



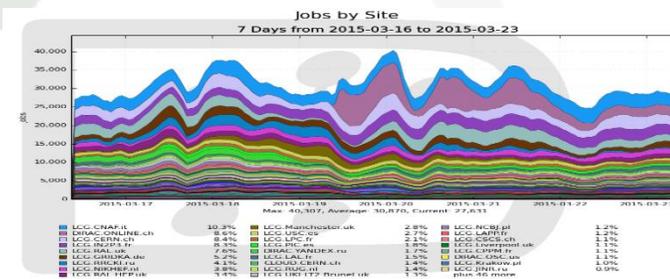
21st International Conference on Computing in High Energy and Nuclear Physics **CHEP2015** Okinawa Japan: April 13 - 17, 2015

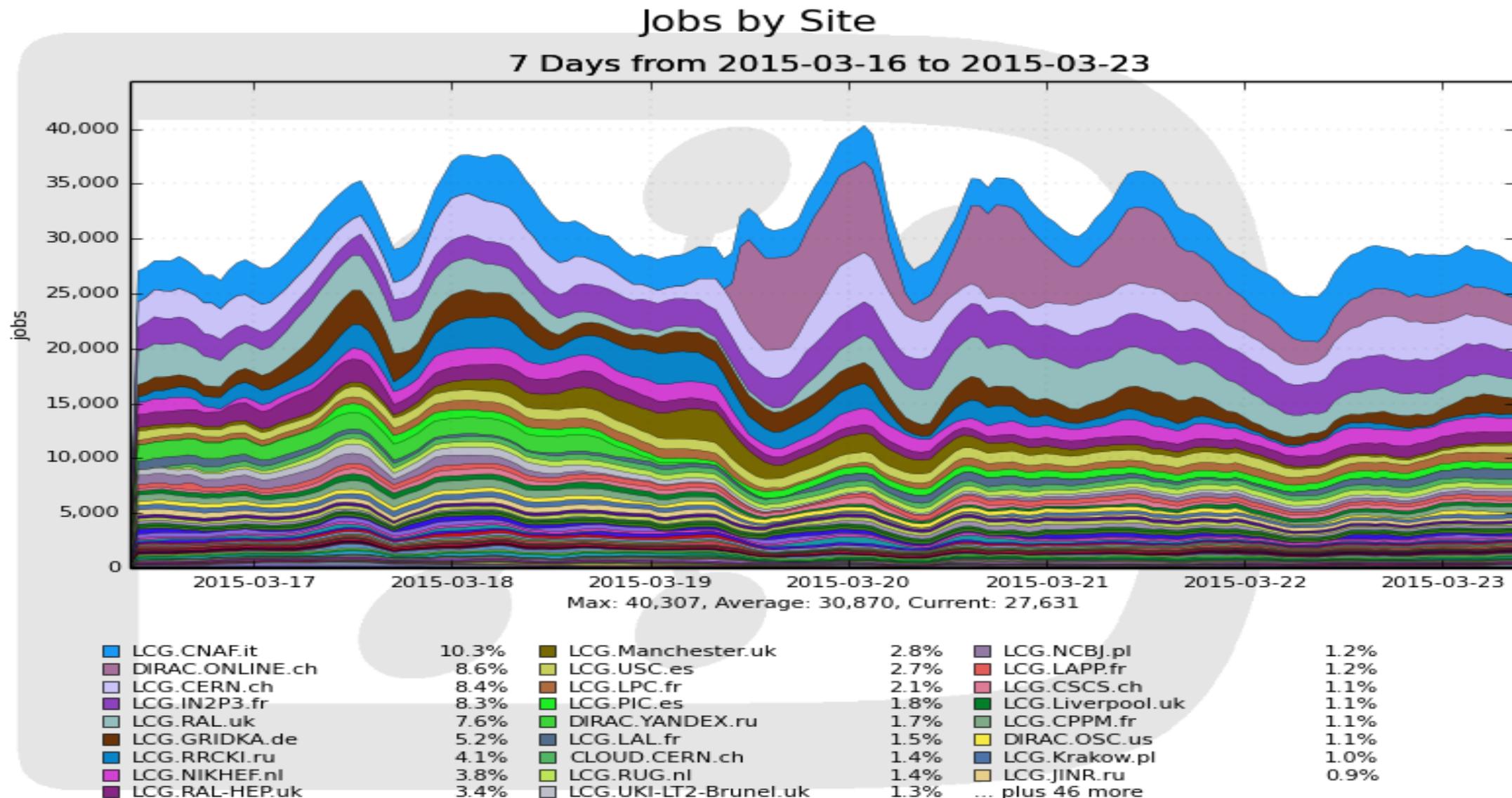


Evaluation of NoSQL databases for DIRAC monitoring and beyond

*Adrian Casajus Ramo, Federico Stagni, Luca
Tomassetti, Zoltan Mathe
On behalf of the LHCb collaboration*

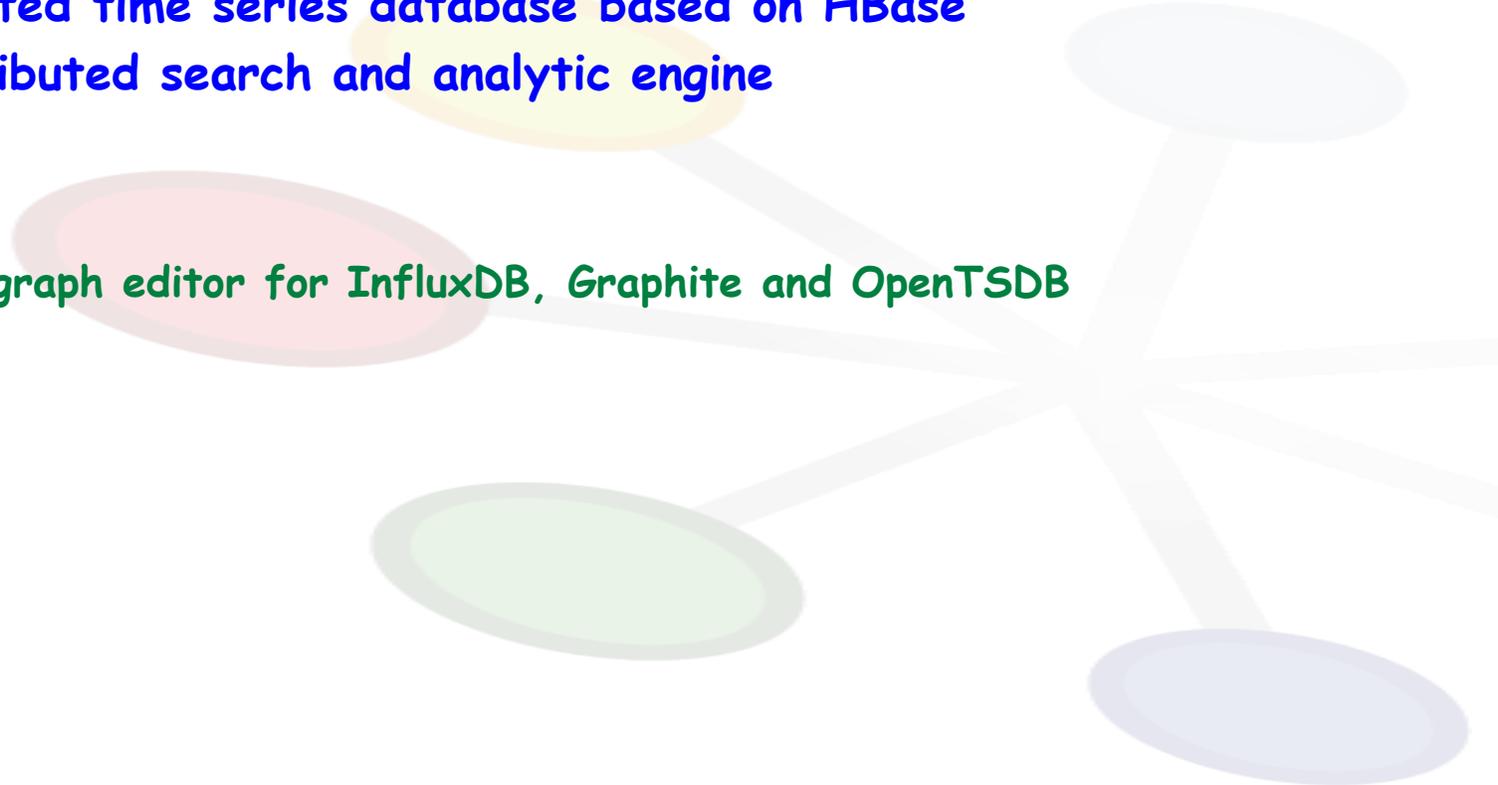
- Develop a system for real time monitoring and data analysis:
 - Focus on monitoring the jobs (not accounting)
- Requirements
 - Optimized for time series analysis
 - Efficient data storage, data analysis and retrieval
 - Easy to maintain
 - Scale Horizontally
 - East to create complex reports (dashboards)
- Why?
 - Current system is based on MySQL:
 - ↳ is not designed for real time monitoring (more for accounting)
 - ↳ does not scale to hundred of million rows (>500 million).
 - ↳ It requires ~400 second to generate a one-month duration plot
 - ↳ is not for real time analysis
 - ↳ is not schema-less:
 - ↳ Often change the data format



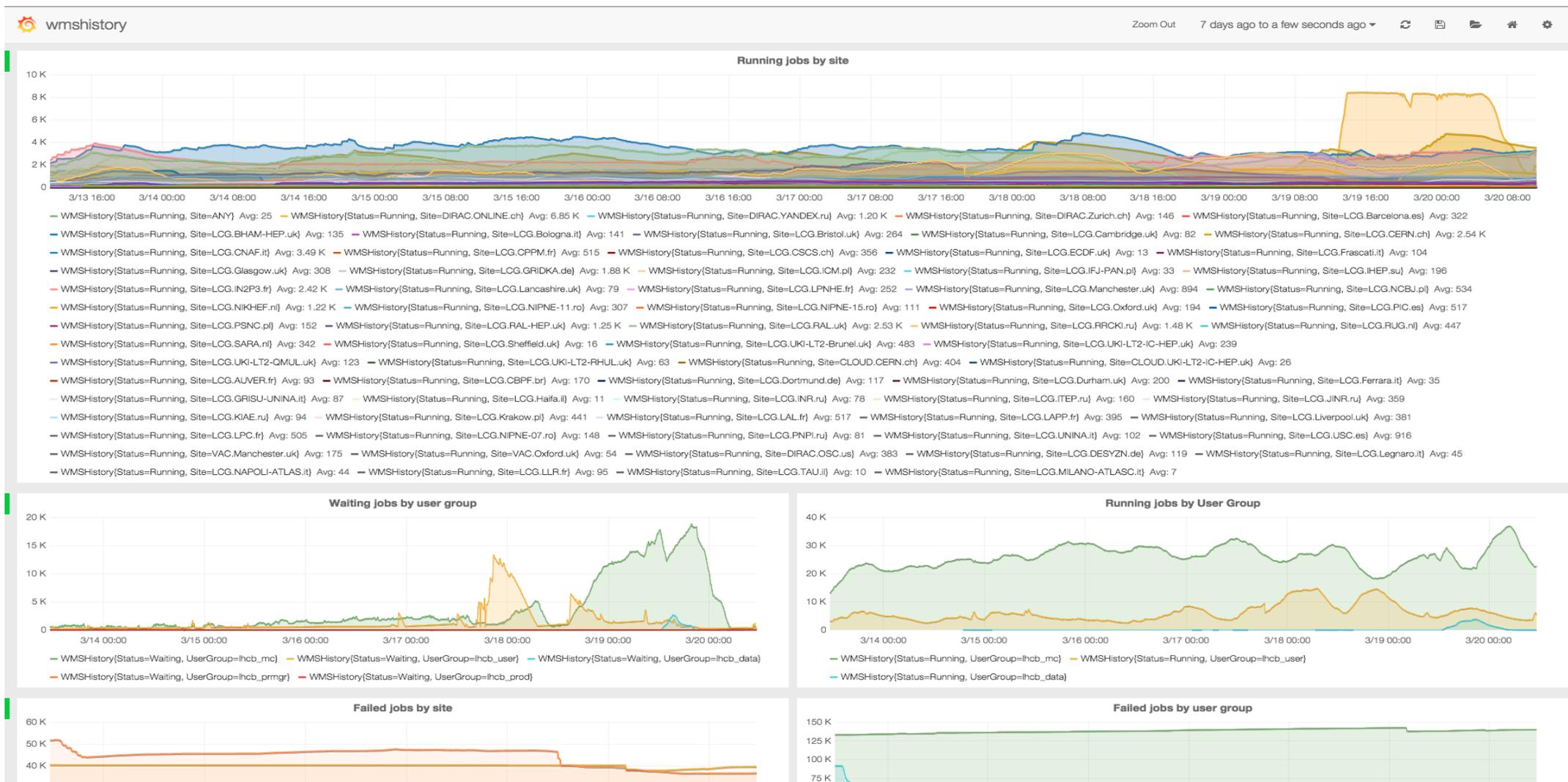


Generated on 2015-03-23 09:31:21 UTC

- Database:
 - InfluxDB is a distributed time series database with no dependency
 - OpenTSDB is a distributed time series database based on HBase
 - ElasticSearch is a distributed search and analytic engine
- Data visualization:
 - Grafana
 - ↳ Metric dashboard and graph editor for InfluxDB, Graphite and OpenTSDB



Grafana dashboard:



- Database:

- InfluxDB is a distributed time series database with no dependency
- OpenTSDB is a distributed time series database based on HBase
- ElasticSearch is a distributed search and analytic engine

- Data visualization:

- Grafana

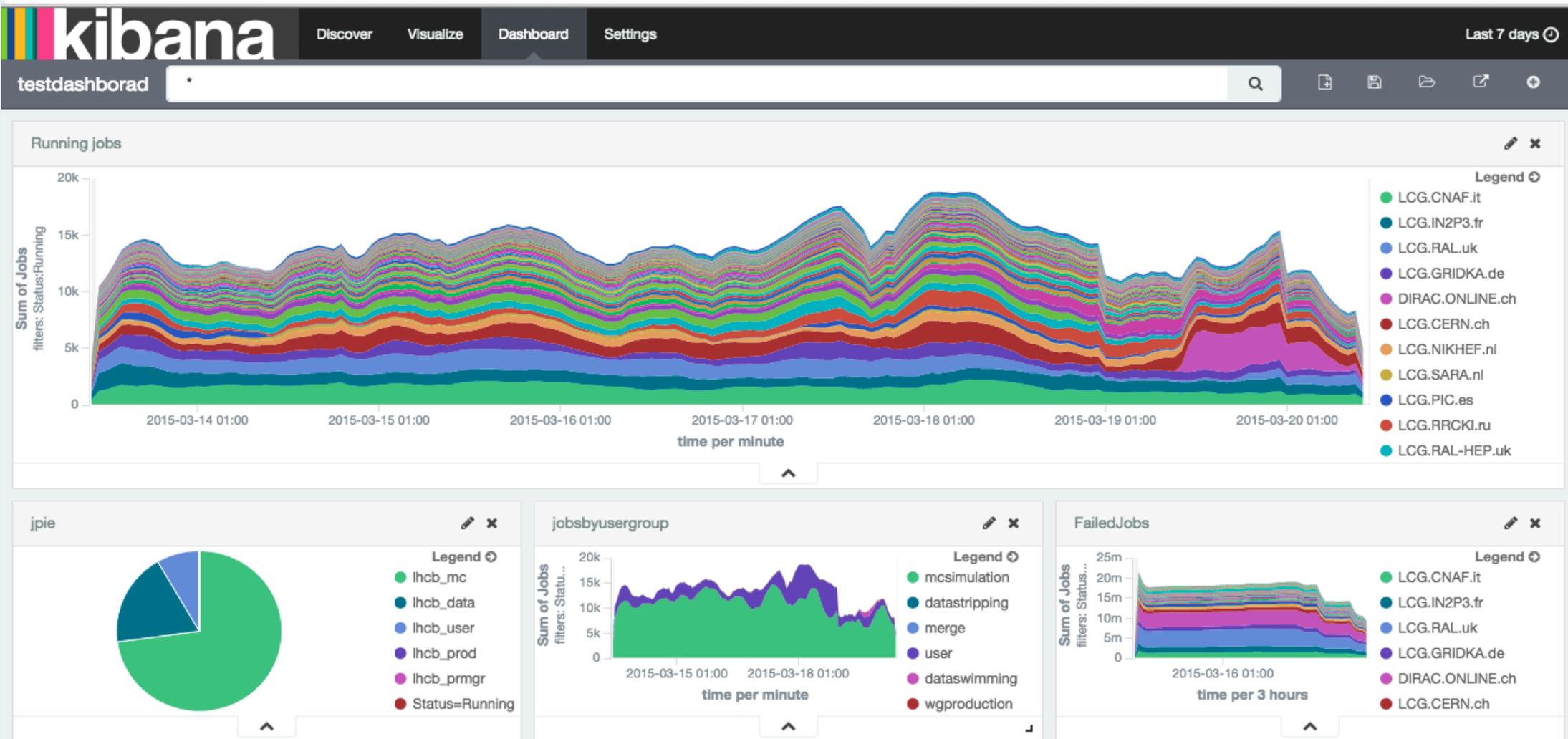
- ↳ Metric dashboard and graph editor for InfluxDB, Graphite and OpenTSD

- Kibana

- ↳ Flexible analytic and visualization framework

- ↳ Developed for creating complex dashboards

- Kibana dashboard:



- Database:

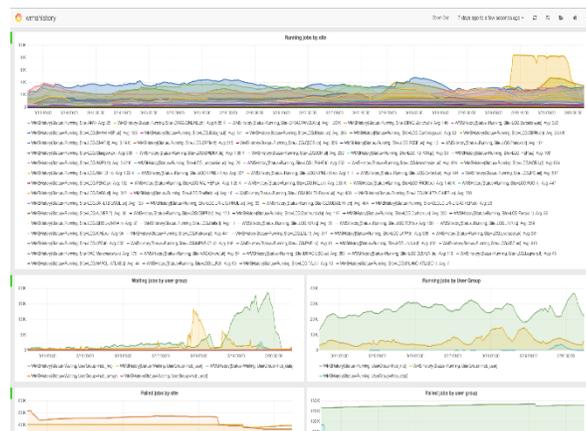
- InfluxDB is a distributed time series database with no dependencies
- OpenTSDB is a distributed time series database based on HBase
- ElasticSearch is a distributed search and analytic engine

- Data visualization:

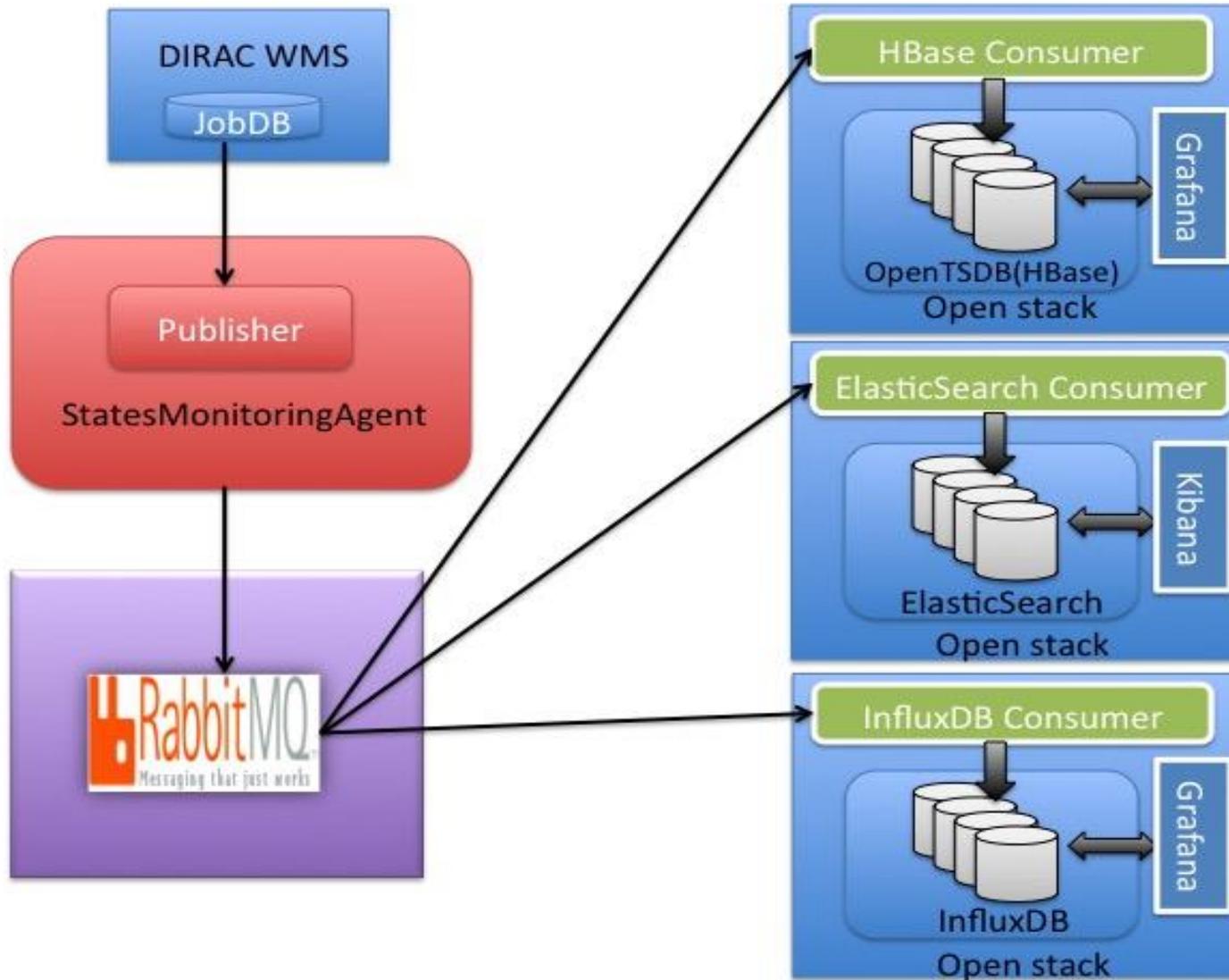
- Grafana
 - ↳ Metric dashboard and graph editor for InfluxDB, Graphite and OpenTSD
- Kibana
 - ↳ Flexible analytic and visualization framework
 - ↳ Developed for creating complex dashboards

- Communication

- RabbitMQ
 - ↳ Robust messaging system

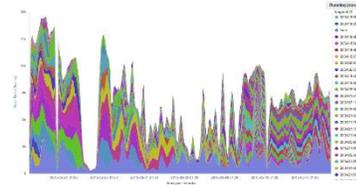


Overview of the System

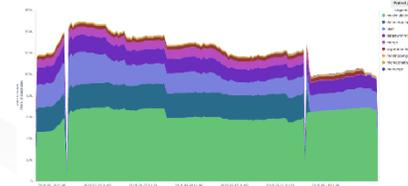
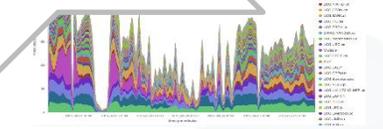
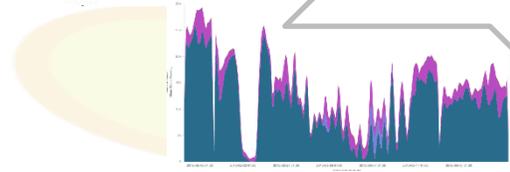


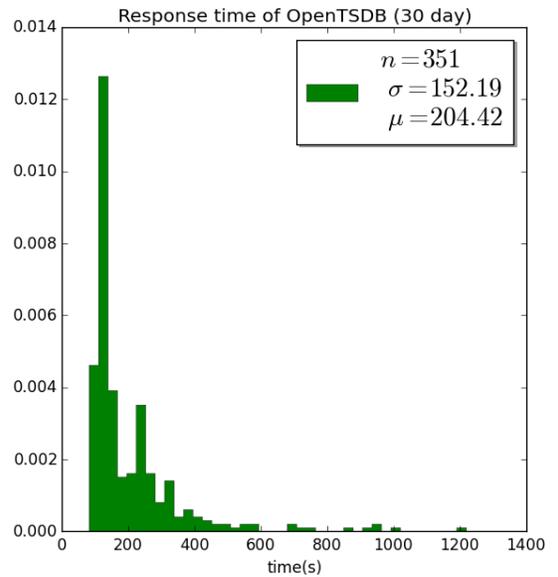
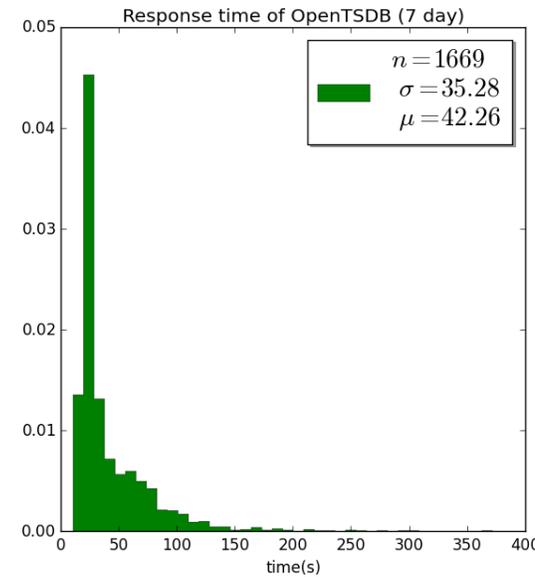
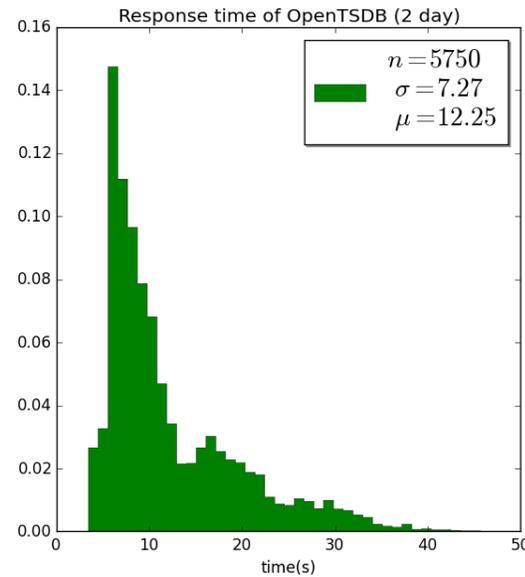
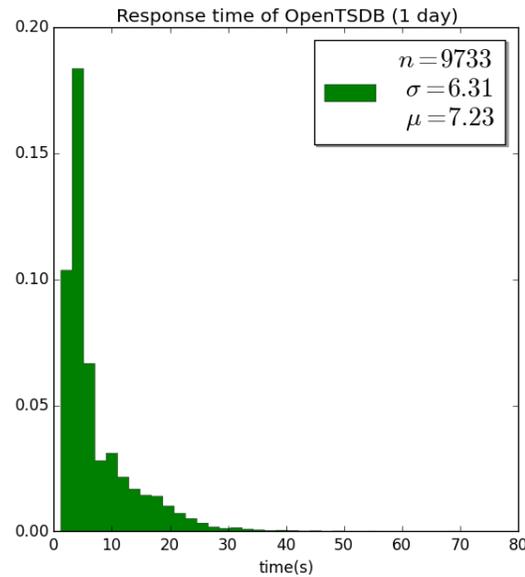
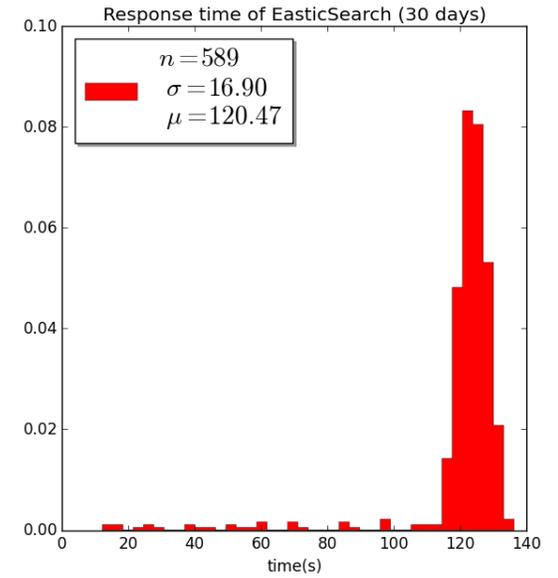
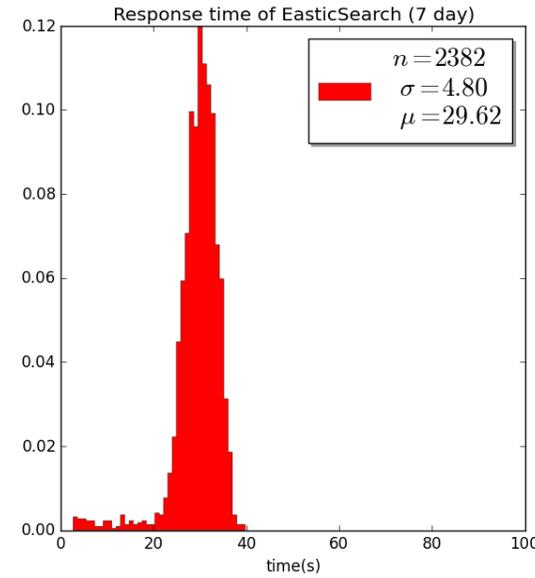
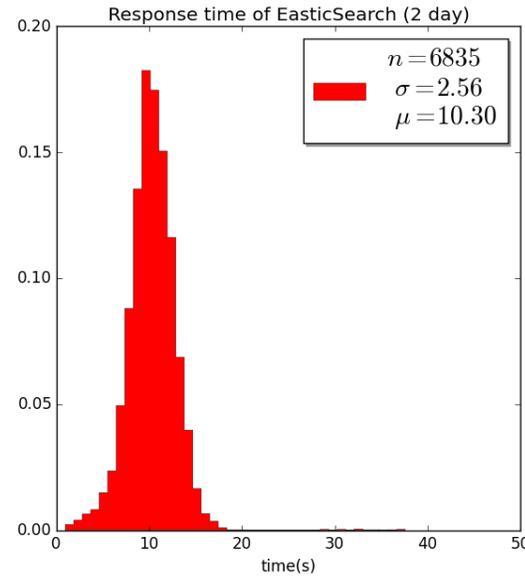
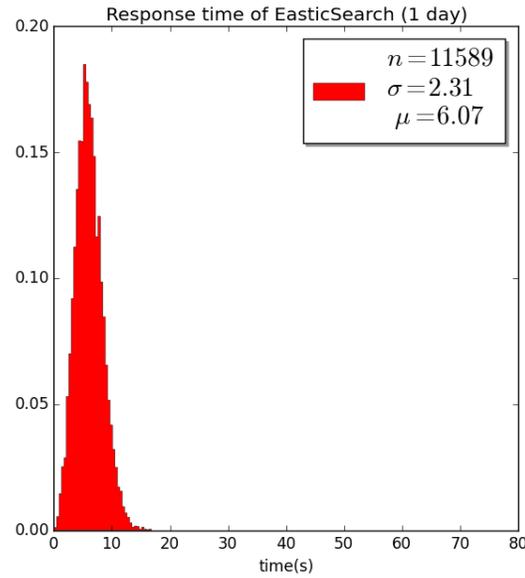
- RabbitMQ
 - one physical machine
- 12 VMs provided by CERN OpenStack
 - Each VM has 4 core, 8 GB memory and 80GB disk
 - We used 3 clusters with 4 nodes
- Data format:
 - The records are sent to the RabbitMQ in JSON format.
 - Each record must contain a minimum of four elements:
 - ℞ metric, time, key/value pairs, value
 - ℞ For example: `{"Status": "Done", "time": 1404086442, "JobSplitType": "MCSimulation", "MinorStatus": "unset", "Site": "ARC.Oxford.uk", "value": 10, "metric": "WMSHistory", "User": "phicharp", "JobGroup": "00037468", "UserGroup": "lhcb_mc"}`

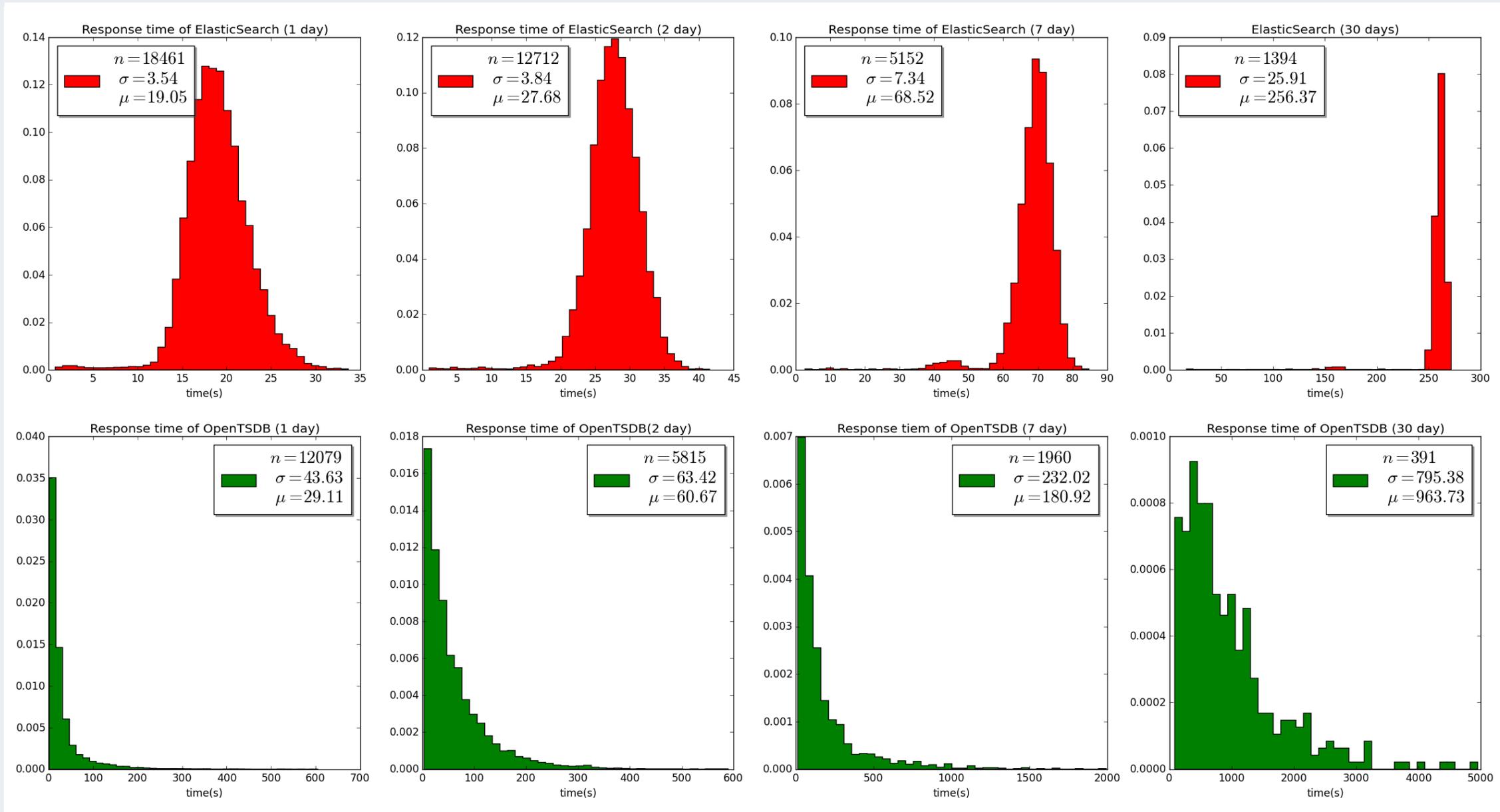
- We have recorded ~600 million records during ~1.5 month
- We defined 5 different queries
 - Running jobs grouped by Site
 - Running jobs grouped by JobGroup
 - Running jobs grouped by JobSplitType
 - Failed jobs grouped by JobSplitType
 - Waiting jobs grouped by JobSplitType
- Query intervals: 1, 2, 7 and 30 day
 - Random interval:
 - ↳ Start and end time are generated randomly between 2015-02-05, 15:00:00 and 2015-03-12 15:00:00
- The high workload is generated by 10, 50, 100 clients (python threads) to measure the response time and the throughput
 - REST APIs are used to retrieve the data from the DB
 - All clients are used a random query and a random period
 - All clients are continuously running parallel during 7200 second

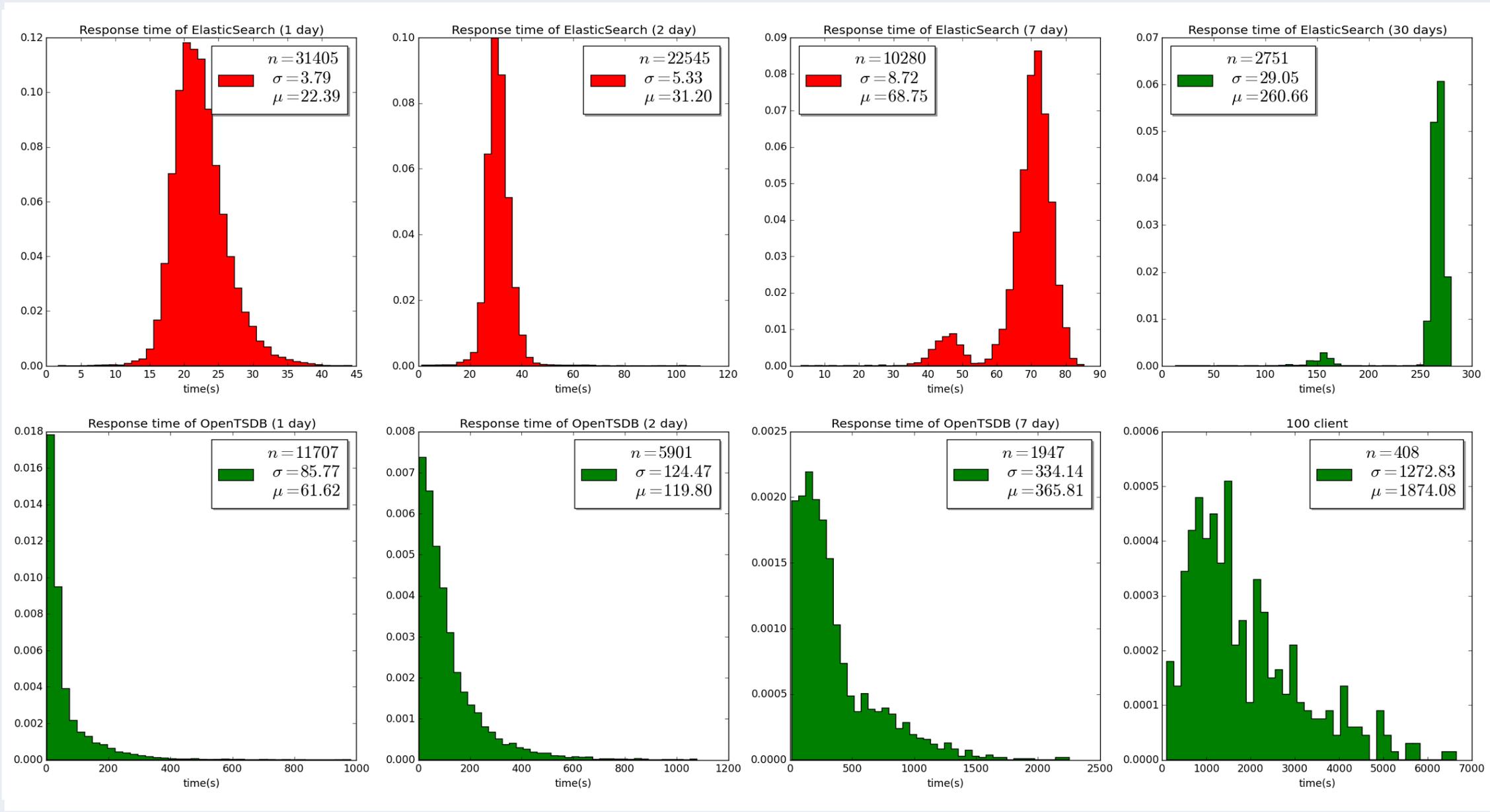


InfluxDB has not scaled after 2 days

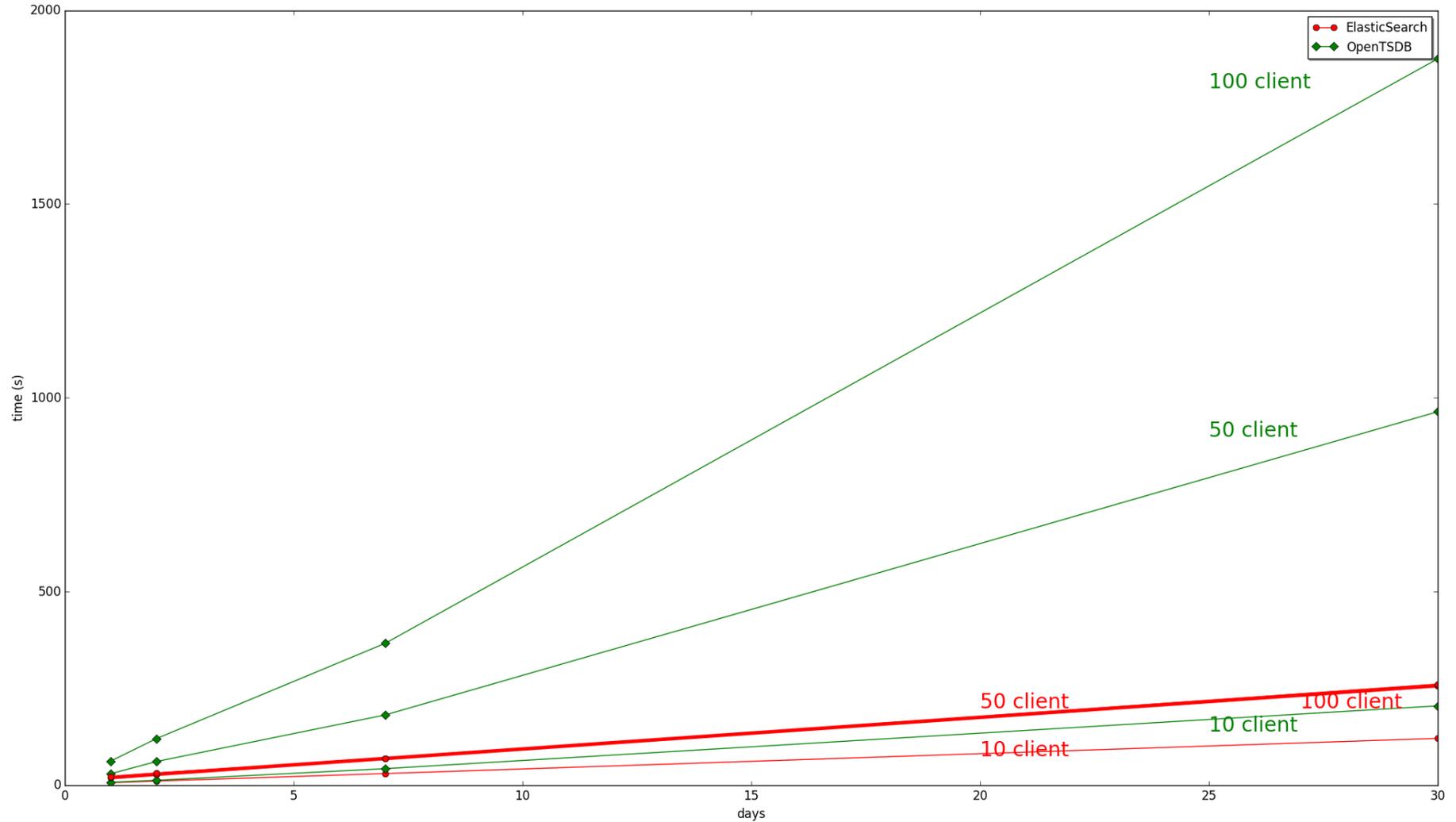




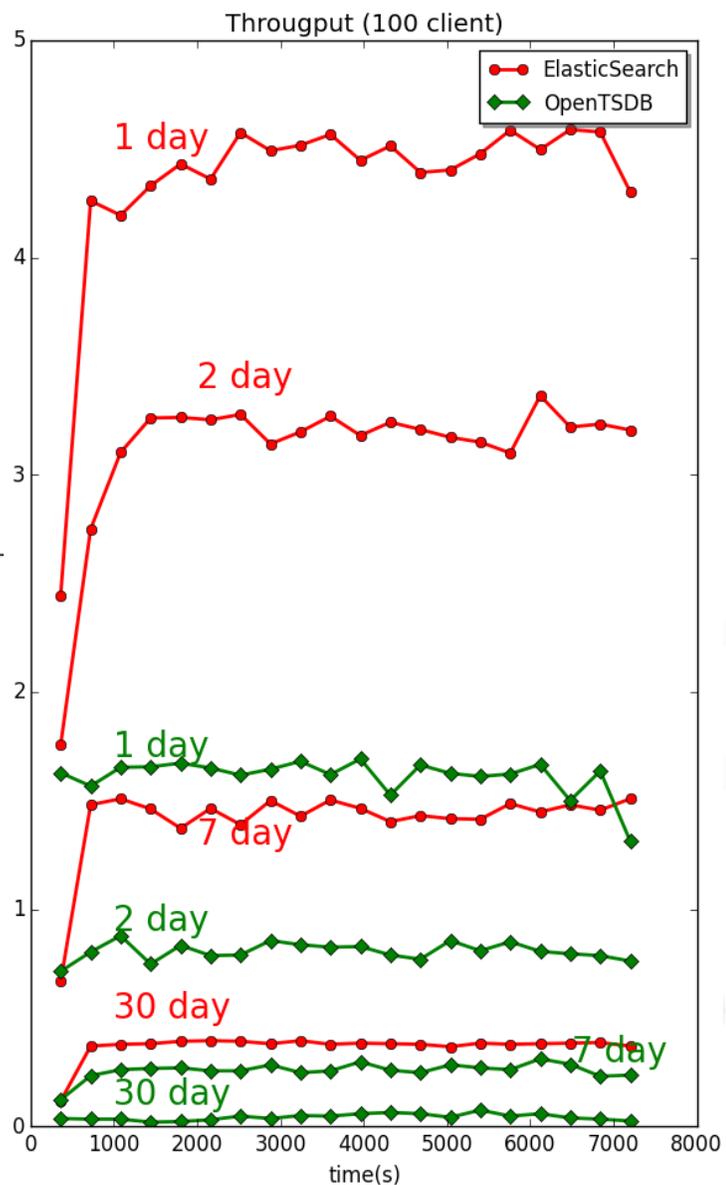
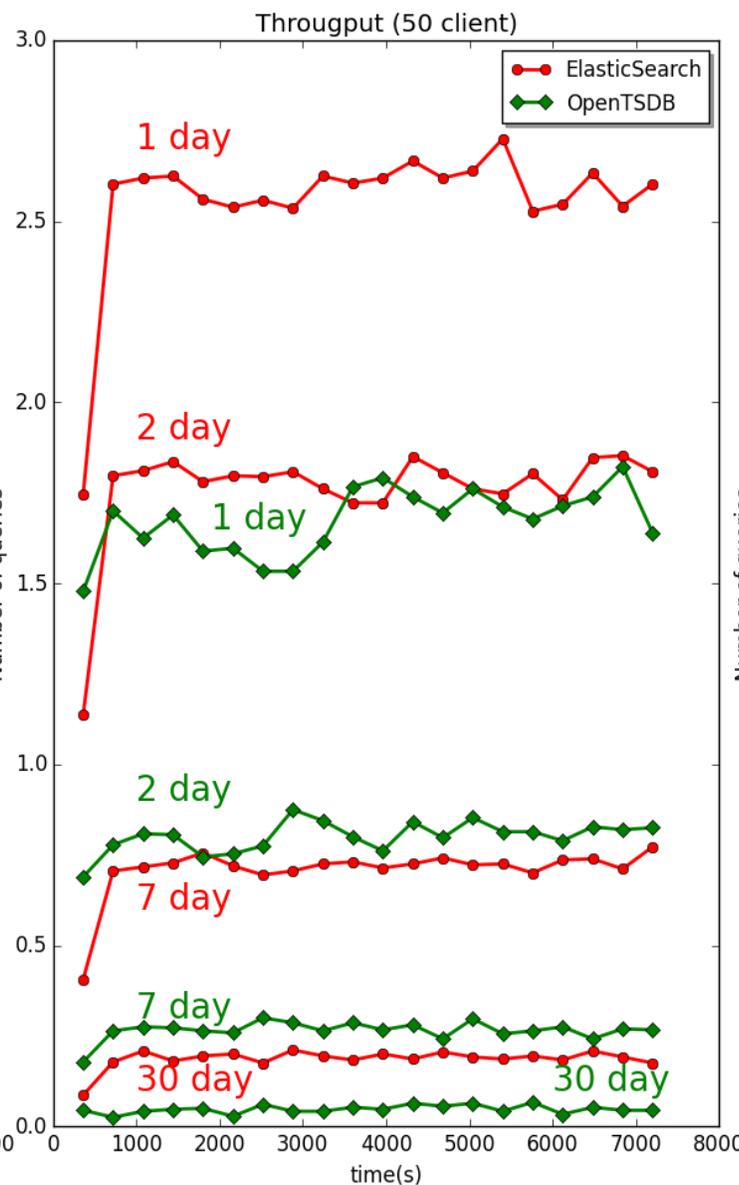
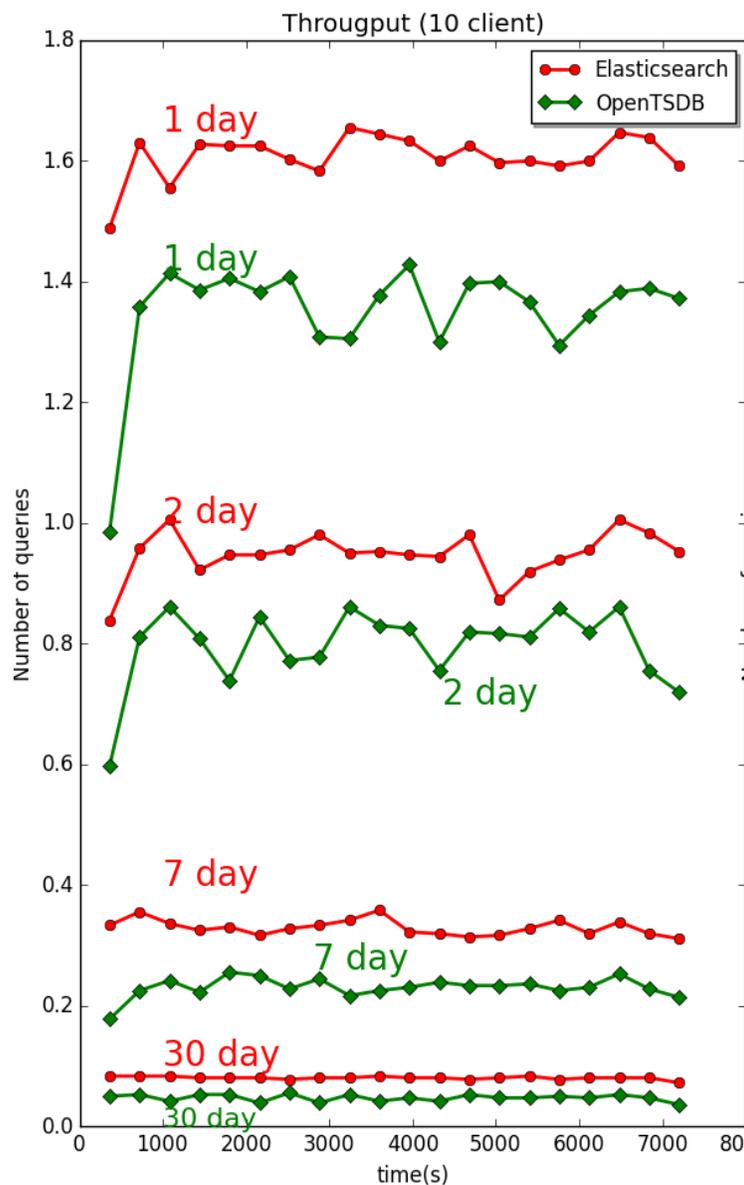




Response time of all experiments



Throughput of all experiments



- ElasticSearch was faster than OpenTSDB and InfluxDB
 - It is easy to maintain
 - Marvel is a very good tool for monitoring the cluster
 - ↳ license required...
 - It can be easily integrated to the DIRAC portal
 - OpenTSDB was slower than ElasticSearch but it may scale better by adding more nodes to the cluster
 - ↳ It is not easy to maintain (lot of parameters which have to be correctly set)
 - ↳ Very good monitoring of the cluster.
- InfluxDB is a new time series database, which is easy to use, but it does not scale
- Kibana can fulfil our needs
 - But we'll look at integration in the DIRAC portal
- According to our experience we decided to use ElasticSerach for real time monitoring of jobs, and for all real time DIRAC monitoring systems

Question, comments

