


Analytics

what can we do today for sites

Ilija Vukotic, Rob Gardner, Lincoln Bryant
University of Chicago

ATLAS Sites Jamboree, January 29, 2016



Analytics Platform

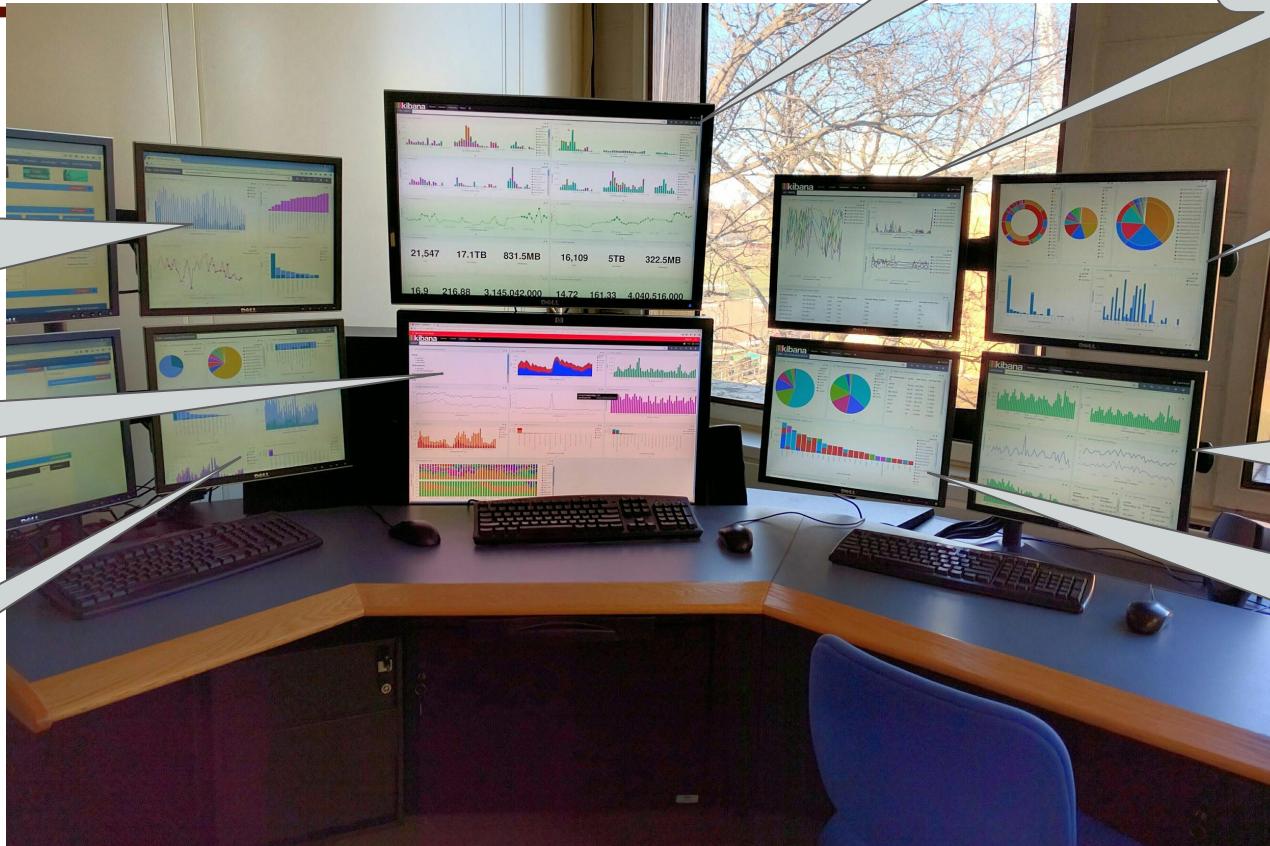
Analytics platform continuously collects a lot of disparate data and makes it possible to easily inspect, visualize, and analyze it. The platform itself is a much larger topic and this talk will be limited only to parts of it that are the most useful to ATLAS sites.

We won't cover: clusters, data transports, hdfs, pig, indexing ...

We will cover: **ES cluster, data sources indexed, Kibana, already created dashboards, how to customize and make your own**

so you can do something like this:

Analytics control center



Jobs
efficiencies,
users,
resource
utilization

FTS
transfers

Inter-site
network

Analysis
errors

Analytics
cluster
overview

PerfSonar

Overflow
jobs

DDM space
tokens,
biggest
users

For future reference

[Analytics TWiki page](#)

egroup: [atlas-adc-data-analytics](#)

Tutorials:

part 1: [Platform description, data indexed, importing, simple data analysis](#)

part 2: [Sqoop, Flume, Pig, ES indexing](#)

part 3: [Using Kibana](#)



ElasticSearch is a Lucene based search engine that indexes the data and provides very fast searches, filtering, aggregations. Programmatic access from CERN is possible at ***cl-analytics.mwt2.org:9200***. Please ask if you need it.

Kibana - web based visualization on top of ElasticSearch. Accessible to everyone at ***cl-analytics.mwt2.org:5601***. For best performance use Chrome.

The cluster is at Clemson University, provided courtesy of the [CloudLab](#) project.

Data sources

The most important for sites:

- **PandaJobs**
- **Rucio DDM data**
- **PerfSONAR**
- **FTS**
- **FAX**
- **Rucio popularity**

Other data:

- xAOD traces
- Rucio service data

The following slides will give links to all the dashboards shown.

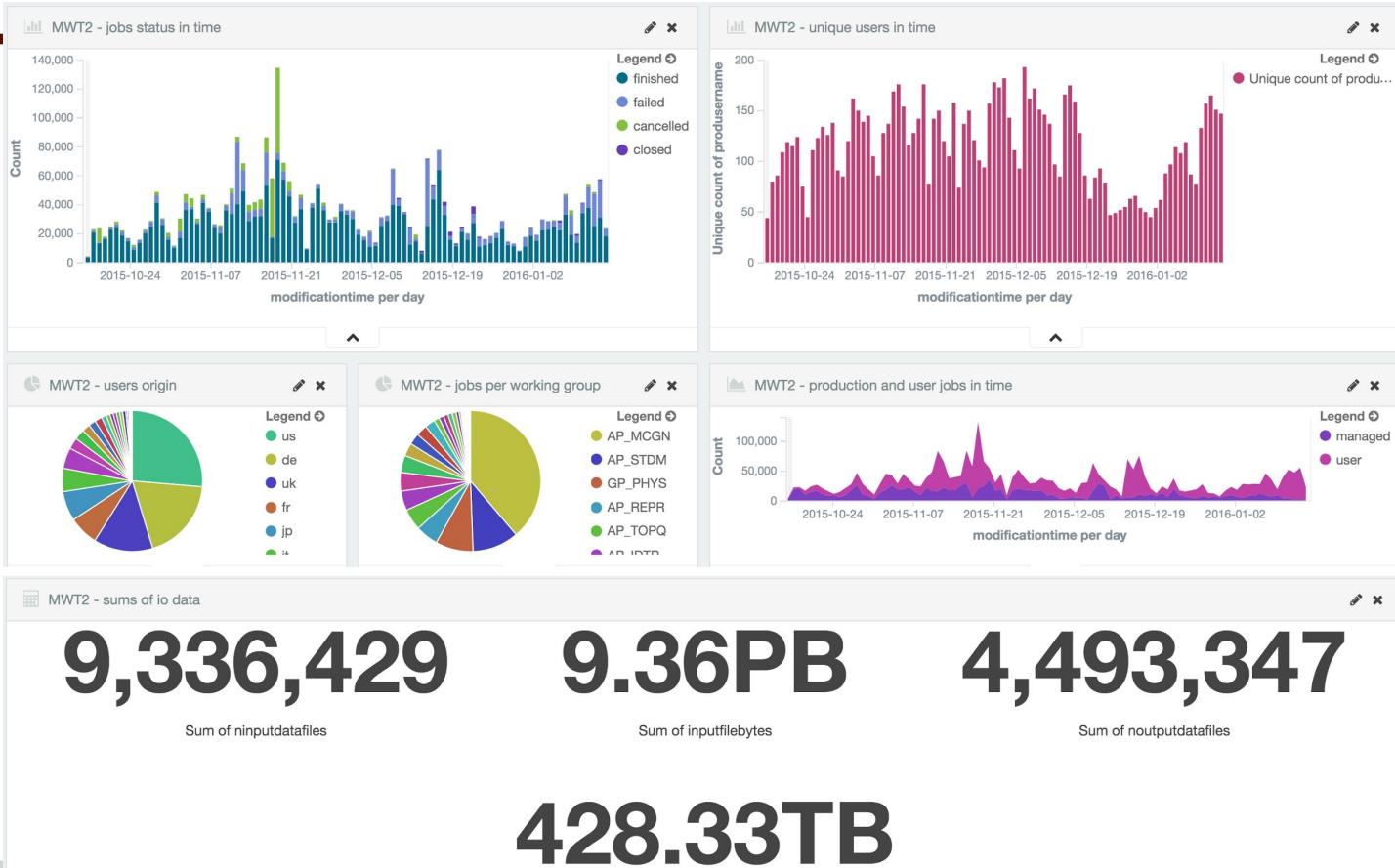
All the dashboards shown are as general as possible, so you can simply filter on them for the specific information you are interested in.

Create site/cloud overview dashboard

Example:

MWT2

US



PandaJobs

Practically all the info on all the grid jobs: 106 variables

Describing: site, host, user, task, priorities, input and output data, timings, status, errors, parameters, memory, etc.

Period: from July 2014 up to last 30 min.

Template: `jobs_archive_*`

Dashboards:

- Errors (codes, transform, pilot), cost of errors, reasons (memory, times)
- Analysis jobs, biggest users, queue wait times
- Data used (counts, times, etc.)
- Job efficiencies (counts, per input type, transfer type), inefficient users

PandaJobs

errors

Filter only your site by typing e.g.
computingsite:ANALY_MWT2* in search
field.

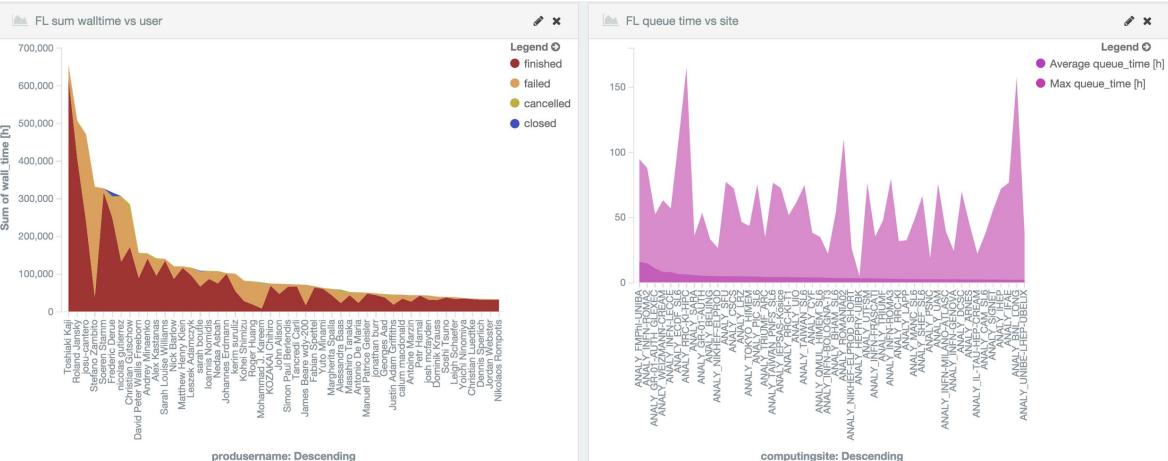
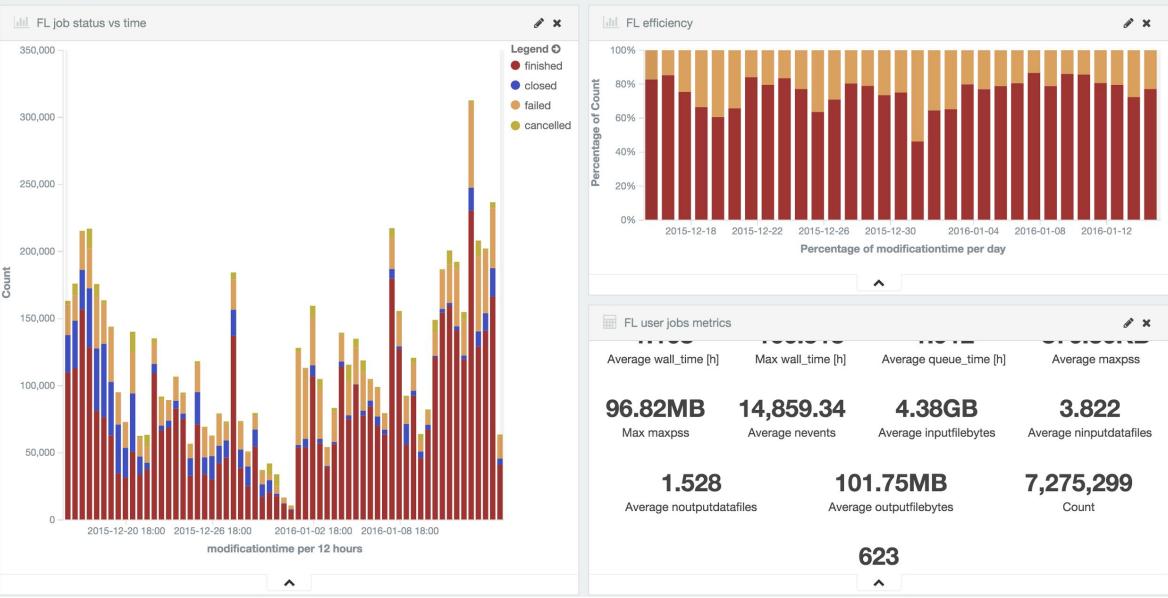
Or simply drill down by clicking on the
bar/slice/line on the plot.

You can save filtered dashboard. Always
name site specific dashboards starting
with your site name.



PandaJobs

Analysis



PandaJobs

Input data



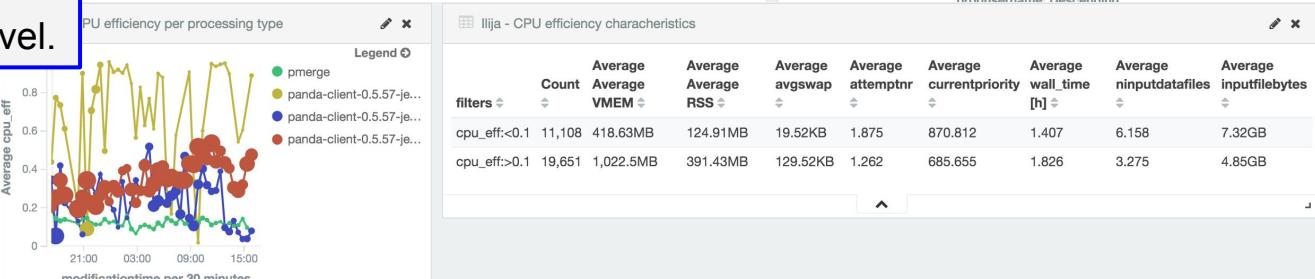
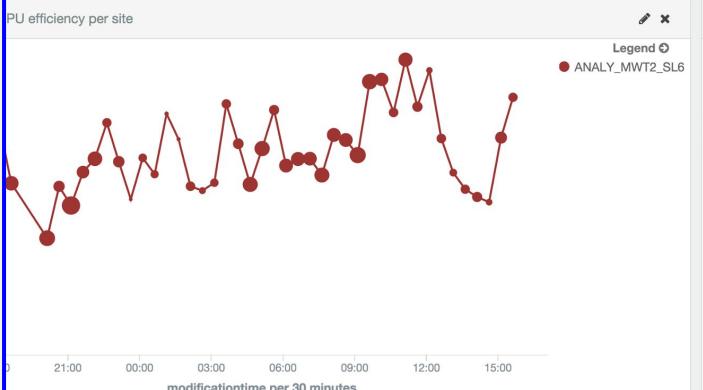
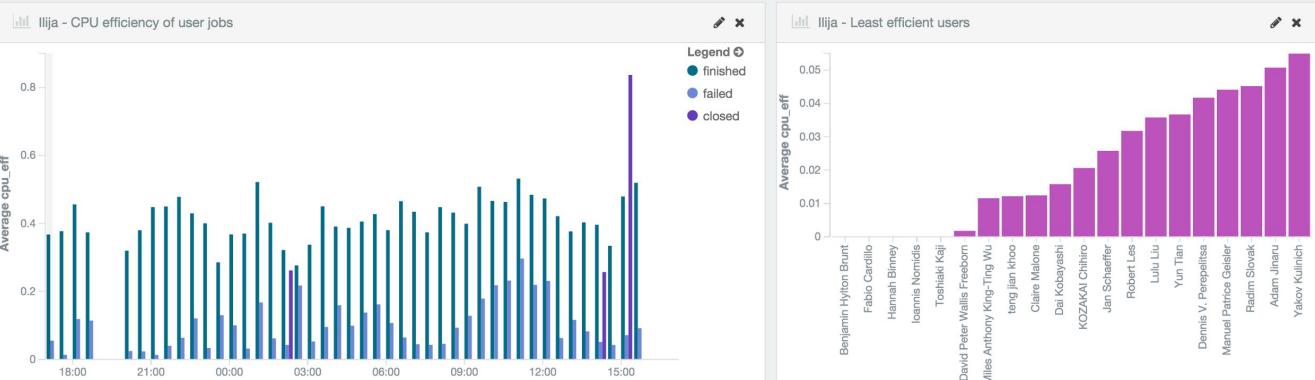
PandaJobs efficiencies

Use to monitor jobs for batches of inefficient jobs.

Investigate reasons for low CPU eff.

Compare efficiency of the same task on other queues.

After drill-down switch to “Discover” mode to go down to a single job level.



DDM

Uses RUCIO daily dumps to index all the datasets at all the RSEs.

For each dataset gives: name, size, tags, data type, time (creation, last access, since last access), RSE

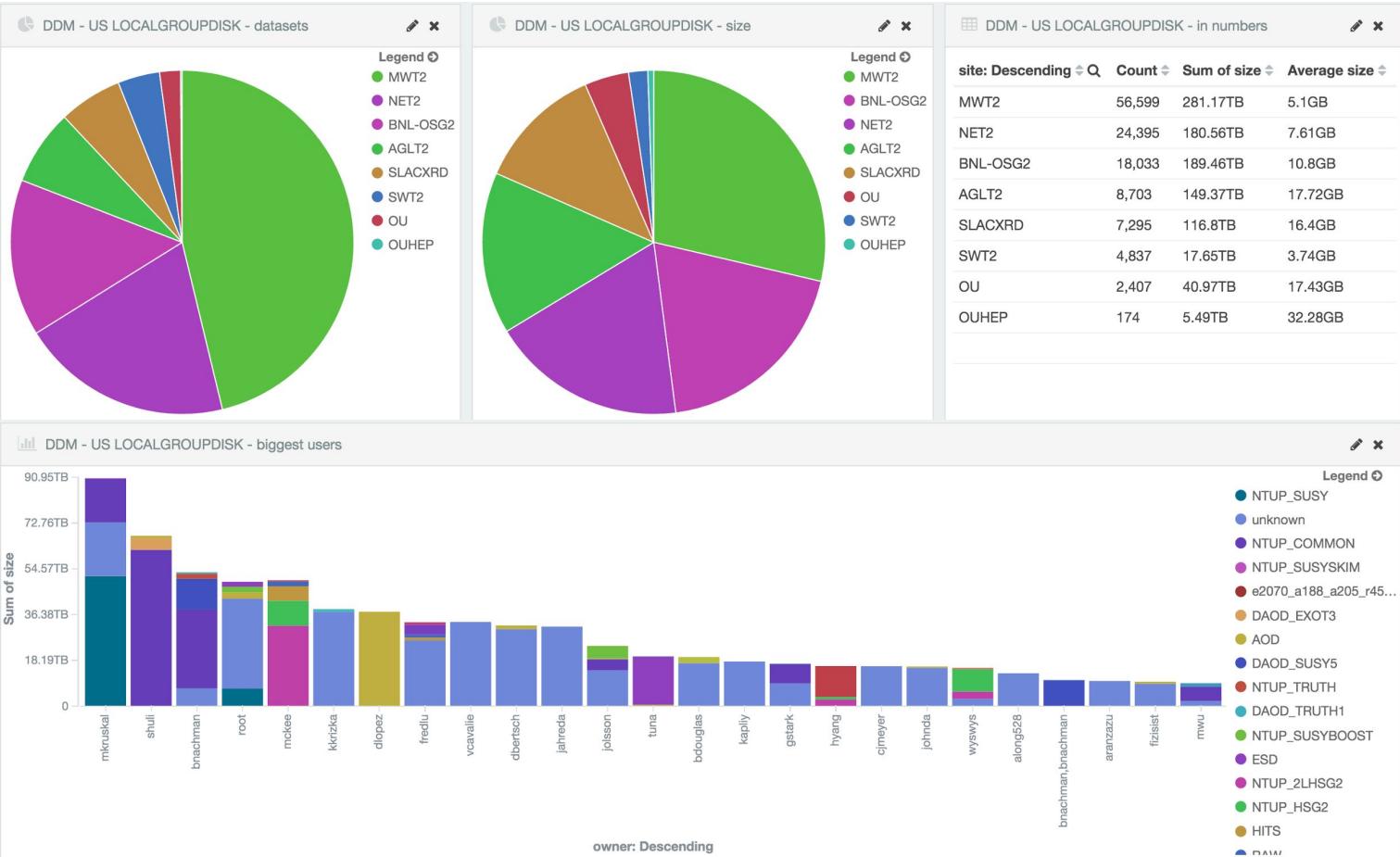
Template: ddm-* . Kept for 1 month.

The same data summed up per scope, owner, RSE available in template:
ddm_aggregated*. Kept indefinitely.

Dashboards:

- global overview of a token (dataset counts, sizes, datatypes, unused data)
- global data distribution

DDM US LOCALGROUPDISK



owner: Descending

PerfSONAR

Gets data from ~250 sites. Three types of data: Throughput, One way delay, and Packet loss rates

Contains: vo, site, hostname, production status of both source and destination, mean, median, sd of delay, etc.

Template: network_weather_2-* Kept indefinitely.

Dashboards:

- global overview (measurement counts, all three measurements vs. time, best sources/destinations)
- all the measurements for one link
- all links of a distributed site

Perfsonar

all the measurements

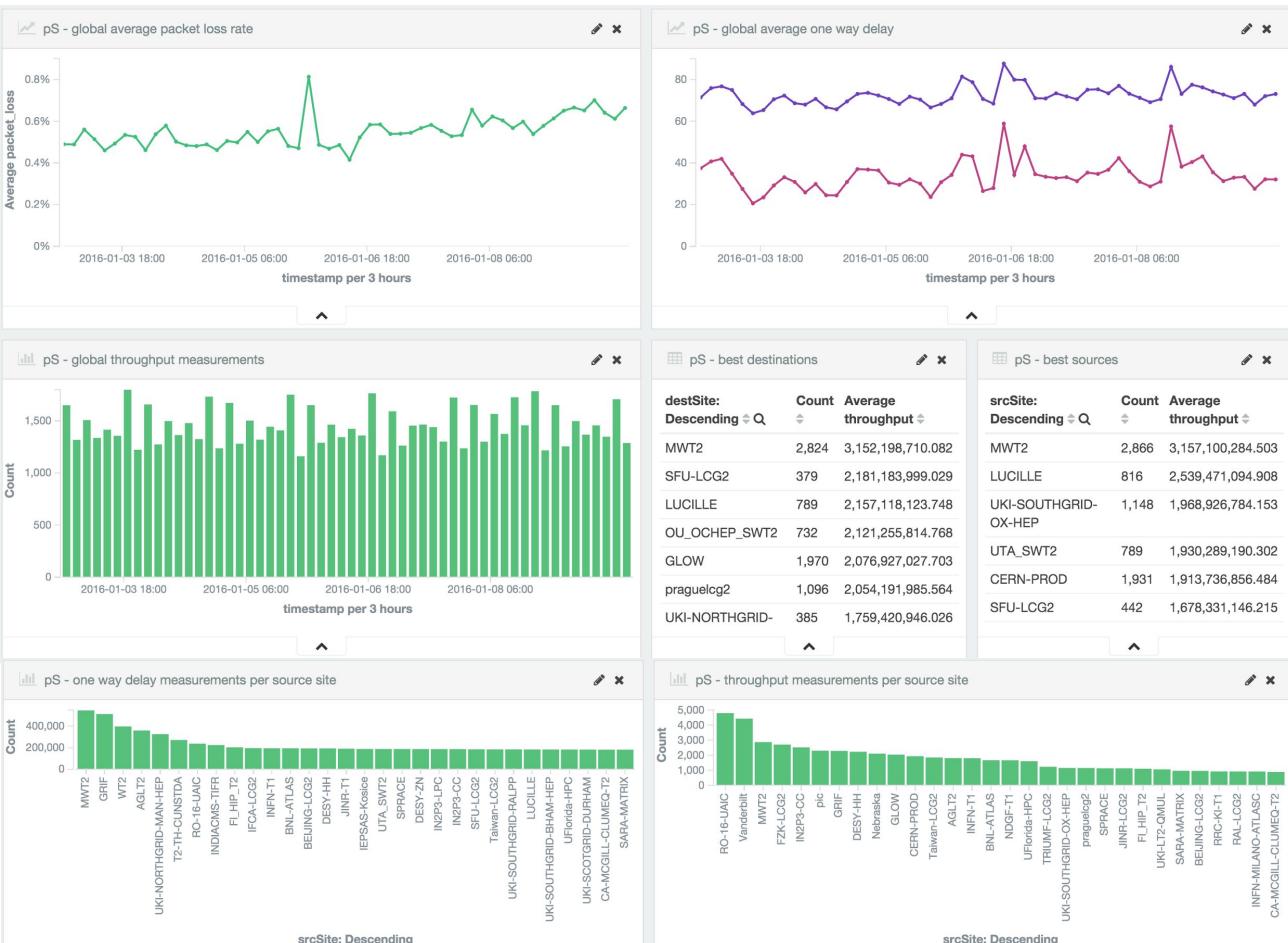
select your site as:

srcSite:MWT2

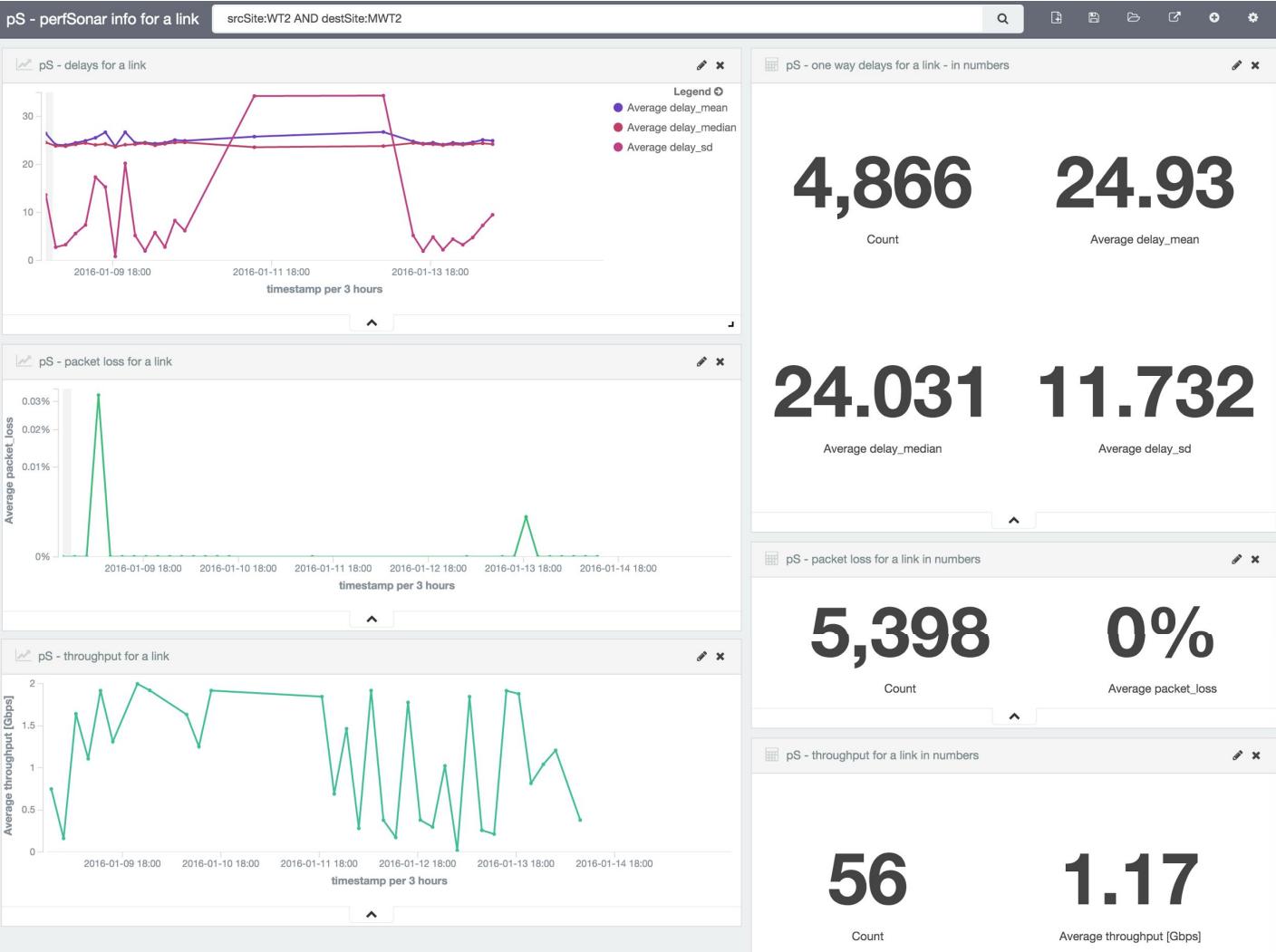
or ***destSite:MWT2***

to ATLAS sites:

destVO:ATLAS



Perfsonar single link



Perfsonar

Inter-site links



src: Descending	dest: Descending	Count	Average delay_mean	Average delay_median	Average delay_sd	Average packet_loss
72.36.96.4	192.170.227.160	7,032	2.528	1.87	6.935	0%
72.36.96.4	149.165.225.223	7,008	4.607	3.933	6.912	0%
149.165.225.223	72.36.96.4	6,989	4.256	4.057	3.064	0%
149.165.225.223	192.170.227.160	6,959	3.714	2.836	7.498	0.01%
192.170.227.160	149.165.225.223	6,947	3.34	2.639	6.483	0.01%
192.170.227.160	72.36.96.4	6,938	1.945	1.74	3.092	0%

Compare your site to others



Top 20 - analysis queues

Top 20 - production queues

Top 20 - FTS sources, destinations

FTS transfers

All the FTS transfers in real time.

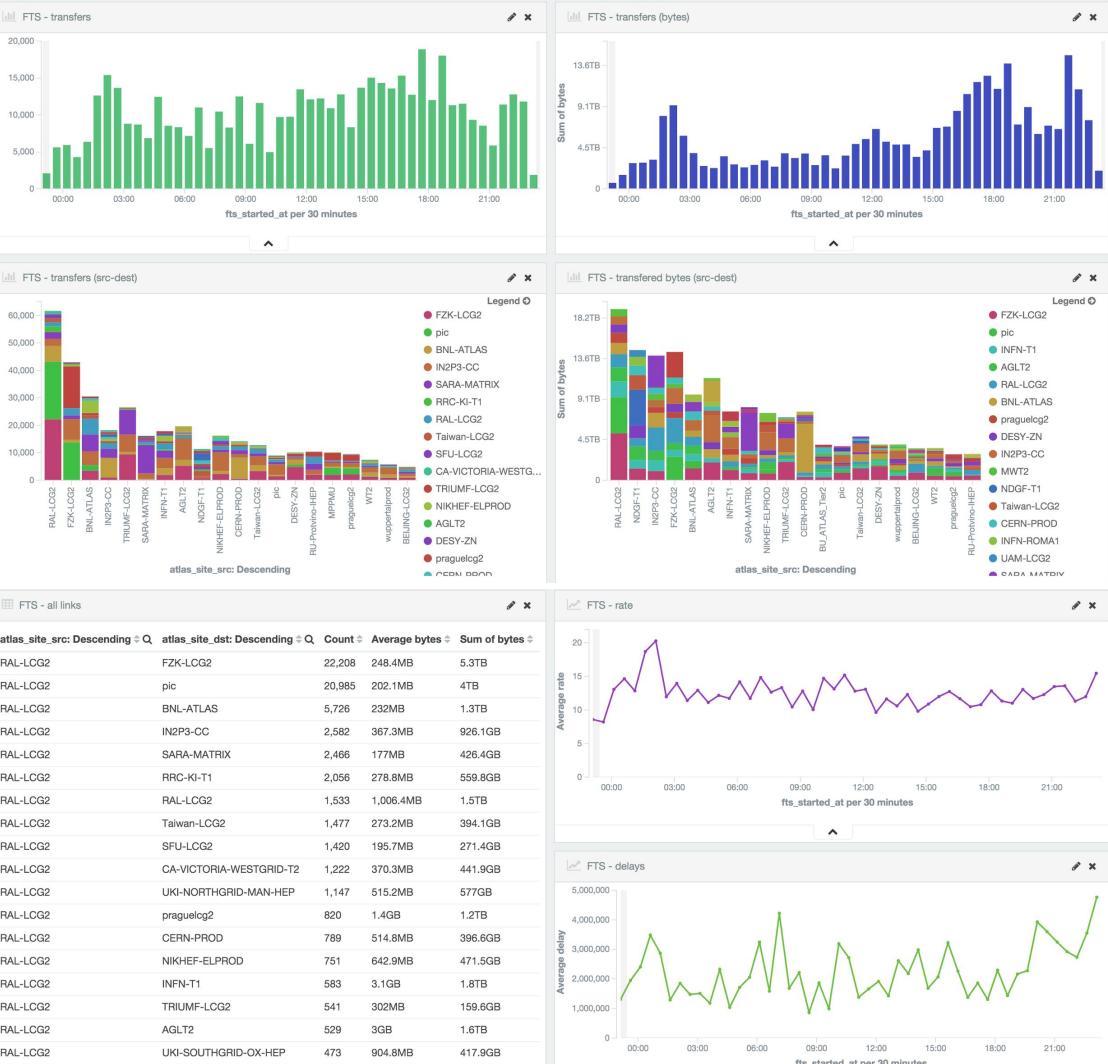
Contains: source and destination site and RSE, activity, bytes, times
(created, started, ended)

Template: [fts-]YYYY.MM.DD

Dashboards:

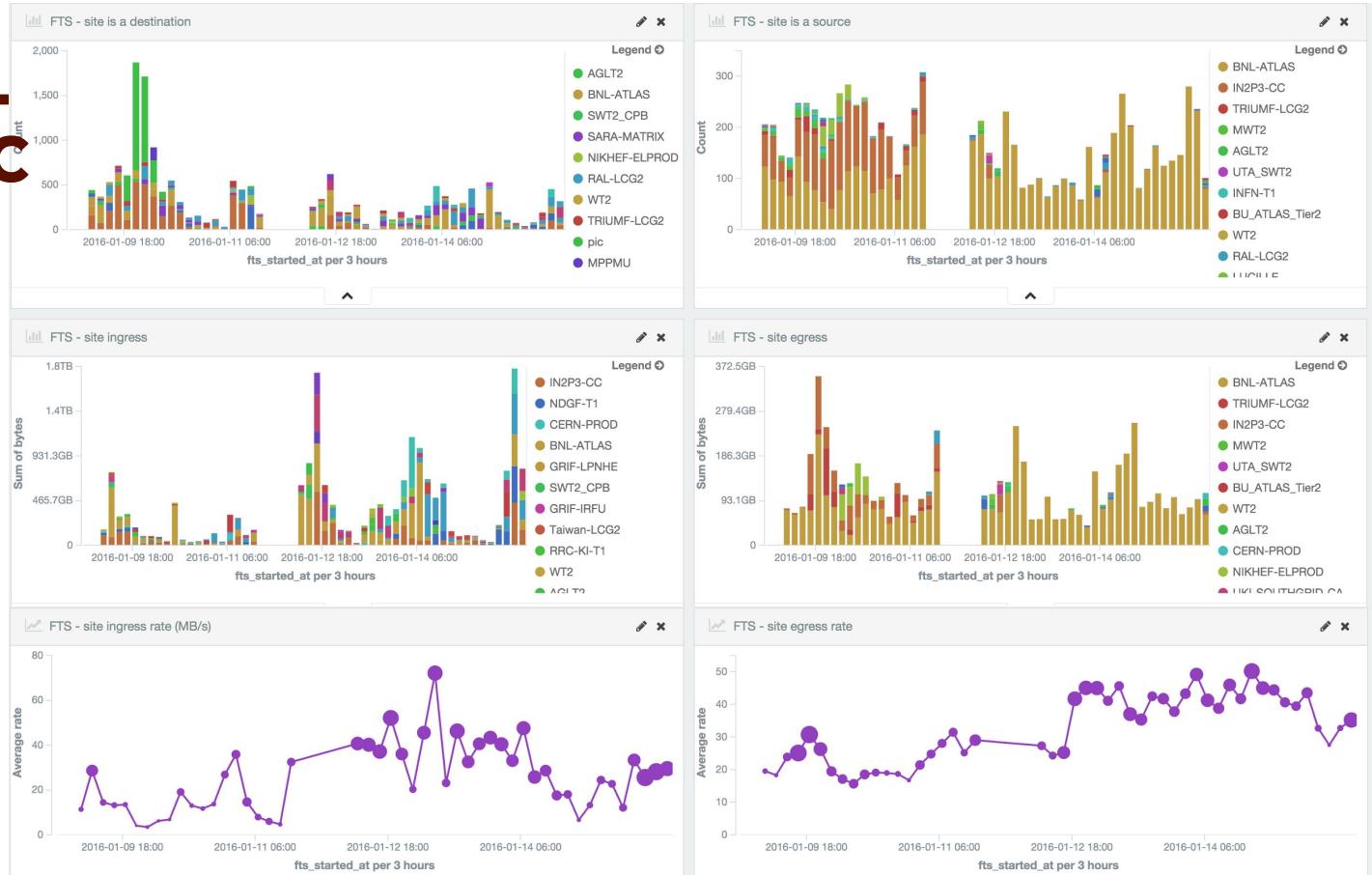
- Global numbers (counts, sizes, delays,etc.)
- Site specific
- Rucio delays analysis

FTS global



FTS

site specific



FAX cost

collects xrdcp rates between largest analysis queues and all FAX storages.

Contains: source, destination, rate

Template: faxcost-*

Dashboards:

- Global overview (counts, ingress, egress rates)

FAX cost



Ale's requested visualizations

jobs by status vs time

number of jobs by status

wallclock consumption of panda failed jobs

jobs done at an ATLAS site

production job durations during last week for one queue

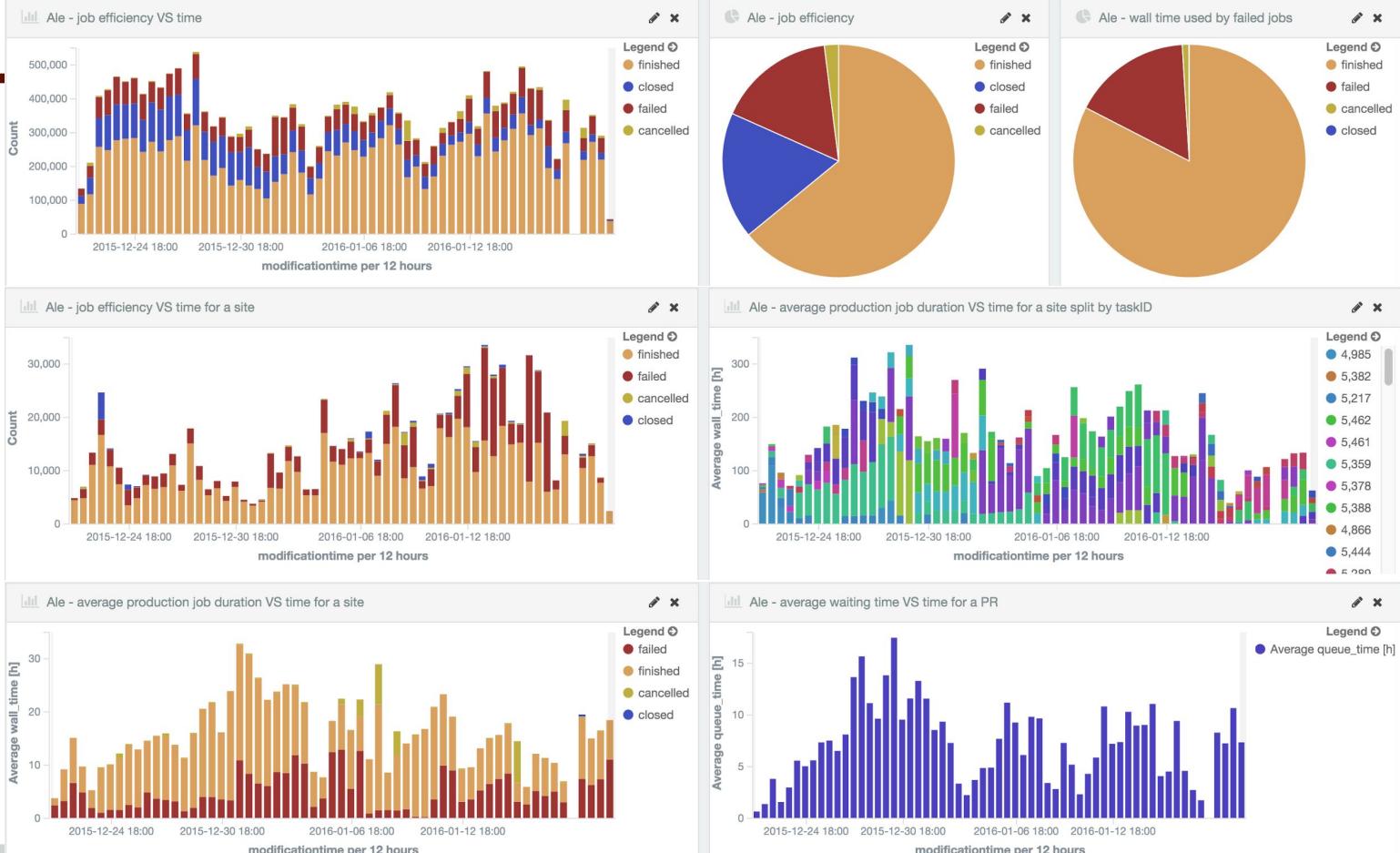
split by reqid

MAXPSS of jobs running at a site over the past month

waiting time of jobs running at PR over the past month

**Ale's
visualizations
all in one
dashboard.**

Ale's dashboard



Reporting from ES

One can use ES to periodically check for irregular situations (high packet loss, low DDM token space, low efficiency jobs,...) and generated reports (send e-mails, sms,...). There is an ES plugin that makes it trivial to do so, but it is not free.

Still it is easy to make a cron job to do so. Cron job should run a query against ES and send mail based on returned result. Simplest is to use curl, for a more complicated things (checking different things and generating one mail with full report) it can be easier to use python. Simple examples can be found [HERE](#).

Two things to keep in mind:

- don't do it every 5 seconds, this is a shared resource
- do all the filtering and aggregations on ES, you don't want to transport back a lot of data.

Advanced topic

We deployed Timelion, kibana plugin that allows for easy time series analysis, derivations, moving averages, arithmetic operations,...

Not yet documented, so only for the brave ones.