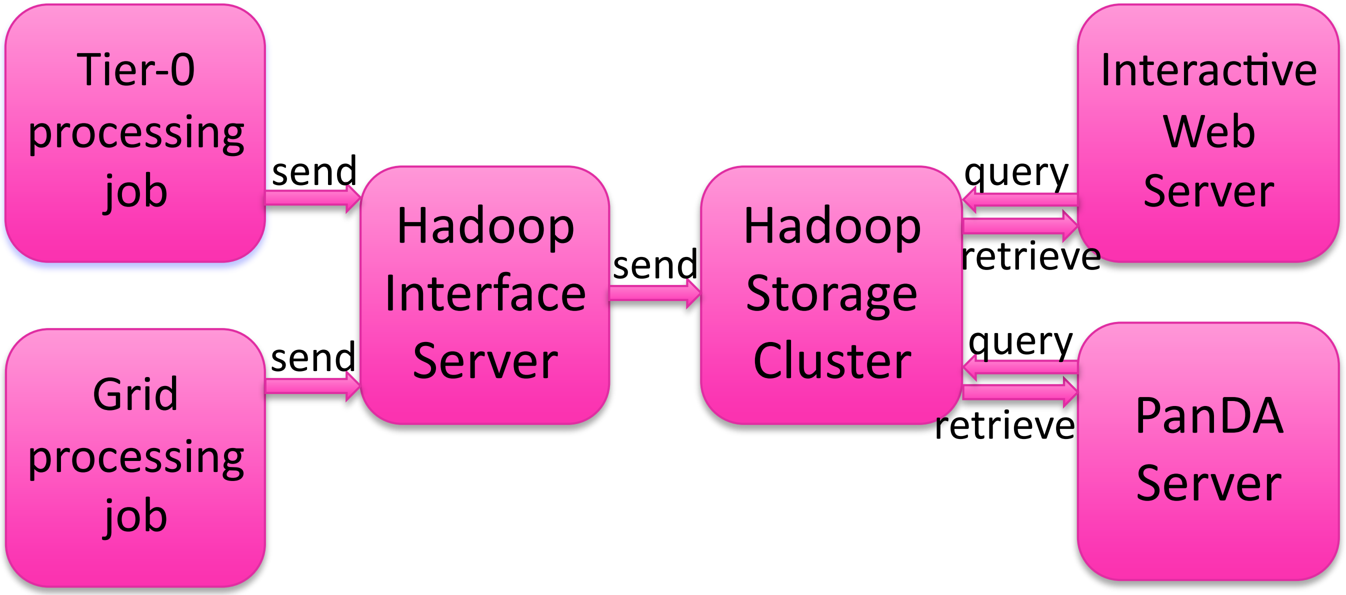
Process events on the Grid with WM system (PanDA)

Display results

EventIndex Project Breakdown

- We defined 4 major work areas (or tasks):

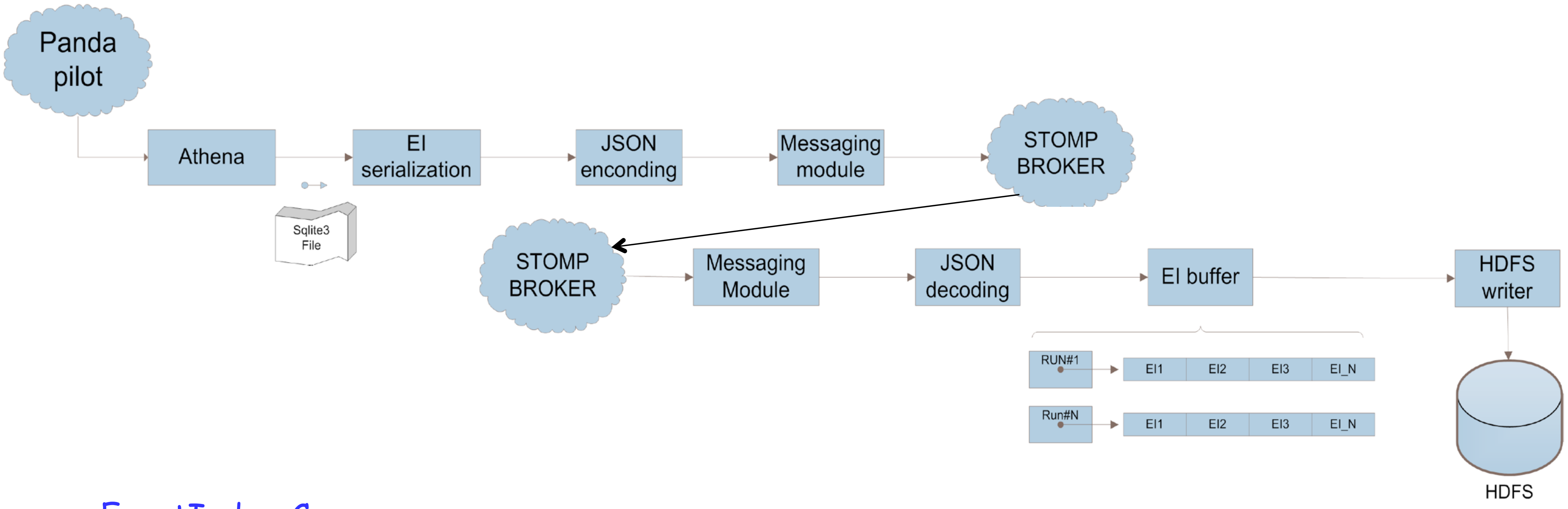
1) Core architecture
2) Data collection and storage
3) Query services
4) Functional testing and operation; system monitoring



Data Collection

More info in the talk by J. Sánchez in this session (contribution #222)

- EventIndex Producer: event processing task which can run at Tier-0 (initial reconstruction at CERN) or on Grid sites (downstream processing)
  - Sends event metadata via ActiveMQ message broker to the Hadoop store at CERN



- EventIndex Consumer: reads the messages from the message broker
  - Organizes data into Hadoop MapFile objects
  - Does validation tasks assessing, for example, dataset completeness
  - Flags aborted, obsoleted, invalid data for further action

Storage and Query Services (1)

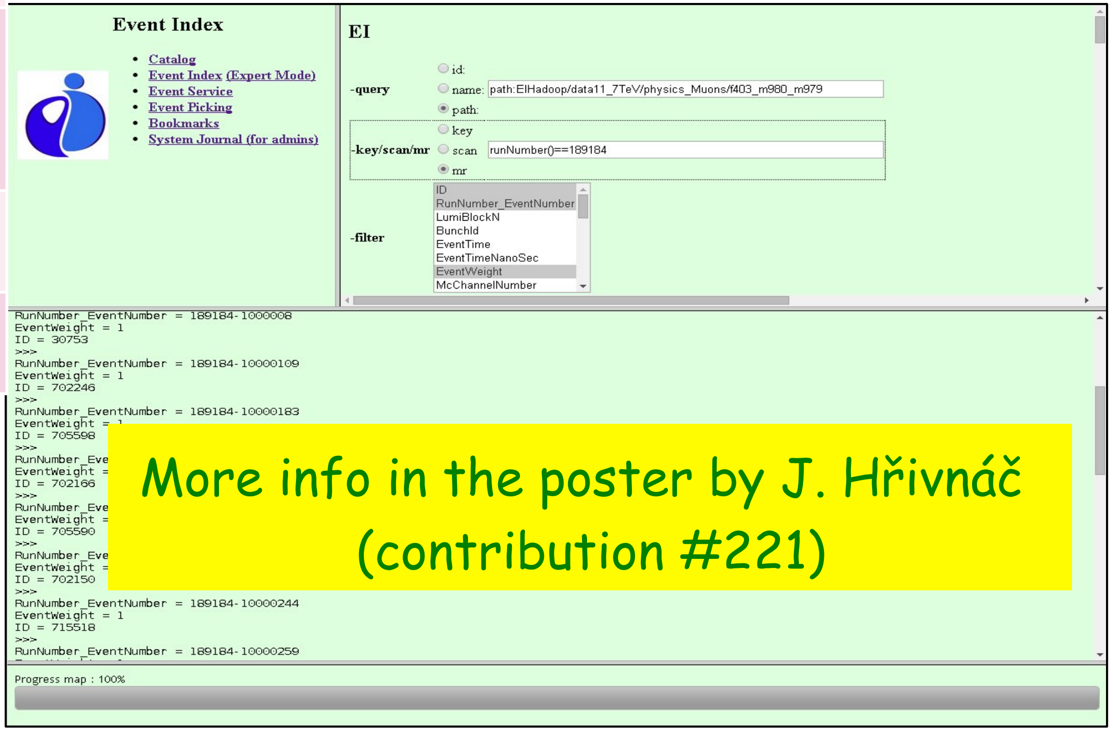
- Hadoop was chosen as the storage technology:
  - Platform is provided and supported by CERN-IT
  - DDM (Distributed Data Management) project also uses Hadoop
  - Plenty of tools to organise the data, index them internally and search them
  - Showed satisfactory performance in prototype populated with a year of ATLAS data (1 TB in the previous TAGDB in Oracle for 2011 data)
- Storage Structure:
  - Data are stored as mapfiles in HDFS (Hadoop File System)
  - Data is catalogued Hadoop HBase: metadata about HDFS files.
- Search performance enhanced using keyed indexes based on use cases:
  - Searches based on a key give immediate results (seconds)
  - Complex searches use MapReduce (MR) and require 1-2 minutes for typical event collections

Storage and Query Services (2)

Query	Search Base	Retrieved	Time (s)
Get Run/ Event	123492895	1	30
Retrieve all	123492895	123492895	3400
Count all	123492895	0	290
Retrieve with trigger stream & sw version	123492895	939220	142
Count with trigger stream & sw version	123492895	0	130
Retrieve with GUID	123492895	41284	204
Count with GUID	123492895	0	192

- Timings measured on the CERN Hadoop cluster with 18 nodes
- Search services via CLI and Web Service GUI

- Typical performance figures for search/ count/retrieve operations on Run1 data:
- Total time depends mainly on the amount of retrieved information (time to write the output file with the search results)
  - "count" is always much faster than "retrieve"



Event Index

- Catalog
- Event Index (Expert Mode)
- Event Service
- Event Picking
- Bookmarks
- System Journal (for admins)

EI

-query ○ id: ○ name: path:EIHadoop/data11_7TeV/physics_Muons/f403_m980_m979 ● path:

-key/scan/mr ○ key ○ scan runNumber()==189184 ● mr

-filter ID, RunNumber_EventNumber, LumiBlockN, BunchId, EventTime, EventTimeNanoSec, EventWeight, McChannelNumber

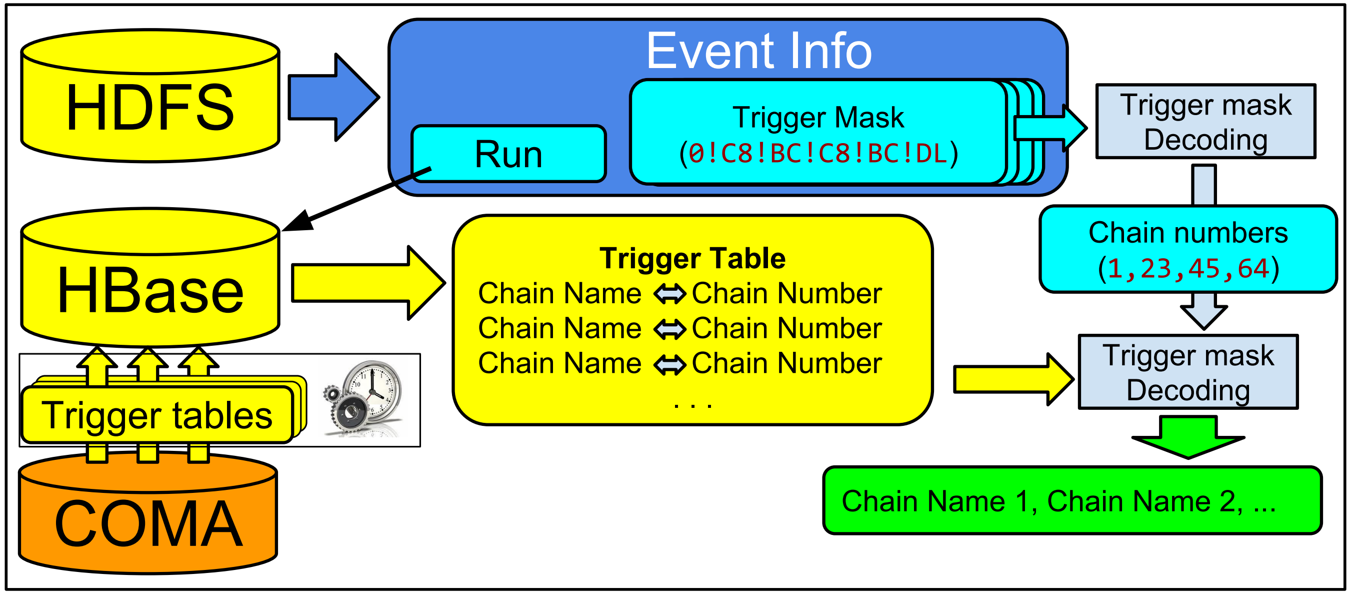
```
RunNumber_EventNumber = 189184-1000008
EventWeight = 1
ID = 30753
>>>
RunNumber_EventNumber = 189184-10000109
EventWeight = 1
ID = 702246
>>>
RunNumber_EventNumber = 189184-10000183
EventWeight = 1
ID = 705598
>>>
RunNumber_Eve
EventWeight =
ID = 702166
>>>
RunNumber_Eve
EventWeight =
ID = 705590
>>>
RunNumber_Eve
EventWeight =
ID = 702150
>>>
RunNumber_EventNumber = 189184-10000244
EventWeight = 1
ID = 715518
>>>
RunNumber_EventNumber = 189184-10000259
```

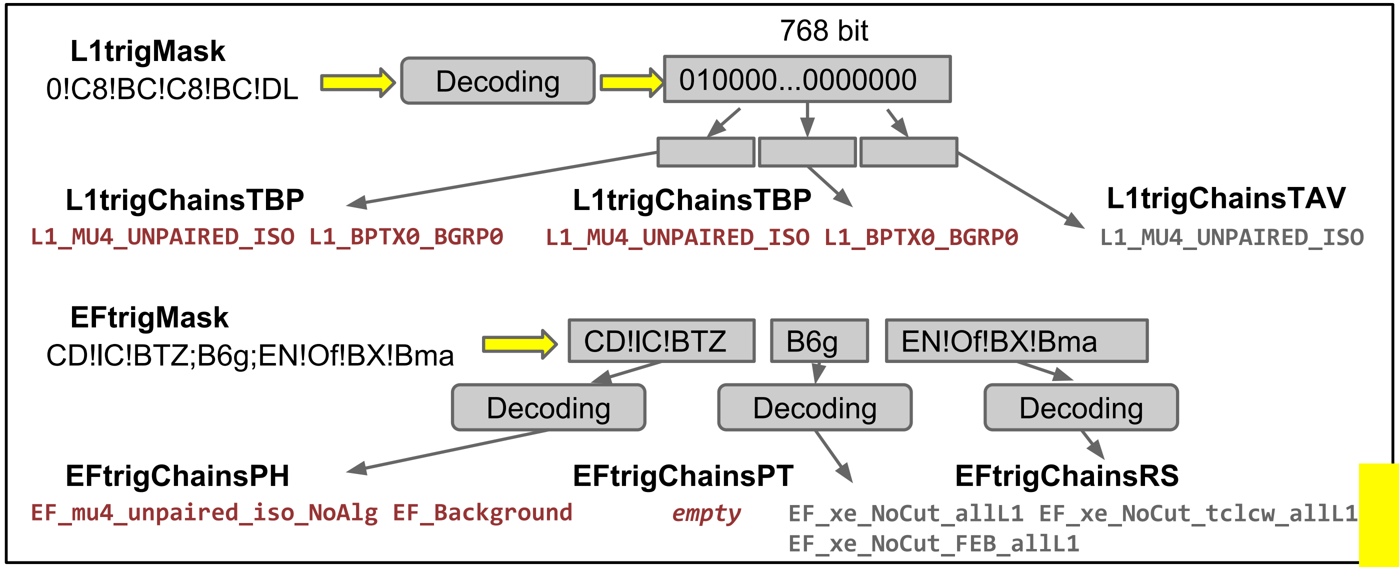
Progress map : 100%

More info in the poster by J. Hřivnáč (contribution #221)

Trigger Decoding

- Event-wise trigger decisions are stored natively in bit masks
- Trigger bit to name mapping is imported into HBase from the Conditions Metadata (COMA) database
- This makes decoded trigger decisions available to Event Index users by trigger name for event counting or selection



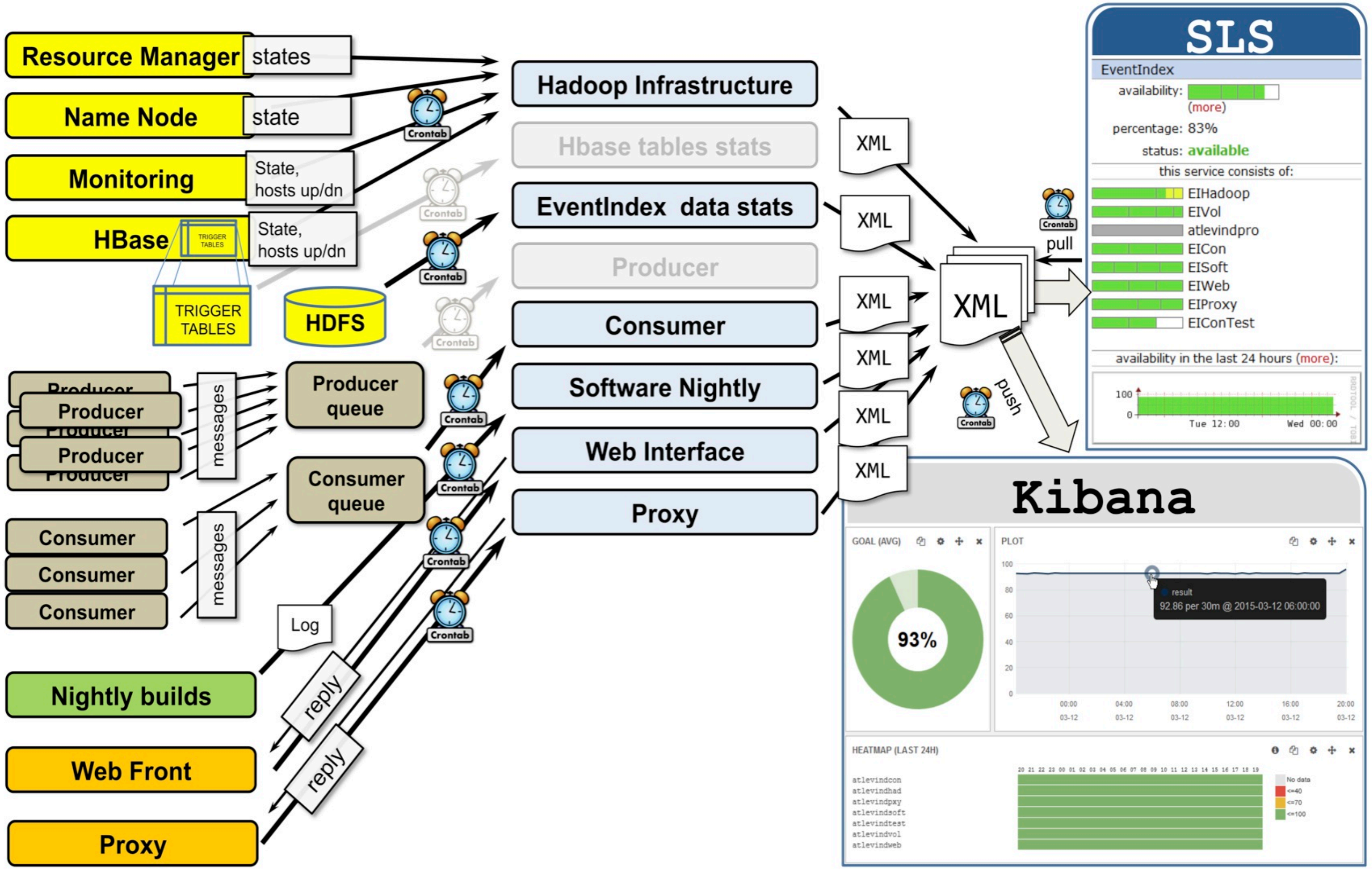


- To facilitate searches the names of fired triggers per event are stored in Hadoop
- In addition, the data may be indexed by trigger to improve performance

More info in the poster by F. Prokoshin (contrib. #220)

System Monitoring

- Monitors the health of all servers and processes involved in the chain:
  - ActiveMQ brokers and Consumers
  - Hadoop cluster and Web servers
- Contents monitoring under development



availability: (more)

percentage: 83%

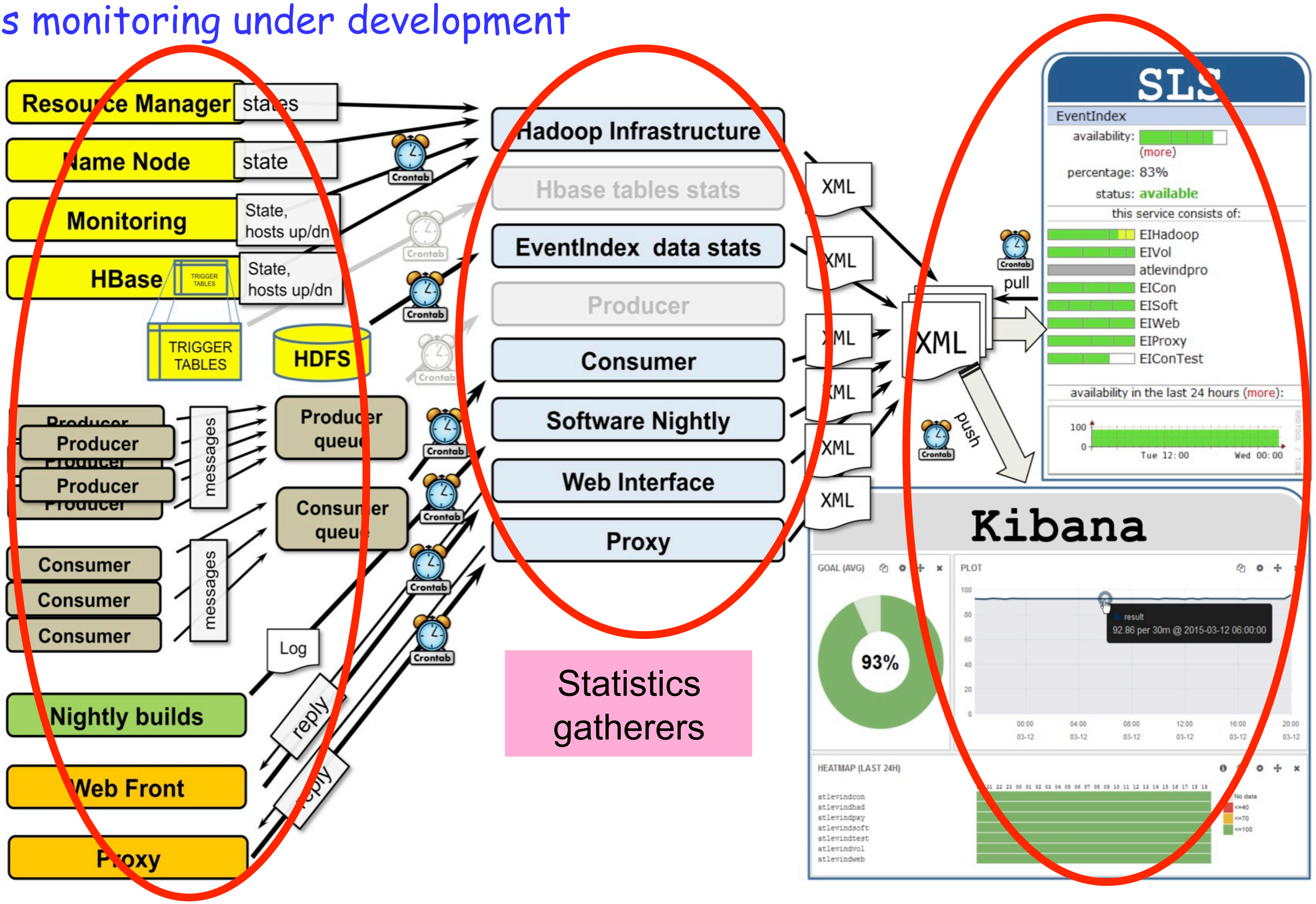
status: available

result
92.86 per 30m @ 2015-03-12 06:00:00

System Monitoring

- Monitors the health of all servers and processes involved in the chain:
  - ActiveMQ brokers and Consumers
  - Hadoop cluster and Web servers
- Contents monitoring under development

More info in the poster by F. Prokoshin (contrib. #220)



EventIndex
availability: (more)
percentage: 83%
status: available
this service consists of:
EIHadoop
EIVol
atlevindpro
EICon
EISoft
EIWeb
EIProxy
EIConTest
availability in the last 24 hours (more):

GOAL (AVG)
93%

PLOT
result
92.86 per 30m @ 2015-03-12 06:00:00

HEATMAP (LAST 24H)
atlevindcon
atlevindhad
atlevindpxy
atlevindsoft
atlevindtest
atlevindvol
atlevindweb

Servers and processes to be monitored

Statistics gatherers

Statistics displays

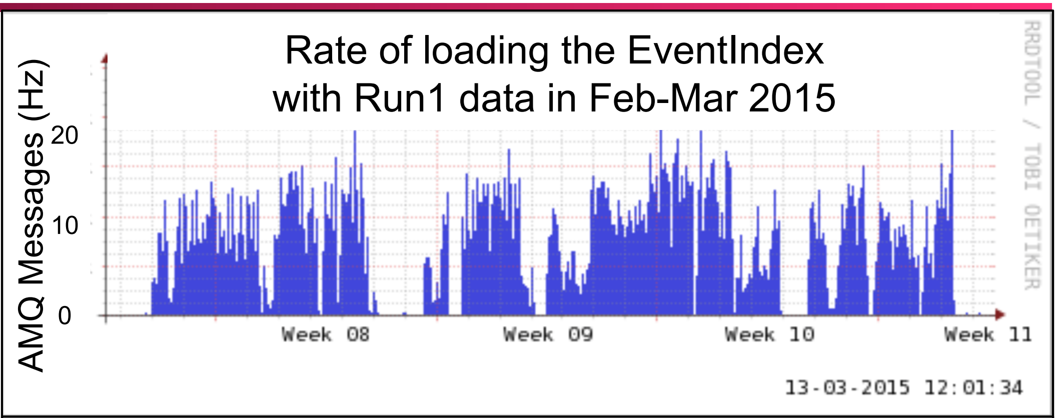
Development Status

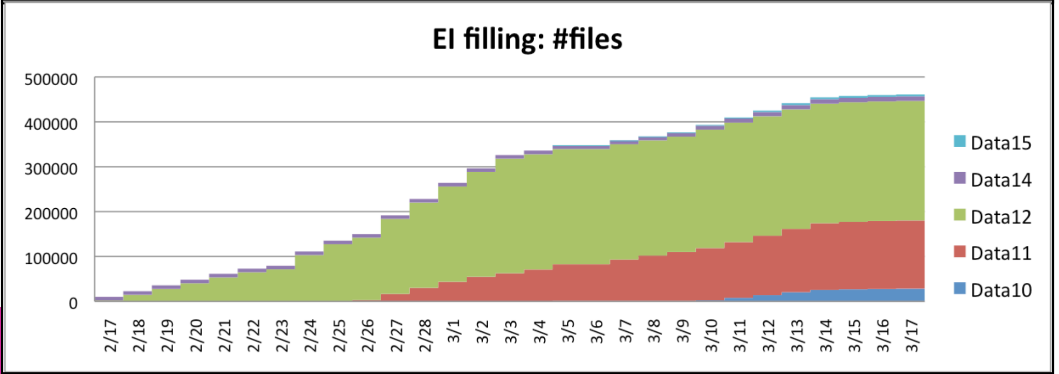
- All major components exist and work satisfactorily:
  - Data Collection: Producer transform runs at Tier-0 and on the Grid
  - Data Collection: Consumer reads data from the ActiveMQ servers, validates them and stores to HDFS
  - Storage System: Data organisation in Hadoop and indexing in catalogue
  - Storage System: Trigger decoding interface
  - Query System: CLI and web interfaces. Also EventLookup for event picking
  - Monitoring: System level monitoring in the new CERN Kibana environment
- Currently working on
  - Further automation of the data flow
  - System interconnections and monitoring
  - Automatic checks of production completeness

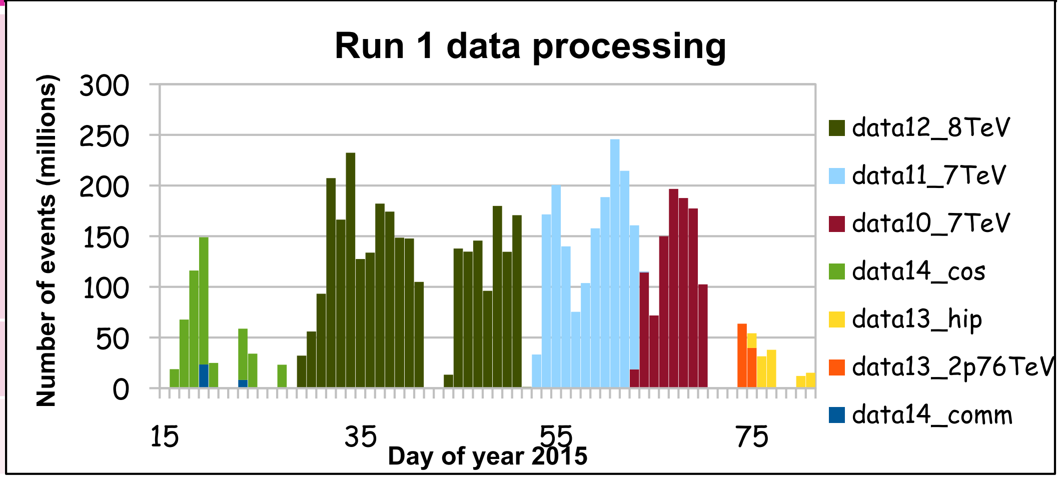
Deployment and Operation

- Run1 data processed since 1st February
  - Loaded all first-pass Tier-0 production (5.8 billion events)
  - Latest version of reprocessed data loading in progress
  - EventIndex data size in Hadoop: ~350 Bytes/event for real data before internal replication
- Message broker data occupancy kept under control using multiple consumers
- Automatic data reformatting and cataloguing in Hadoop
- Run2 data now flowing continuously from Tier-0 and available in real time

	Datasets	Files	Events	Tier-0 proc.	Reproc.
data10	3509	69526	1.0 G	✔	in progress
data11	3529	152492	1.7 G	✔	
data12	4190	267365	2.8 G	✔	
Heavy Ions	502	175198	0.3 G	✔	
data14	492	27374	0.4 G	✔	
data15	← in progress →				







Summary and Outlook

- The EventIndex infrastructure that was designed, developed and deployed in 2012-2015, is now in operation
  - Run1 Tier-0 processing data fully indexed
  - Run1 reprocessed data being filled in
  - Run2 new data indexed in real time
- Initial use cases all satisfied with good performance
- Work continuing on system optimisation and increased functionality
  - Automatic data validation
  - Robustness against network problems and hardware failures
  - Recording "offline trigger" (data derivation framework) stream counts and overlaps
  - Additional internal monitoring
  - Performance (timing) improvements for common queries