Atlas Event Index – WP2 Data collection and Retrieval. An update

Álvaro Fernández, Santiago González, Javier Sánchez

Instituto de Física Corpuscular (IFIC)
Universitat de València - CSIC

Producer code changes

00-02-07 → 00-02-08

- Allow input types: 'POOL', 'AOD', 'ESD', 'EVNT', 'HITS', 'RDO' and 'AOD_MRG'
- Add changes to allow EI to be imported as a substep
- .AODMergeEI_tf.py: Merge + EventIndex in one pass. Do not use in production
- sendEI.py: add GUID in message header (zero events DAOD file processing).

Producer code changes

<u>00-02-08 → 00-02-09</u>

- sendEI.py: modify logic to allow multiple file processing

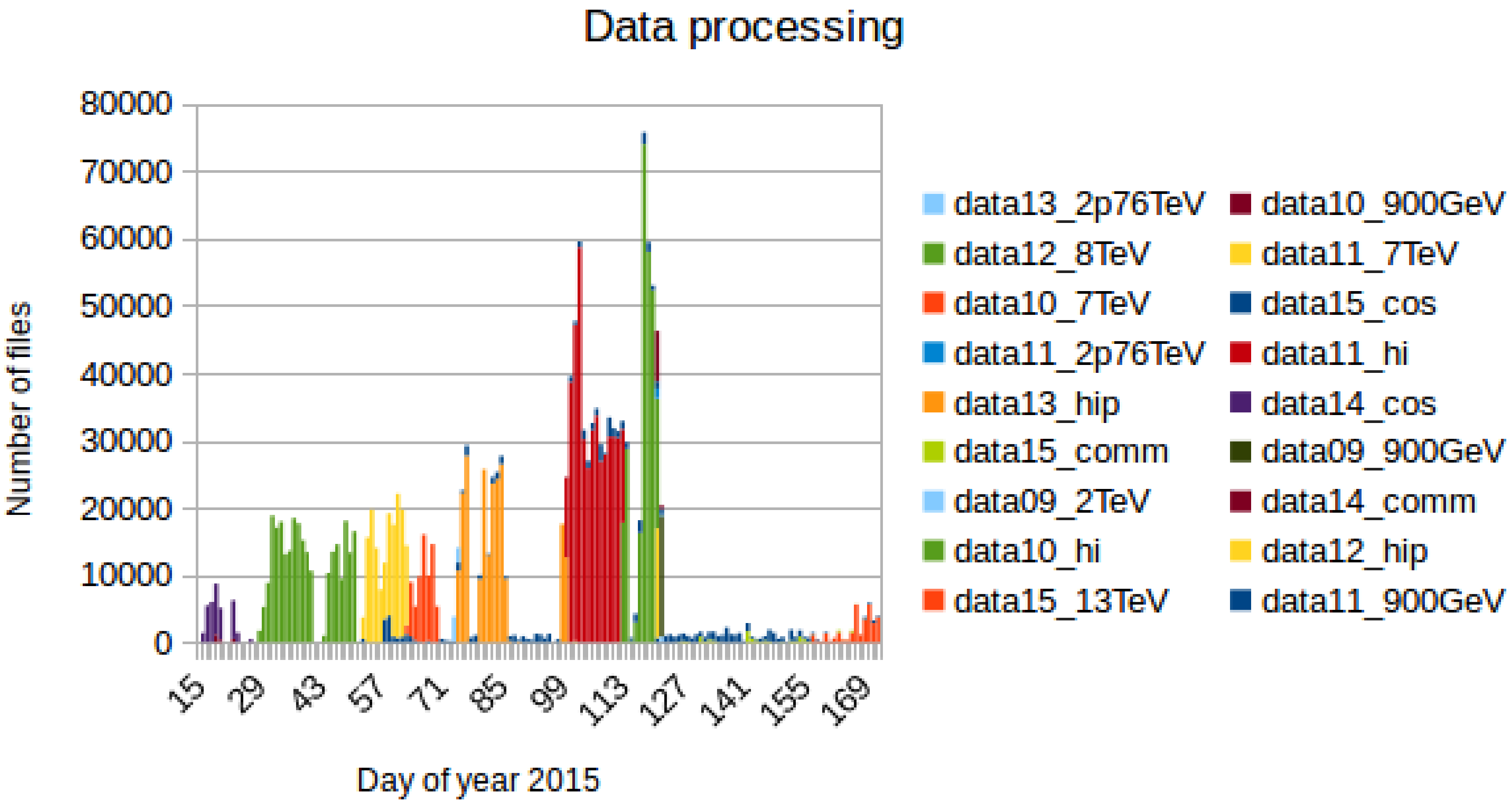
<u>00-02-09 → 00-02-10</u>

- added workaround for reading coral::Attribute::data<string> for ROOT6

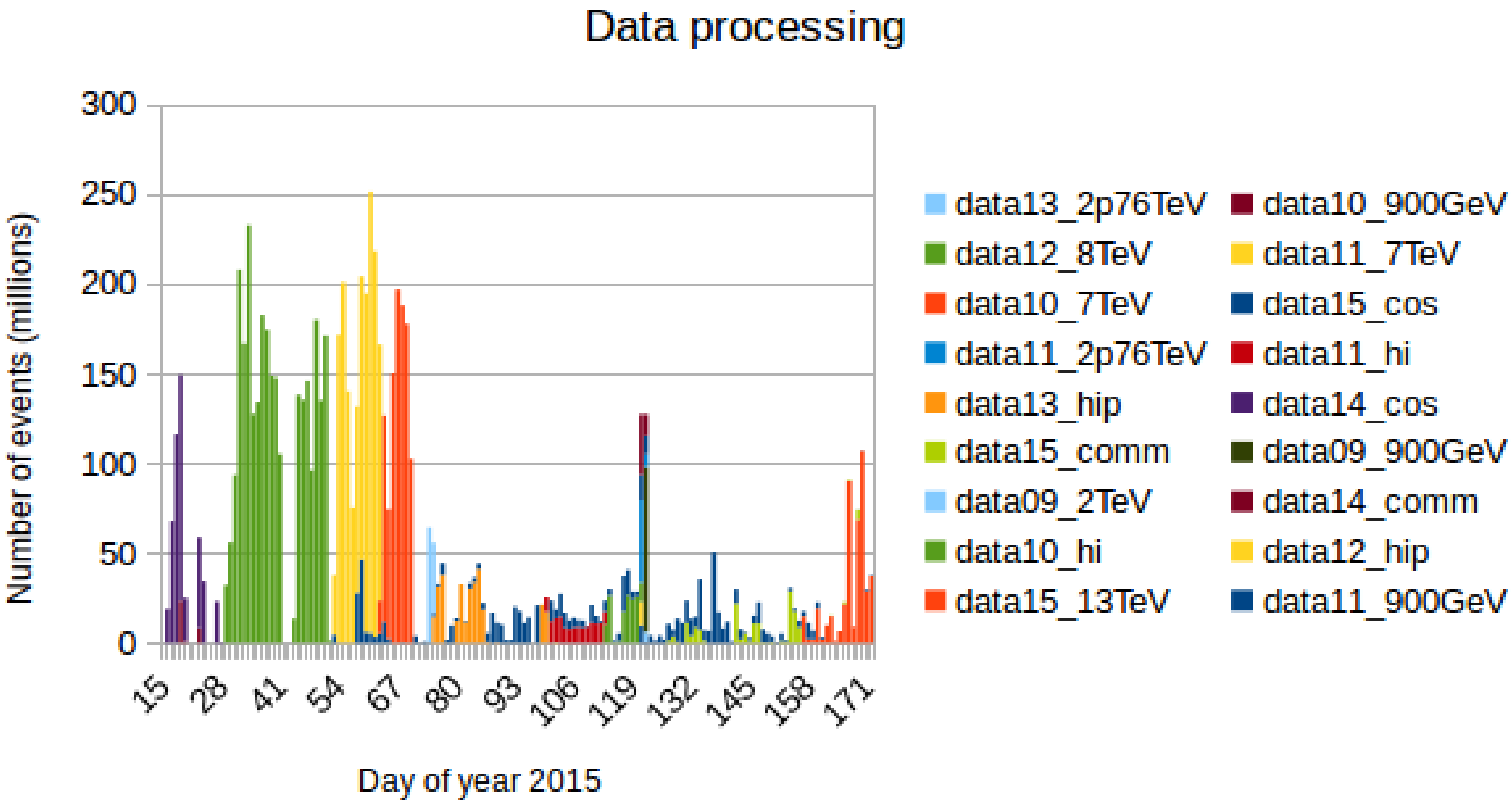
TODO

- xAOD processing
- MC
- Add persistency to messages
- Possible python messaging library upgrade

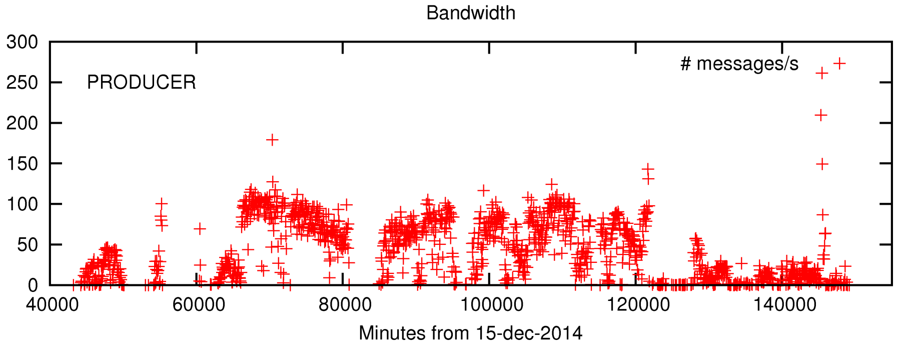
Data processing

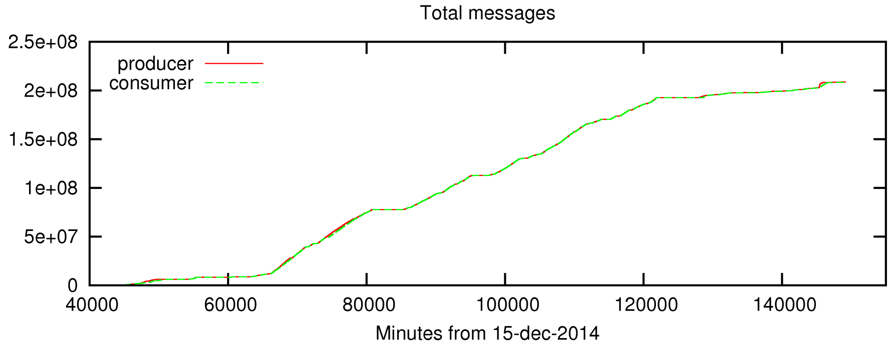


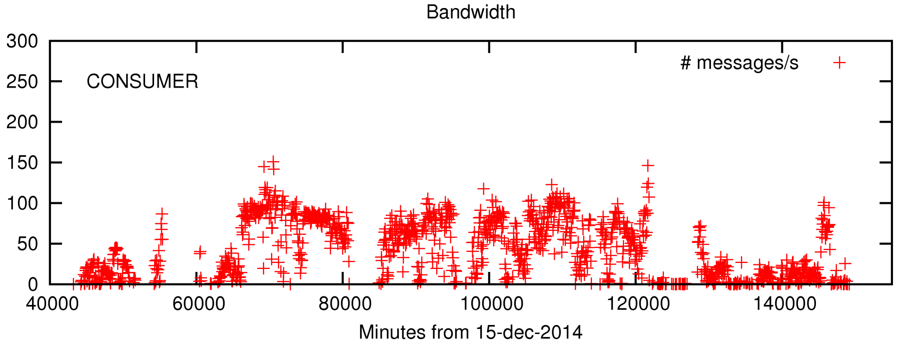
Data processing

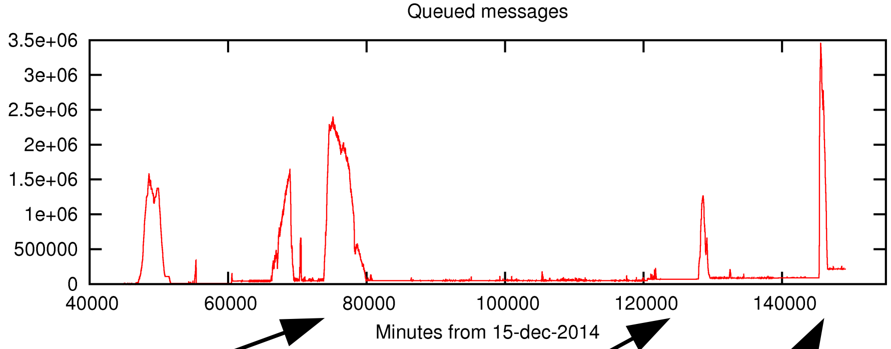


Performance









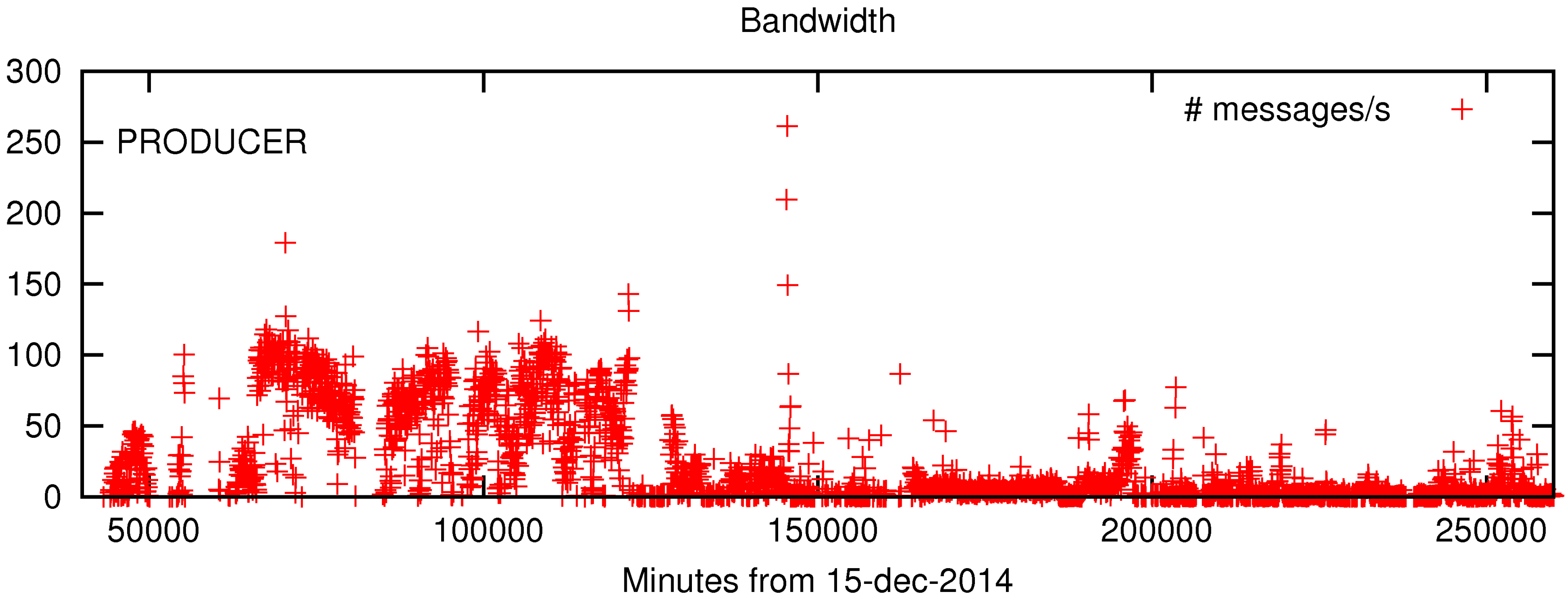
Some backlog due to consumer performance
Improved later with a refinement of writing algorithm

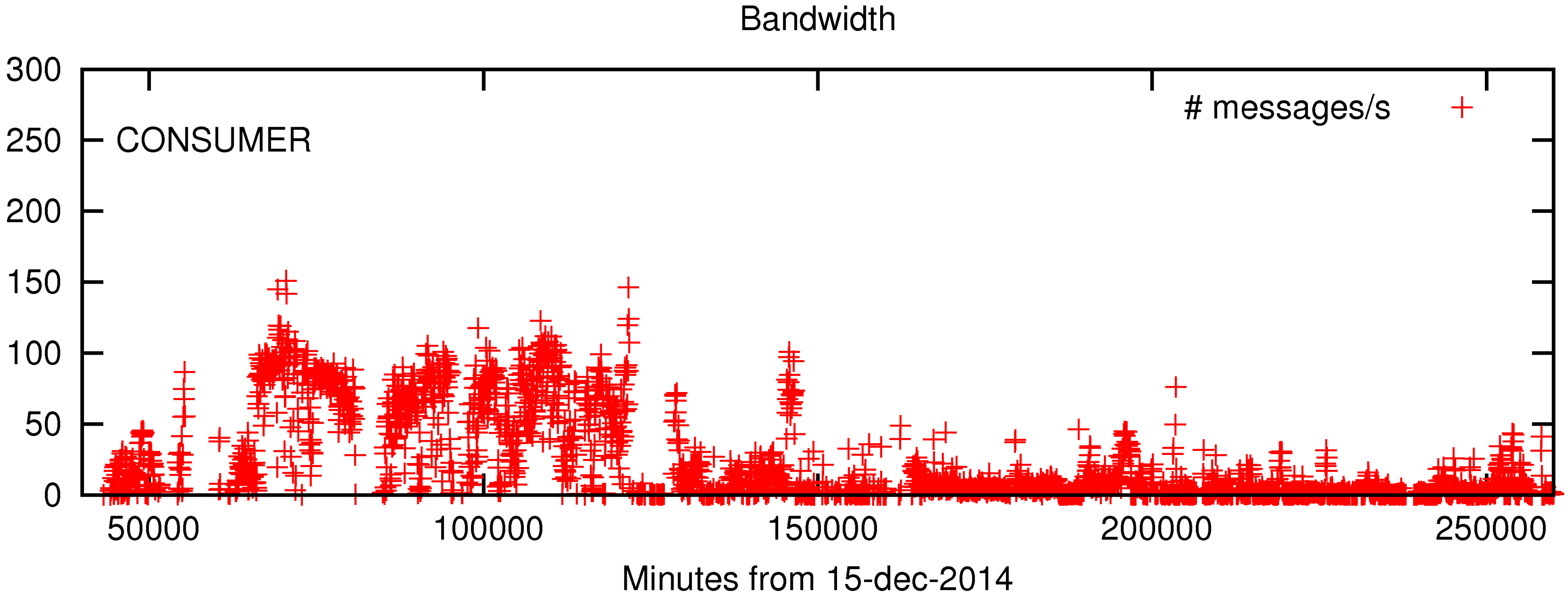
15 mar

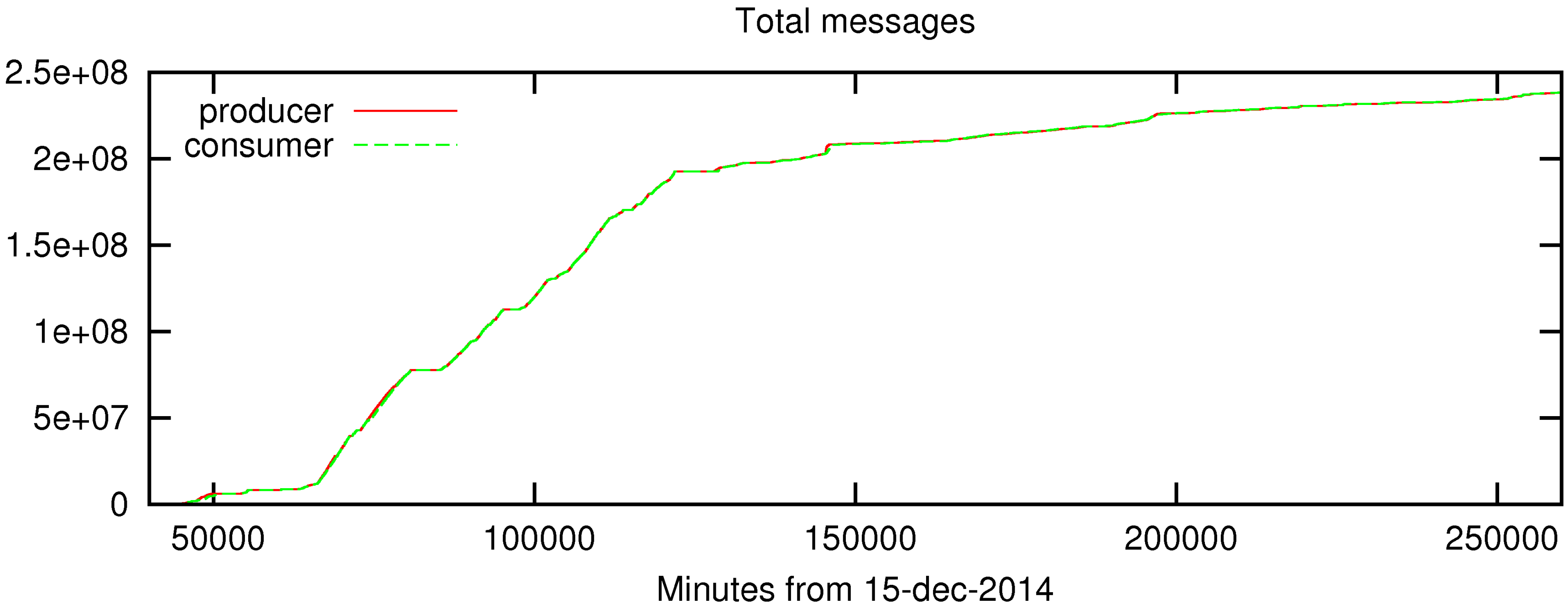
Problems with aiatlas017

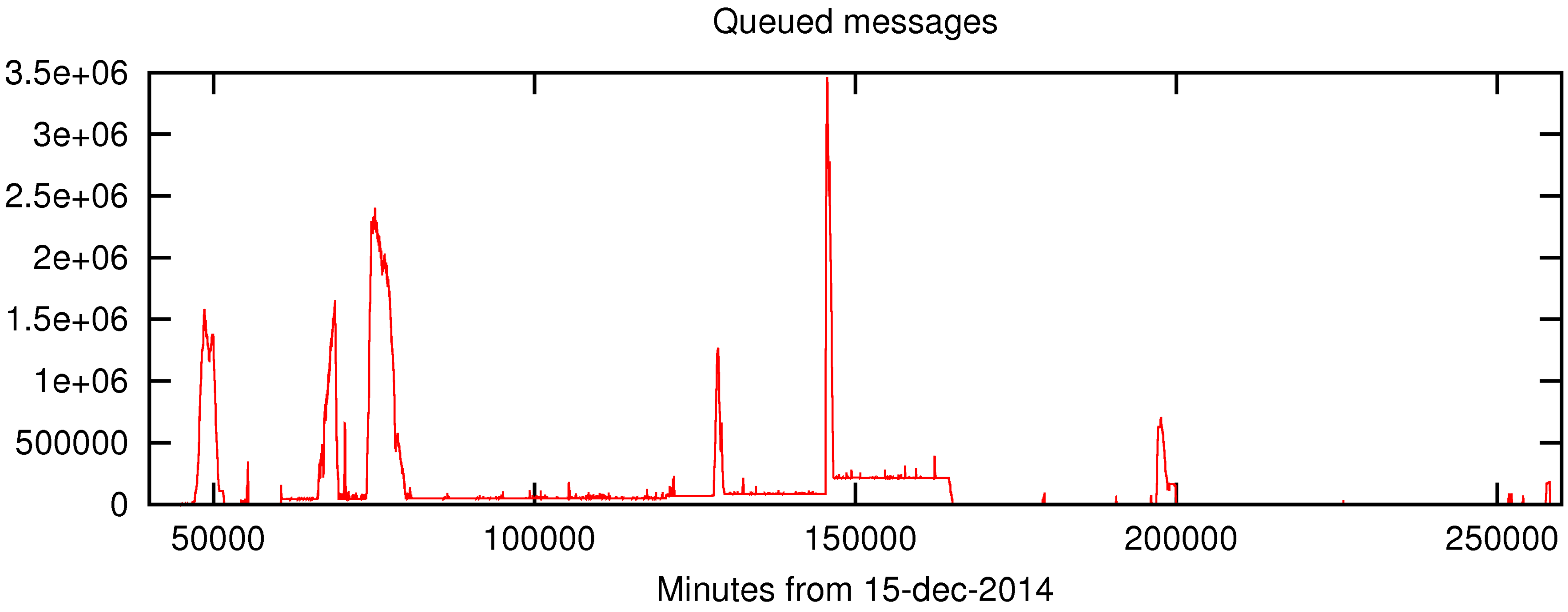
28 mar

High rate from producers

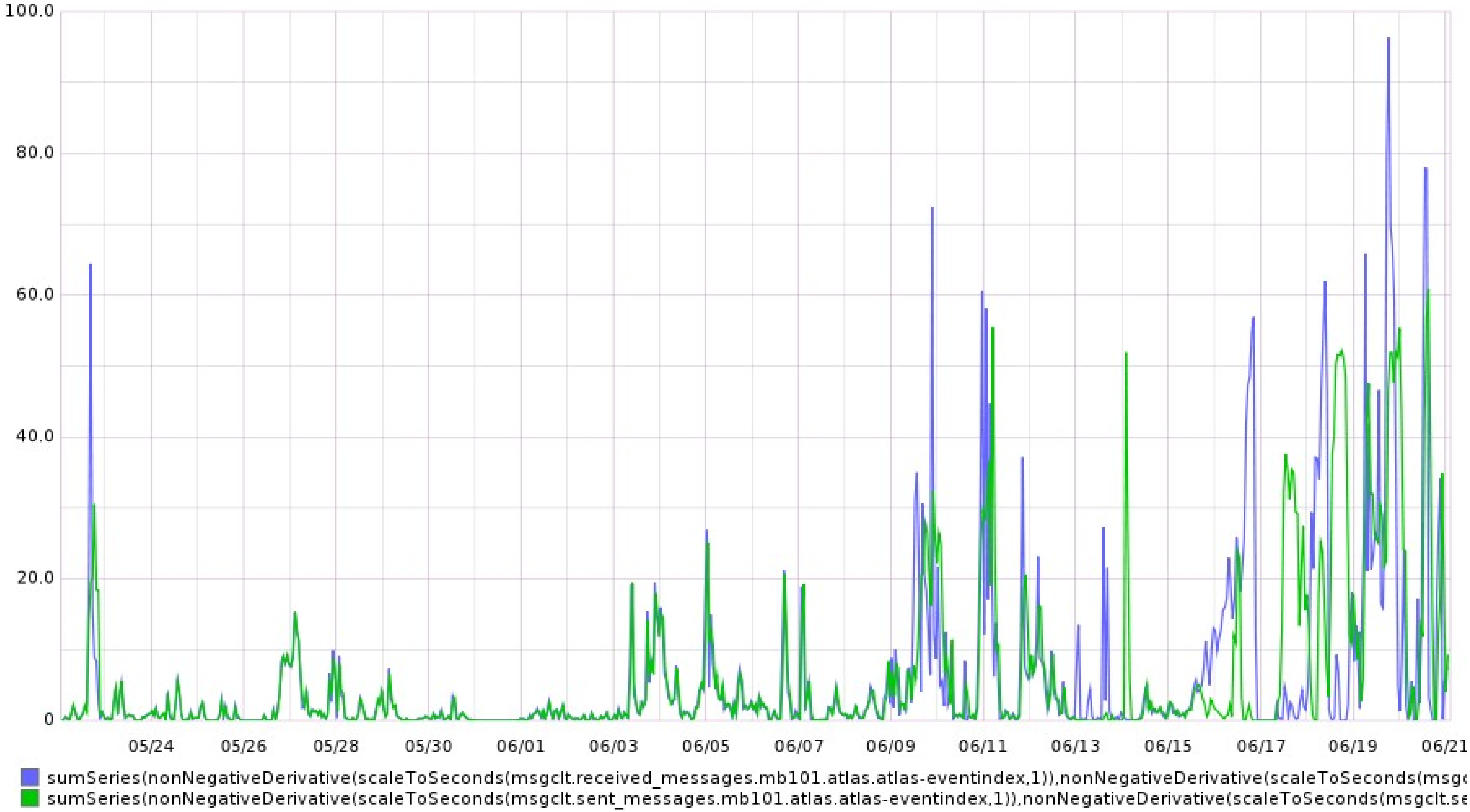

Bandwidth
PRODUCER
# messages/s
0
50
100
150
200
250
300
50000
100000
150000
200000
250000
Minutes from 15-dec-2014


Bandwidth
CONSUMER
# messages/s
0
50
100
150
200
250
300
50000
100000
150000
200000
250000
Minutes from 15-dec-2014

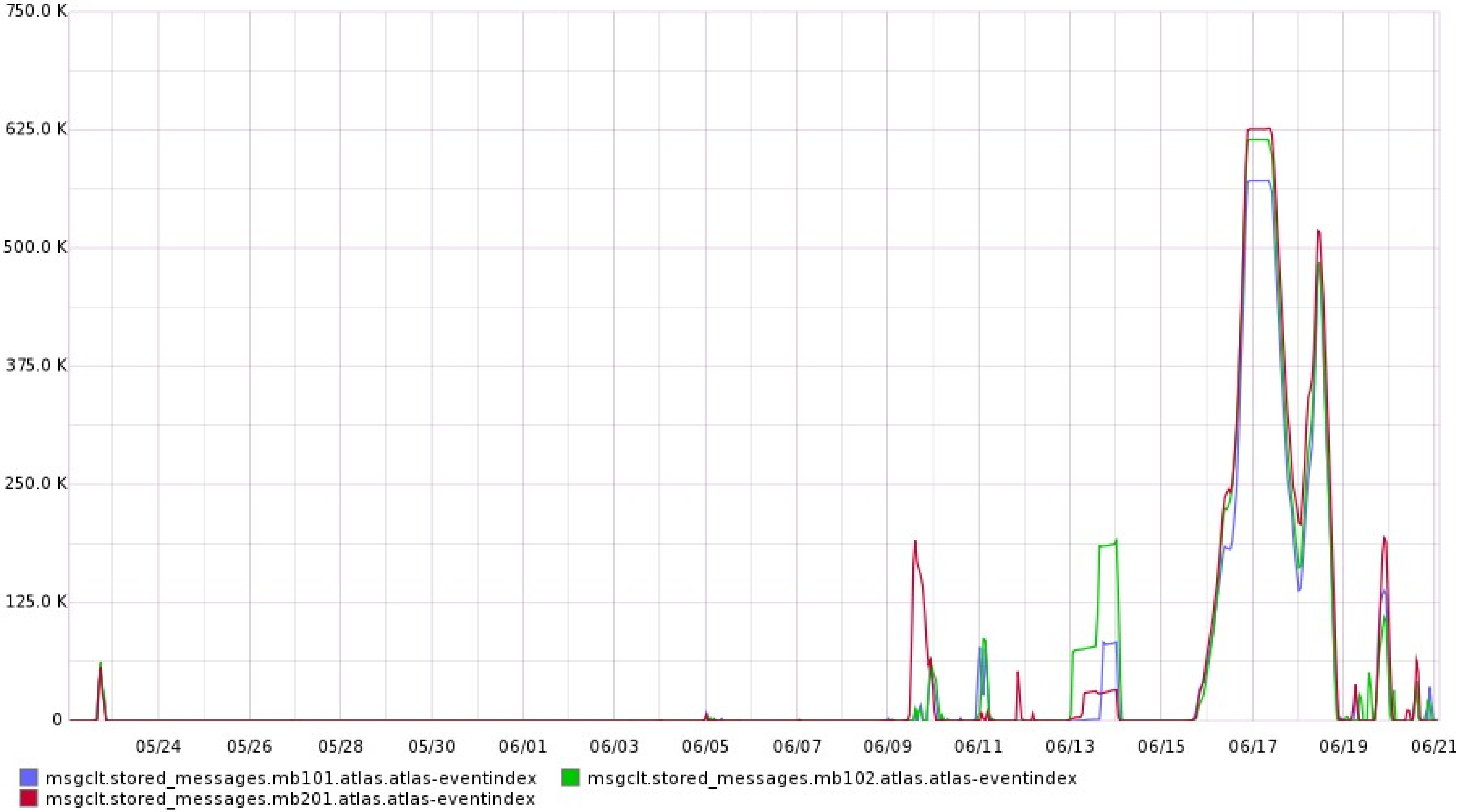

Total messages
producer
consumer
2.5e+08
2e+08
1.5e+08
1e+08
5e+07
0
50000
100000
150000
200000
250000
Minutes from 15-dec-2014


Queued messages
3.5e+06
3e+06
2.5e+06
2e+06
1.5e+06
1e+06
500000
0
50000
100000
150000
200000
250000
Minutes from 15-dec-2014

Rate last month



Unconsumed messages@ brokers



Monitoring

- Remember we produce statistic messages:
- Producer
    - Guid, ProducerId, StartTime, EndTime, #Events, #Messages, #Bytes
- Consumer:
    - every processed GUID:
        - Guid, ConsumerId, StartTime, EndTime, #Events, #Messages, #Bytes,
    - every 60 seconds, summing up processed data:
        - ConsumerId, StartTime, EndTime, #Events, #Messages, #Bytes

Monitoring

- Could be used in the “monitoring service”:
  - Trace data (by GUID), get traverse time from production to storage in HDFS, ...
- If needed, we could produce other Messages:
  - (ie:)Every validated Dataset: final time when data is validated

Declare missing or invalid data

- Failures happens. Rate is small, but some times data might not be received correctly, ie. past incidents:
  - Data: Negative EventNumbers due to initial out of range
  - Consumer: Hdfs/HBase failures, out of order messages in the past
  - Brokers: Failure/Restart of the broker missing stored messages.
- We try to recover right data from received one whenever possible.
- If not possible, we re-send manually new (grid) jobs to process data again.
- But we need an automatic way to declare missing or invalid data (detected after Validation).

Consumer: HBASE

- One table containing ALL events
- Row key: binary composite key:

guid **runNumber** **evtNumber**

- Count events doing scans over row key:

guid **00000000** **00000000** - **guid** **ffffffff** **ffffffff**

Consumer: HBASE

- One family: 'tk', contains token components (columns):

ct **cl** **t** **o1** **o2** **pv**

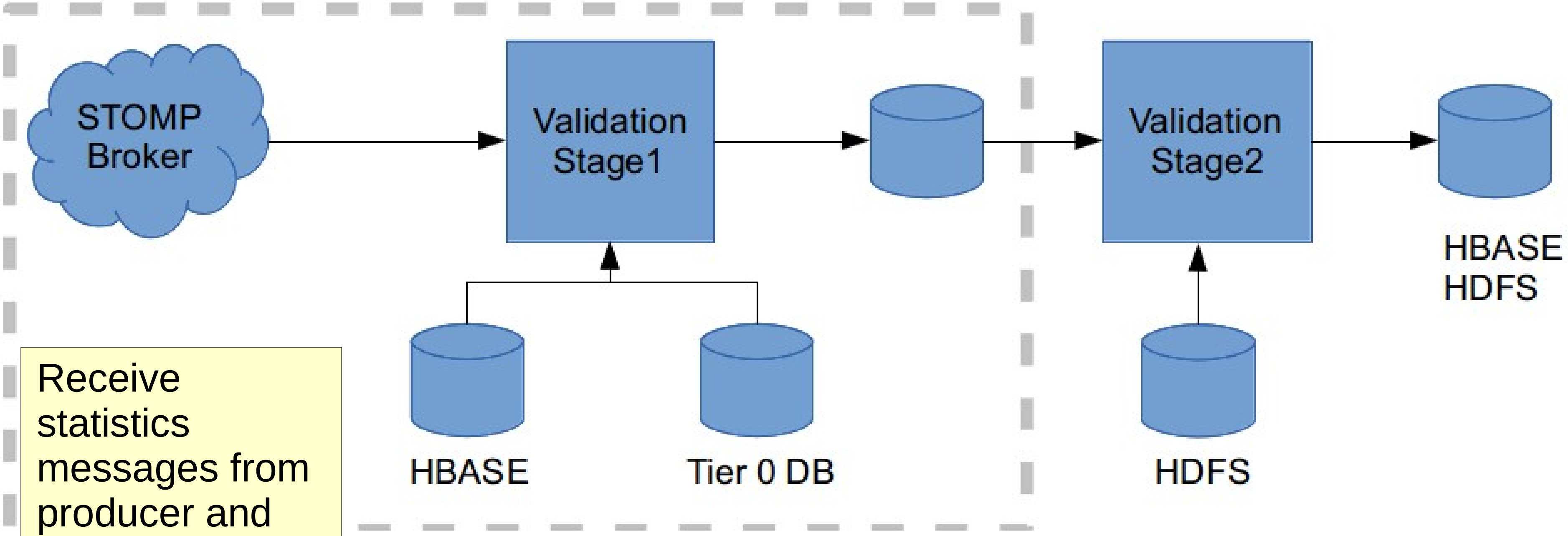
- Token:

[DB= **guid**][CNT= **ct**][CLID= **cl**]

[TECH= **t**][OID= **o1** - **o2**]

- **Added new column:**
  - Key to provenace entry **pv**

Validation



Receive statistics messages from producer and consumer.
Get GUIDs produced and consumed and number of events.

Get number of events per GUID

Get tasks (datasets) and number of jobs (files) from Tier 0 farm

Check files present on HADOOP

Signal Core to import data and make it available.