Providing Network Awareness to ATLAS And Beyond

Ilija Vukotic
University of Chicago

HEPiX Fall 2016, LBNL

Goals

Primary functions:

- Aggregate and index network related data of interest not only to ATLAS but also WLCG, OSG communities
- Provide a generalized network analytics platform
- Network anomaly detection, alarm and alert system
- Serve derived network analytics (eg. to ATLAS production, DDM & analysis clients)

Big Data technologies have proven to be very useful for storage, visualization and analysis of the ATLAS distributed computing (meta)data. The ATLAS analytics platform:

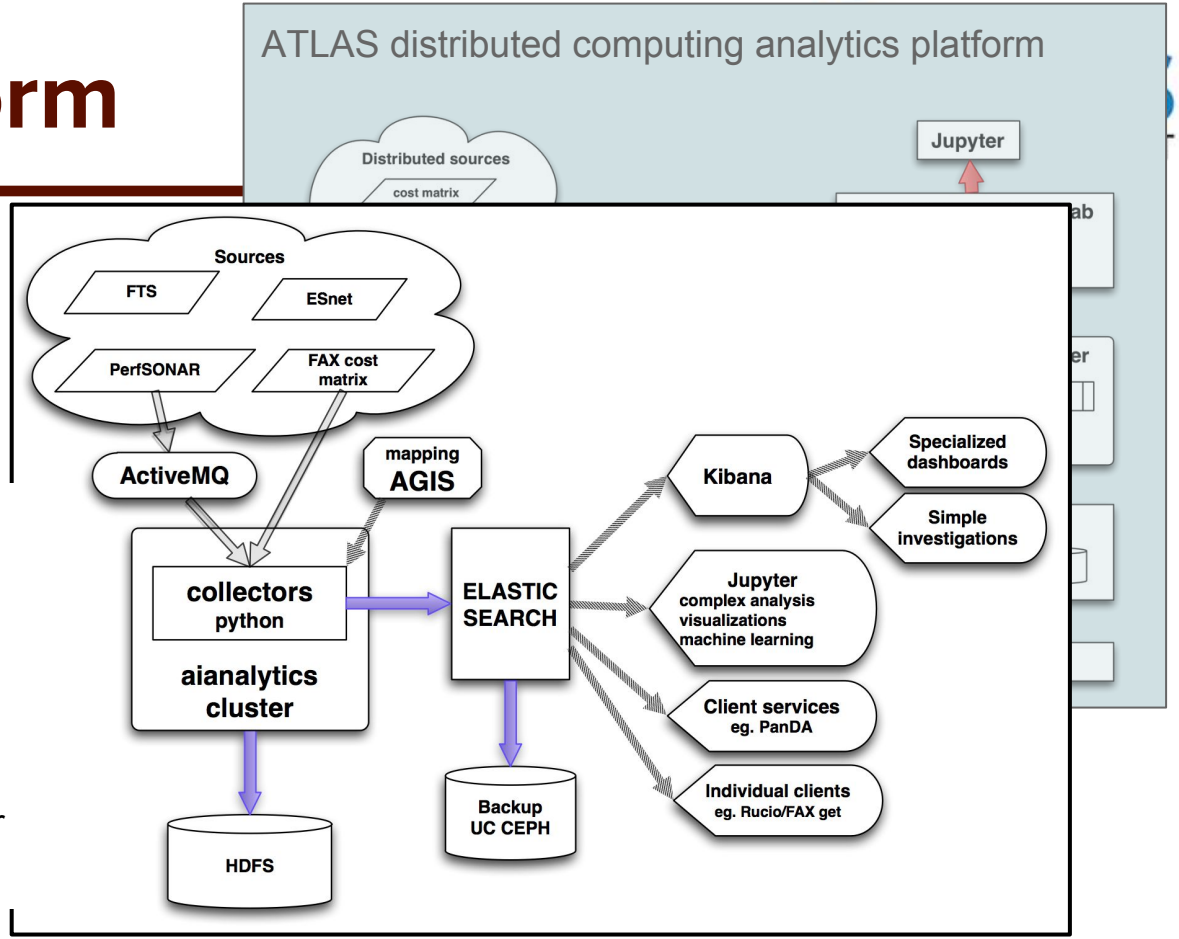
- Scales well to data rates, data size we need
- Monitoring comes for free
- Easy to do quick investigations without writing any code, directly in a web browser
- Serve as a backend to different clients

Analytics platform

ATLAS analytics platform combines many more data sources we only need a part of the system.

Our workflow:

- Each source has dedicated python collector
- Data indexed in Elasticsearch
- Fast visualization in Kibana
- Open data access through Elasticsearch REST API
- Data analysis on a co-located Jupyter cluster



Elasticsearch cluster

Elasticsearch is a distributed, real-time data and analytics, high availability open source search engine built on top of Apache Lucene. Works with structured JSON documents, schema free, by default all fields are indexed, data are compressed.

Indices:	Memory: 246GB /	Total Shards:	Unassigned Shards:	Documents:	Data:	Uptime: 4
2088	433GB	18644	0	8,481,799,362	9TB	months

THE UNIVERSITY OF CHICAGO

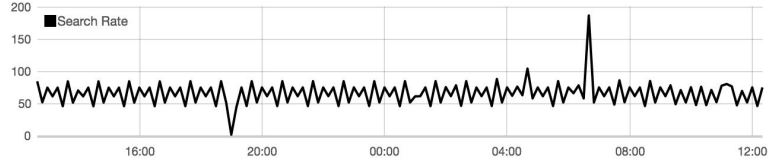
5 data + 3 head nodes

E5-2620v4, 64GB RAM, 4x800 GB SSD, 10Gbps NIC

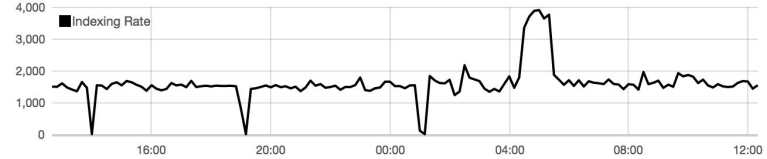
Used for ATLAS Distributed Computing analytics

Quite high base load

Search Rate: 74.46 /s



Indexing Rate: 1,536.95 /s



Jupyter

At University of Chicago

- Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz x2
- Tesla K20c x2
- 128GB RAM , 4.5TB RAID-5 for /scratch, 1TB RAID-1 for /
- Simple universal user, pass authentication, notebooks automatically checked in a git repo.

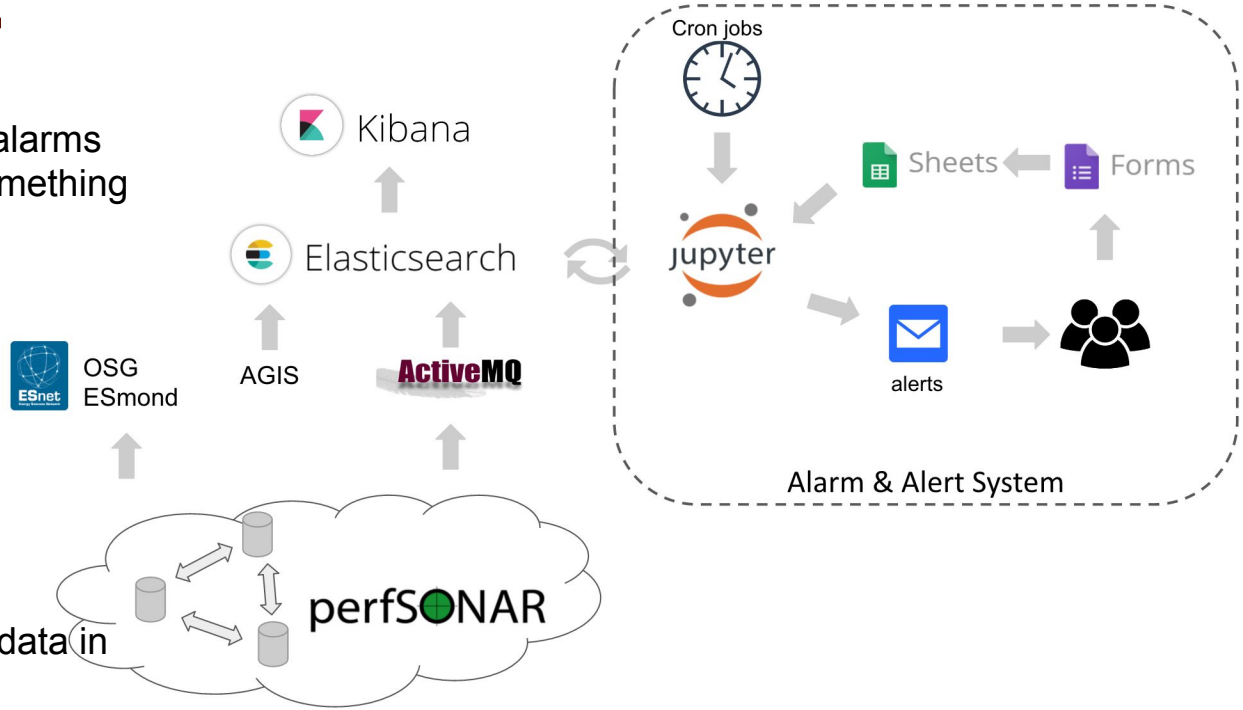
Advantages:

- Local to Elasticsearch
- Packages made specifically for the network data investigations
- Most of ML libraries installed, if there is anything missing we can easily add it.
- Learn from codes of others. See what was already done.
- Easily integrate new tests with the Alarm & Alert service

Alarm & Alert Service

Motto: less is more.

- Be conservative when raising alarms
- Alert only those that can do something about it.
- As little support as possible

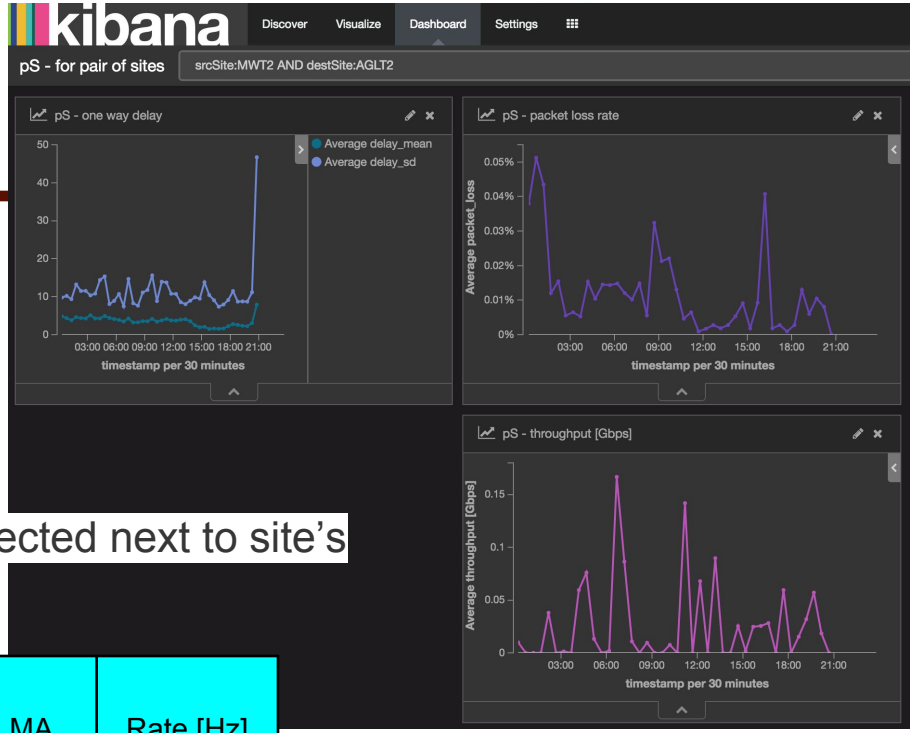


Anything that can be derived from the data in ES can be used to create alarms.
Cron jobs run notebooks, stores data in both ES and google spreadsheets.
Subscriptions made through a google form.

perfSONAR

A widely-deployed test and measurement infrastructure that is used by science networks and facilities around the world to monitor and ensure network performance.

perfSONAR servers are dedicated machines connected next to site's storage.



Measurement type	Sources		Destinations		Links	MA	Rate [Hz]
	sites	instances	sites	instances			
Packet loss rate	73	196	79	205	9041	110	20.1
One way delay	73	196	78	205	8971	110	19.9
Throughput	75	236	74	236	8933	128	0.24
Traceroute	83	233	93	361	7456	146	4.38

We index 2.4GB data/day.

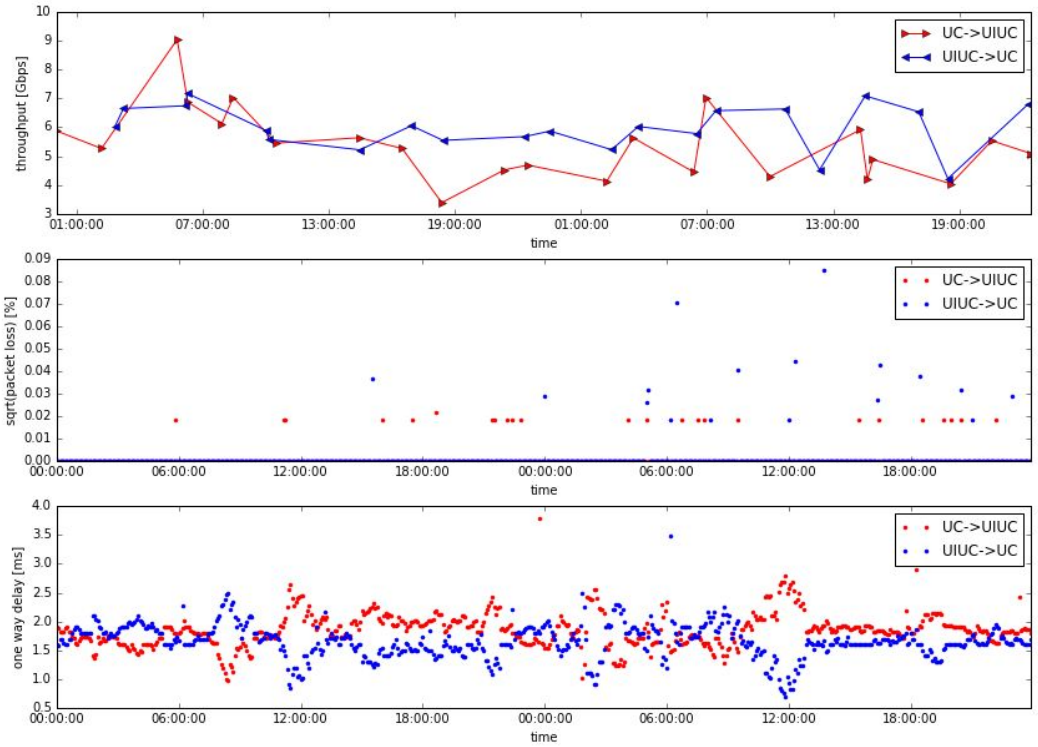
perfSONAR cont'd

Throughputs - single stream, 25 or 30 seconds memory to memory transfers. Rate limited to what NICs at source and destination server can do (ie.can't test 100Gbps link).

One way delays - mean, median, sd, in 5 min intervals. Precise to ~1ms. Sensitive to clock synchronization (uses 4+ NTP servers for this).

Packet loss rates - in percents, averages over 5 min intervals

Traceroutes - hops, time to live, round trip times.



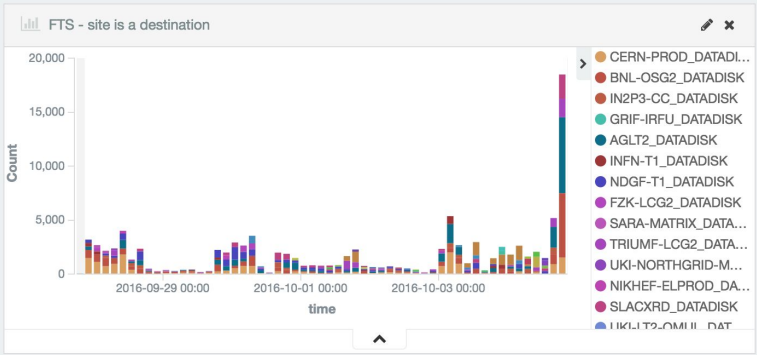
FTS

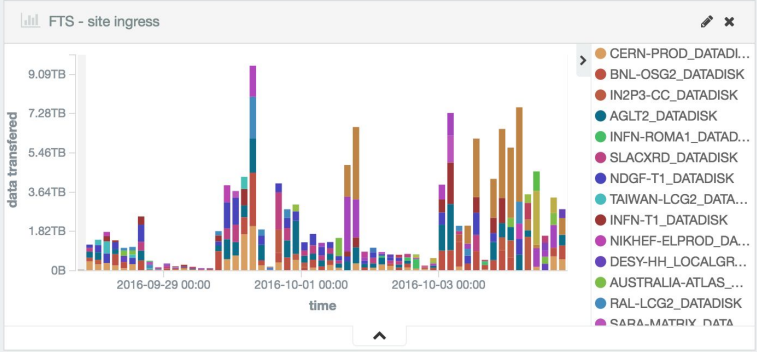
File Transfer Service - the lowest-level data movement service doing point-to-point file transfers on behalf of Rucio.

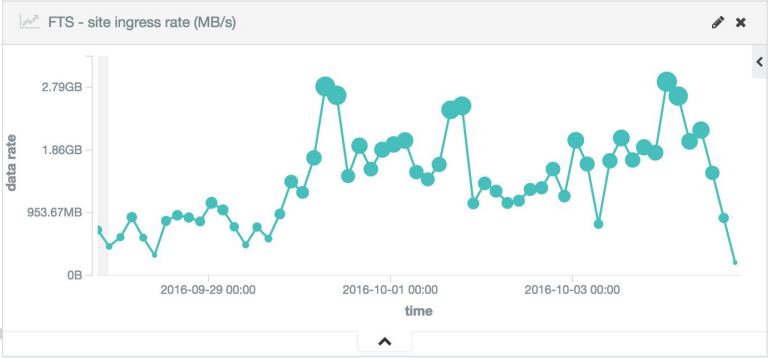
Each file transferred reports:

- Request time
- Start time
- End time
- Filename
- Filesize
- Source
- Destination
- Reason behind the transfer
- Transfer status

This is the largest source of traffic.

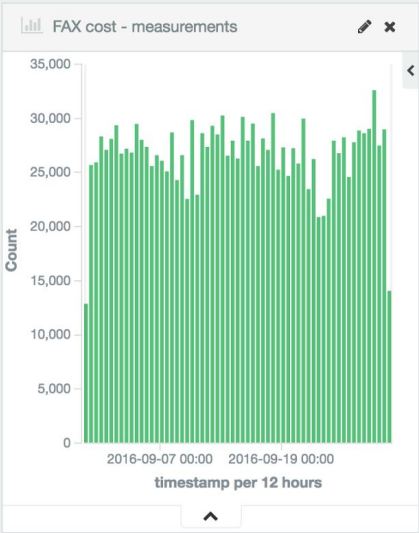


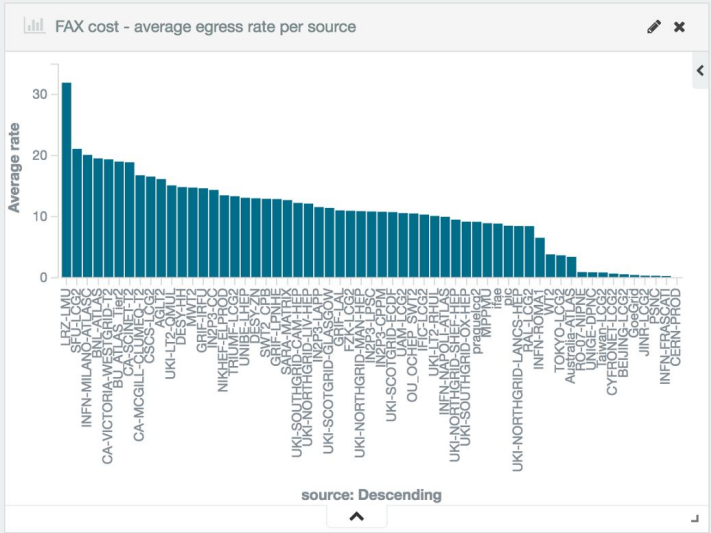


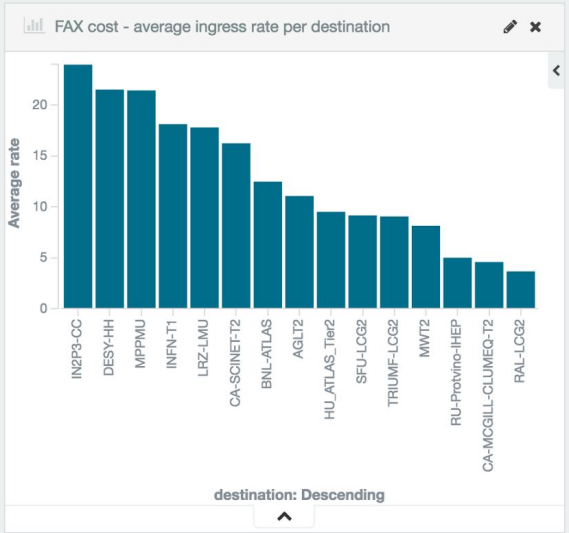


FAX

Measures rate of transfer using xrdcp from all the FAX enabled storages to 20 largest analysis queues WNs. Gives a measure of a rate an actual grid job can expect. Results are used in JEDI job brokering.





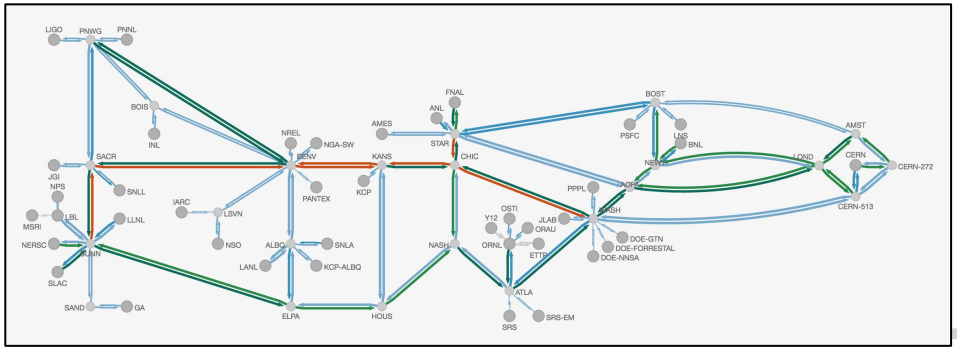


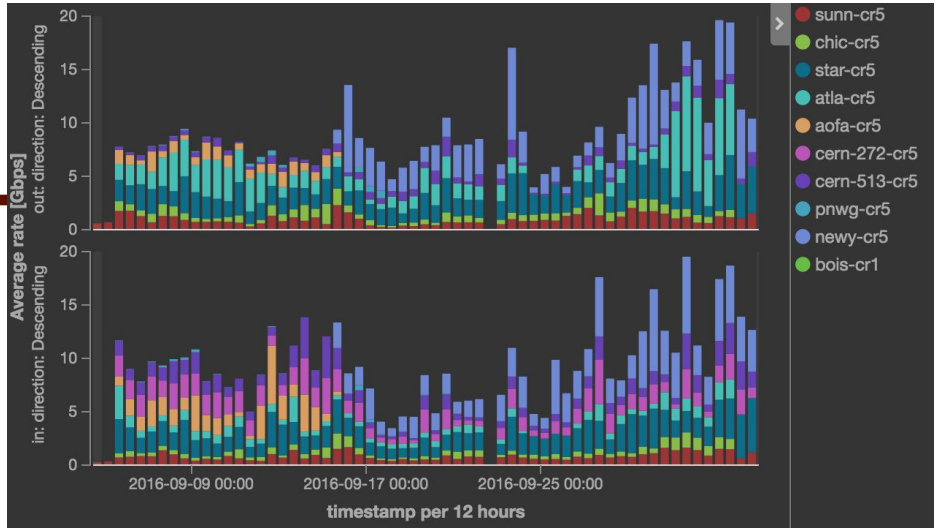
ESnet

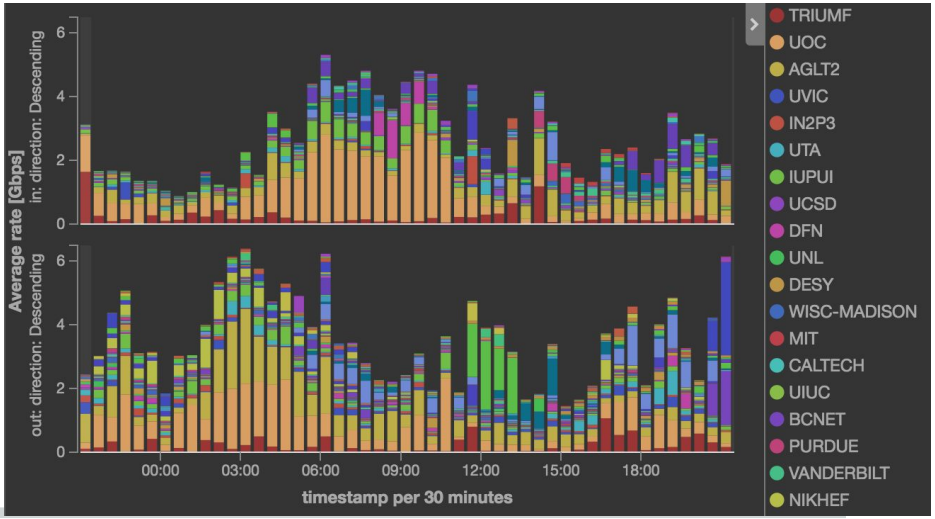
Gives rates observed through their network backbone devices (interfaces) and data flows split per site, protocol, etc.

All data at 1 min intervals.

Soon to come LHCONE data.







Analytics - task list

Measurements validation, masking, fixes

Crude alarms/alerts - single link, large time averages, simple cuts

Annotations of (ir)regular periods for supervised ML methods

Understanding of general correlations

Testing different models of anomaly detection

Testing performance/ free capacity prediction models

Understanding of corner cases

Quick response, fine tuned autonomous alarms/alerts

Integration in feedback loops of client services

Identification of problematic devices.

Integration of all the sources

Cleaning the data

Even these rather simple data require quite a bit of cleaning, validation.

Traceroutes

- Traffic shaping and redundancy providing routings need to be identified
- Higher the link utilization more ICMP packets dropped, adding complexity to path changes detection.

One-way-delays

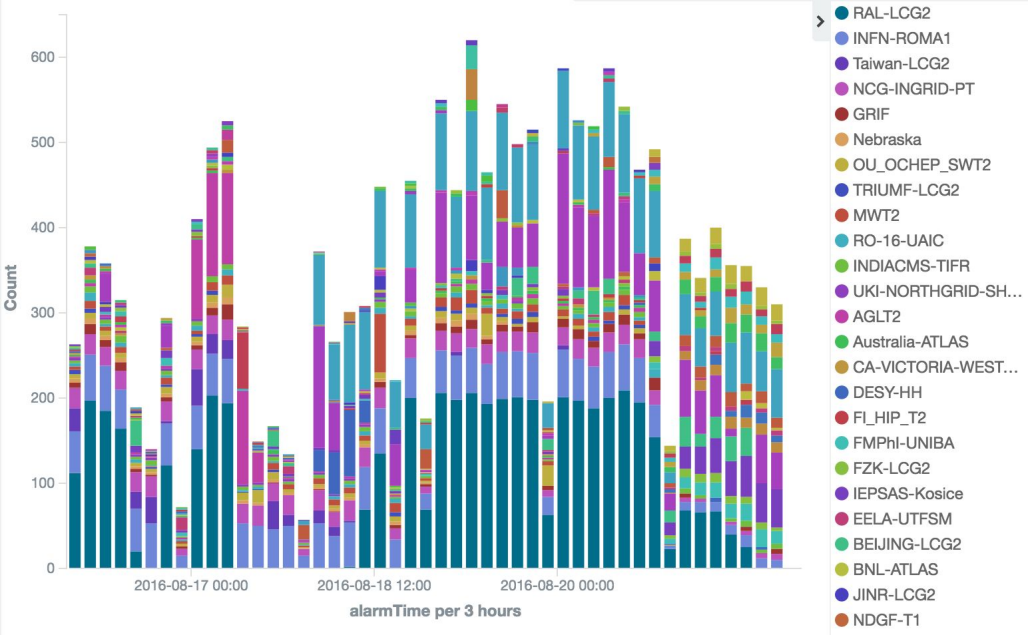
- Find drifts due to bad configurations or NTP server.
- Define minimal OWD's for "steady state" paths.

All data

- Check mappings to sites, GeoIP info, responsible people…

First crude alarm

Alarm generated when more than 15 links to/from a site show more than 2% packet loss over more than 30 min.

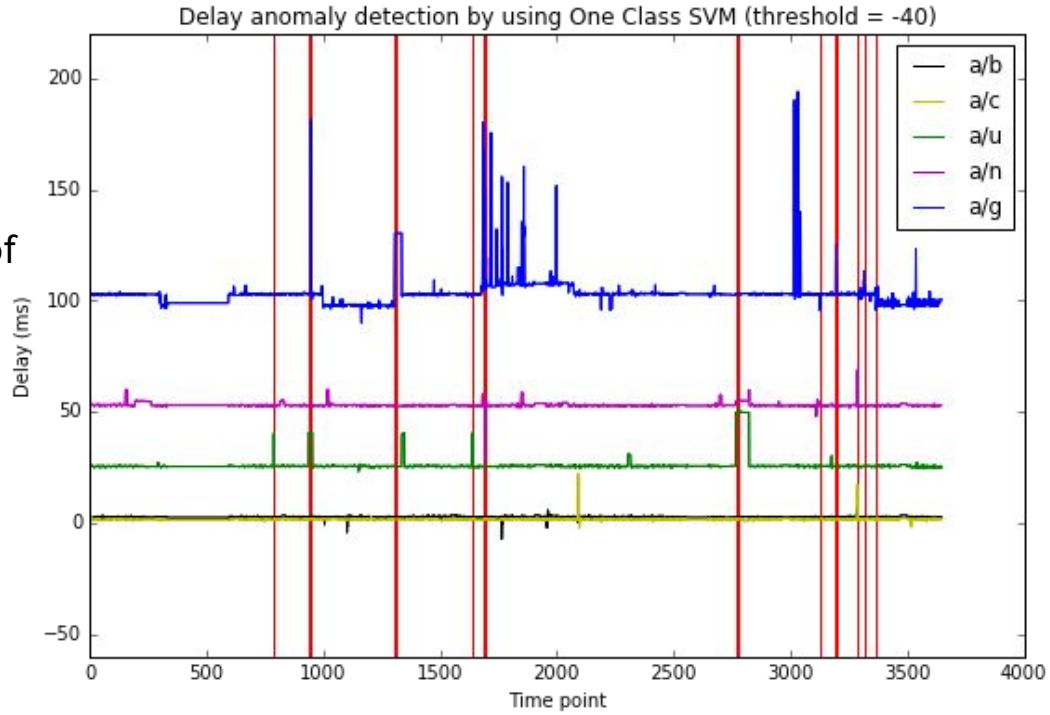


Seems we have quite a bit of “low hanging fruit” to pick…

SVM for anomaly detection

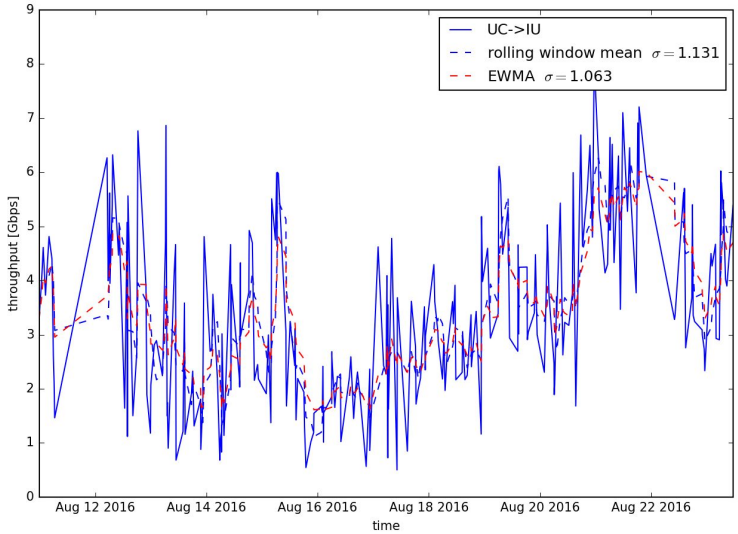
First try to detect routing changes from delays:

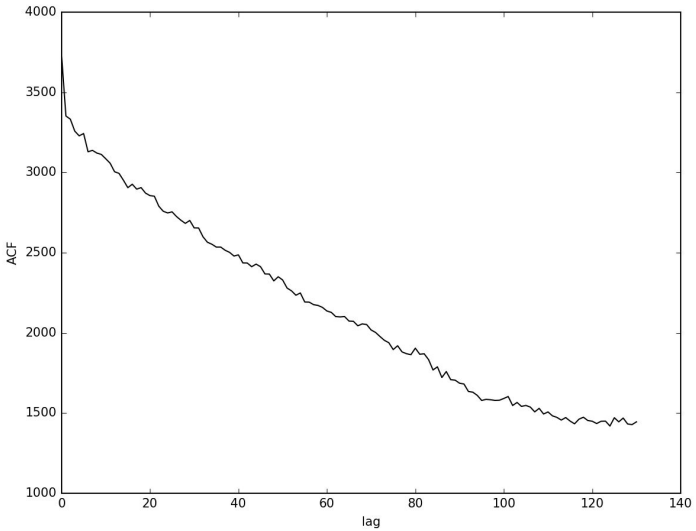
- One class SVC
- Requires identification of the “default” good route, annotation of periods to train on. Difficult for long paths.
- Sensitive to clock synchronization issues and high utilization effects
- Adding traceroute measurements complicated by equal-cost multi-path routing (multi-IP devices, failover routing), and frequent ICMP packet drops.



Work done by **Xinran Wang**

First crude predictions





Throughput measurements are low frequency

Does not show any time dependency

Both simple rolling window mean and EWMA give reasonable short term prediction.

Understanding correlations

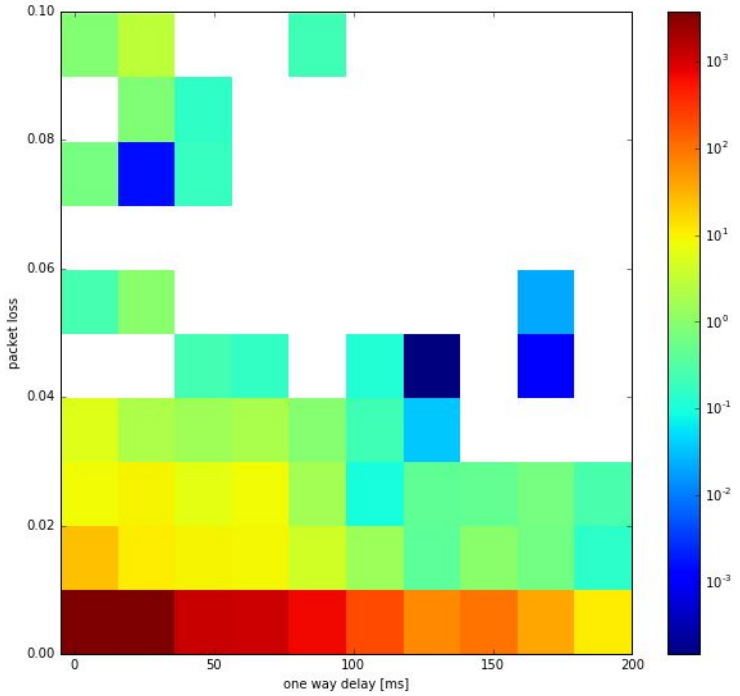
From Mathis et al. Rate < (MSS/RTT)*(1 / sqrt(PL))

MSS - maximum segment size

RTT - round trip time

PL - packet loss

It gives no predictive value for packet loss lower than our sensitivity.



All links, avg. throughput over one week.

Detecting high utilization

It would be prohibitively expensive to run frequent throughput measurements in order to get link utilization estimates.

High utilization can be detected through changes in high frequency packet loss and one way delay measurements (10Hz).

Delays increase since devices will keep packets longer in buffers.

Packet loss starts as buffer fill.

To validate that perfSONAR measurements are sensitive enough to detect latency and loss due to network congestion we compare perfSONAR data with router utilizations on networks which have simple topology.

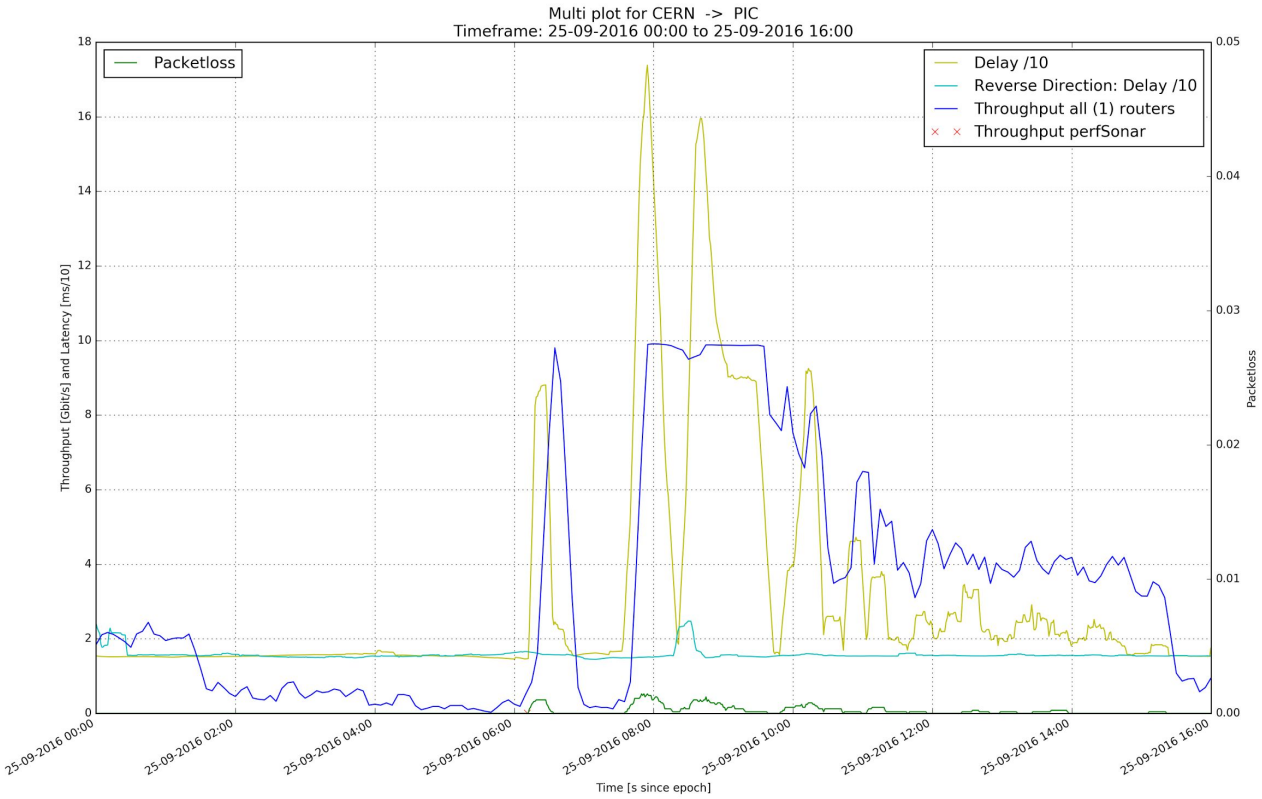
LHC private optical network (LHCOPN) chosen due to single hop links, device throughput controlled and data provided by CERN (**Edoardo Martelli**).

Detecting high utilization

Work done by **Hendrik Borras** and **Marian Babik**.

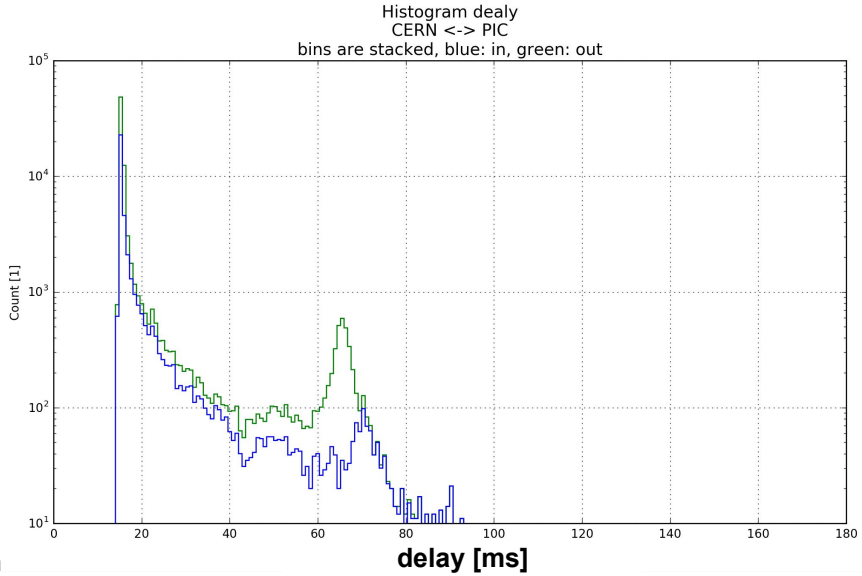
Similar pattern visible on all LHCOPN links.

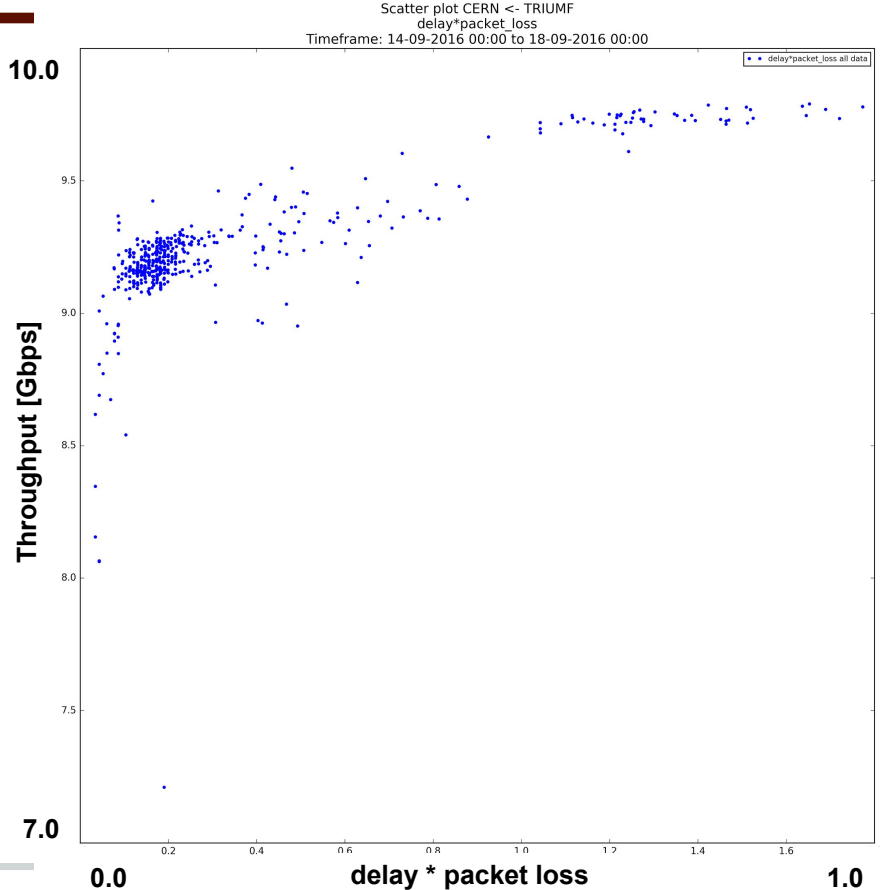
Both delay and packet loss very sensitive to high utilization.



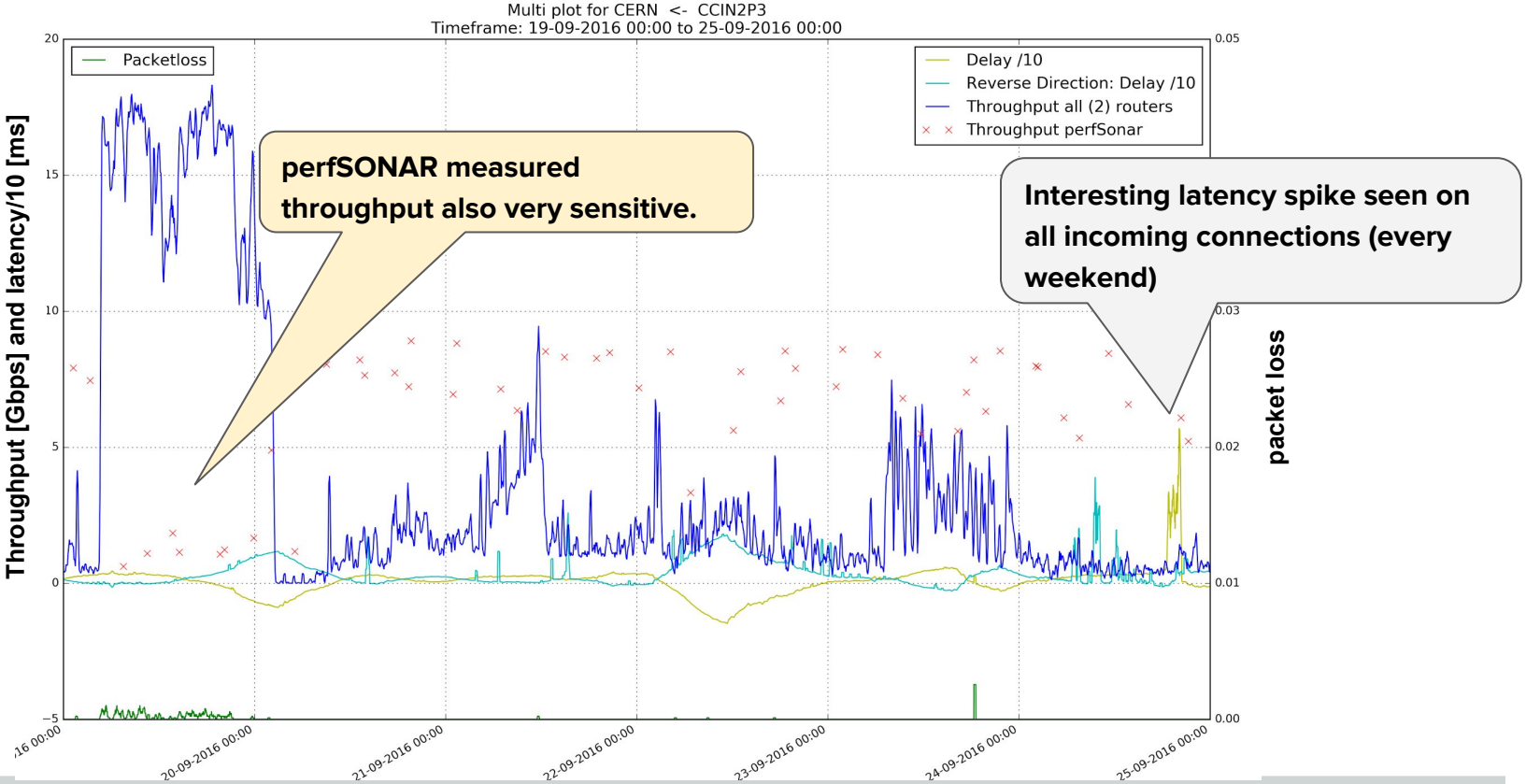
Detecting high utilization

Even simple cut would probably work.





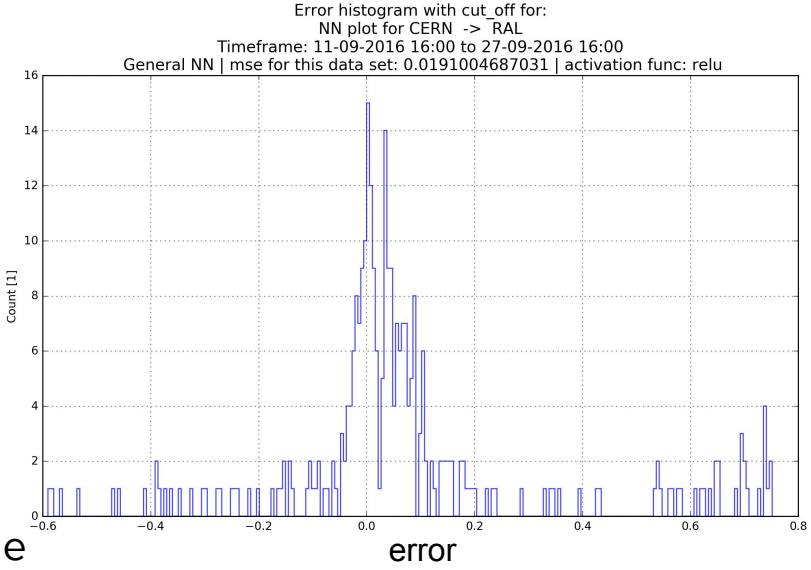
Detecting high utilization cont'd



Detecting high utilization cont'd

Simple cut picks most high utilization periods:

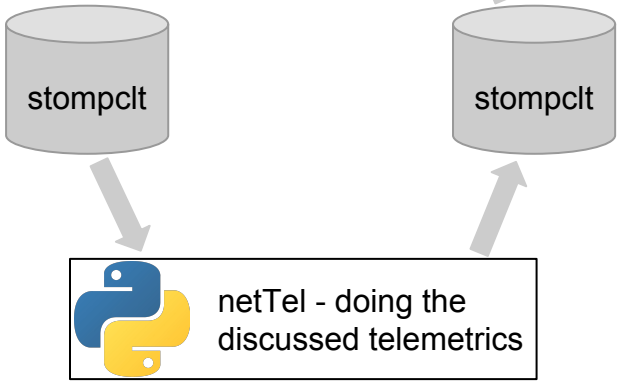
- Packet loss > 0.0001 && OWD > 1σ



Feedforward NN:

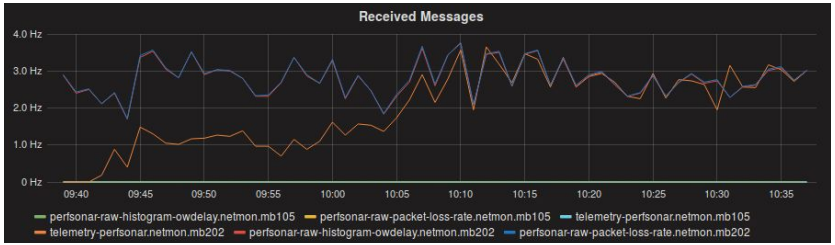
- Inputs - 15 previous PL and OWD measurements (15 min span), mean and variance values of previous one week.
- 3 layers, one output, rectified linear unit activation
- Trained to predict utilization between 70 and 100%
- Investigating one network per site, longer time frame recurrent NN.

Predictions in production



netTel

- Implements both empirical and machine learning approach
- Buffers up to 32 minutes of raw data
- Uses: Keras, Theano, scikt-learn, pandas and numpy
- Single threaded, under 500 MB memory usage
- Publishes results to ActiveMQ, full buffers after ~40 min

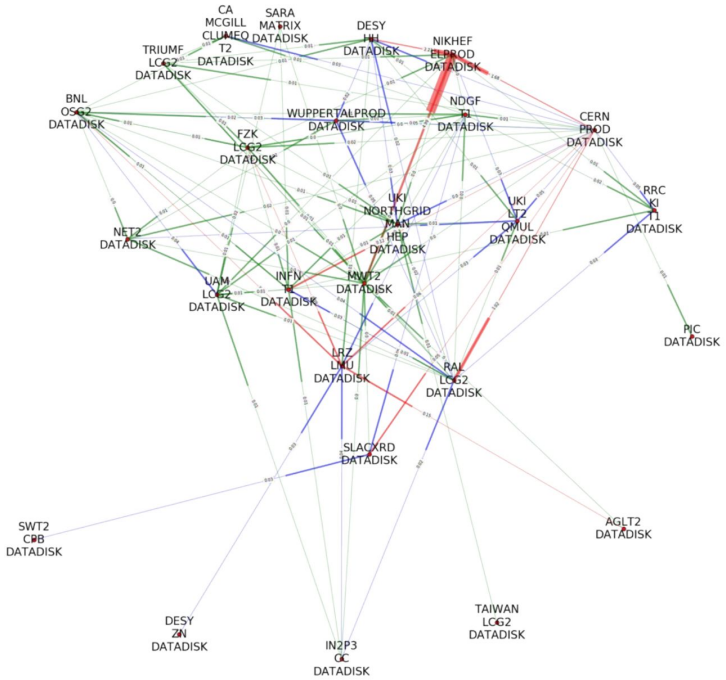


ATLAS usage of network info

Job brokering for remote data access uses FAX cost values as these directly map and is being revised.[1]

DDM topics[2]

- Estimator of time-to-complete transfer using random forest regressor of many trees
- Heavy Ion data placement. Not only network metrics but also storage space, CPU resources come into play.



[1] ATLAS WORLD-cloud and networking in PanDA
[2] Machine Learning for ATLAS DDM Network Metics

Conclusions

- A wealth of network related data collected in ATLAS analytics platform.
- All analysis software tools are in place.
- First steps in data understanding and cleaning are being made.
- Simple cut based filters can already be used for alarm and alert generation.
- First tests of different ML methods for finer alarms / predictions performed.
- Still a lot more data investigations to be done, but that's the most interesting part.