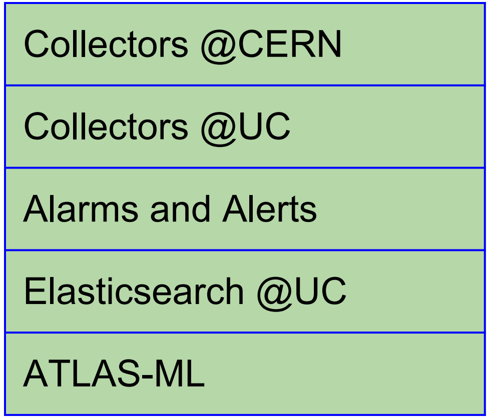
ADC Analytics news

Site Jamboree
2019-03-07

Topics covered

- Infrastructure
- Spark on demand
- GATES
- Future services (Operational Intelligence)

Analytics infrastructure



Collectors @CERN

Collectors @UC

Alarms and Alerts

Elasticsearch @UC

ATLAS-ML

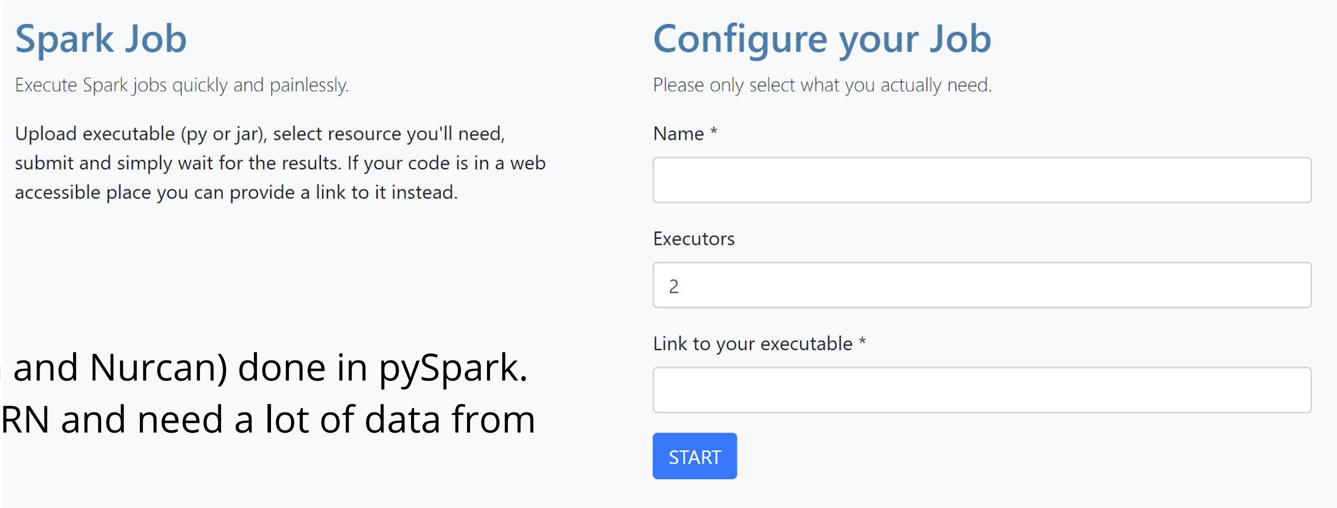
All systems running smoothly.

Moved away from sqoop and pig collectors and processing

- Much faster
- More reliable
- No hdfs copy of some of the data
  - If someone need it we can manually run sqoops to hdfs.

Kibana objects are being reorganized in different Spaces.

Analytics infrastructure - Spark on Demand



Spark Job

Execute Spark jobs quickly and painlessly.

Upload executable (py or jar), select resource you'll need, submit and simply wait for the results. If your code is in a web accessible place you can provide a link to it instead.

Configure your Job

Please only select what you actually need.

Name *

Executors

2

Link to your executable *

- Frontier studies (Millissa and Nurcan) done in pySpark.
- Rather slow to run at CERN and need a lot of data from Elasticsearch at UC.
- Developed a generally useful spark job submission for our ML platform.
- Based on spark 2.4.0 and using spark clustered k8s deployment.

This is a very new service.

Please use, test, report problems, ask for features.

ГАТЕС - General AB Testing Service

From Wikipedia, the free encyclopedia

A/B testing (**bucket tests** or **split-run testing**) is a randomized experiment with two variants, A and B It includes application of statistical hypothesis testing or "two-sample hypothesis testing" as used in the field of statistics. A/B testing is a way to compare two versions of a single variable, typically by testing a subject's response to variant A against variant B, and determining which of the two variants is more effective.

Experiments have large data stores collecting data from different computing systems: job scheduling, data distribution, FTS, PerfSONAR, etc.

While that is great for monitoring, accounting, and finding issues, it is not sufficient for the system optimization.

One can try to guess what kind of effect a change will made, but without validation it does not mean much.

First proposed by Friedrich H. and used for measuring impact of C3PO. Took 2-3 months to implement it.

We need a way to quickly test different options and get actionable answers.

ГАТЕС - General AB Testing Service

Should be very general, not only ATLAS, even not only HEP

Examples:

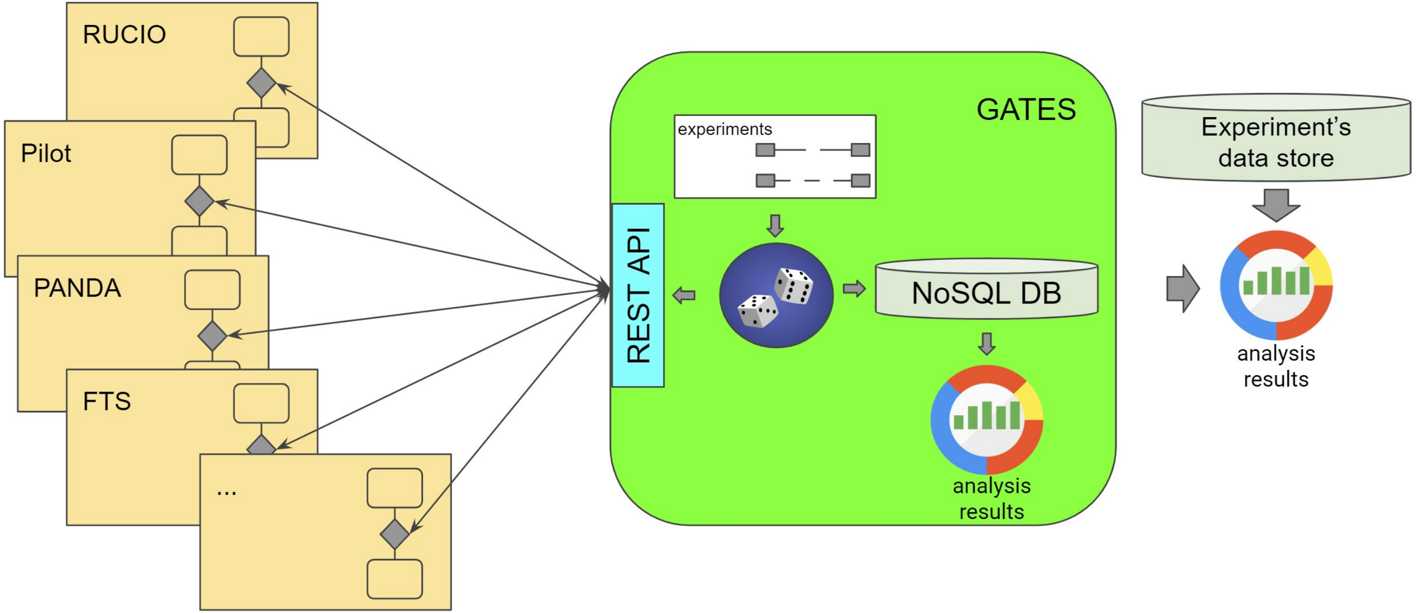
- Local vs Remote access
- Different job scheduling algorithms
- One data mover vs another
- Different FTS scheduling algos
- FTS limits per site/link
- Is a Rucio rule helping or not?
- Is there an impact from a different memory allocator (tcmalloc, jemalloc)?
- Is it better to oversubscribe servers: jobs/core=1,1.1 or 2?

ΓATEC - General AB Testing Service

Requirements for an A/B testing service:

- General (avoid experiment specific dependencies)
- Simple to instrument code (shell, python, c/c++)
- Fast and reliable (millions of experiments per day, with ms latency)
- Flexible (accommodate all different test options)
- Easy to correlate hypothesis with outcomes

ΓΑΤΕC - General AB Testing Service



Development already started but it will take some serious effort.
Volunteers needed!

Operational intelligence

Currently we have every system monitored separately, by different tools and different people (FTS - Joaquin, Panda - Ivan, Frontier - Nurcan,...)

To options:

1. We can continue in that way and just try to provide several more tools for most frequently encountered tasks (eg. anomaly detection in timeseries, issue classification, annotation tool, cost calculation tool.
2. Full rehaul: have models of every subsystem and create a monitoring / alerting system similar to Inductive Monitoring System. Can take into account interactions between subsystems. Gives global picture.