Analytics Platform for ATLAS Computing Services

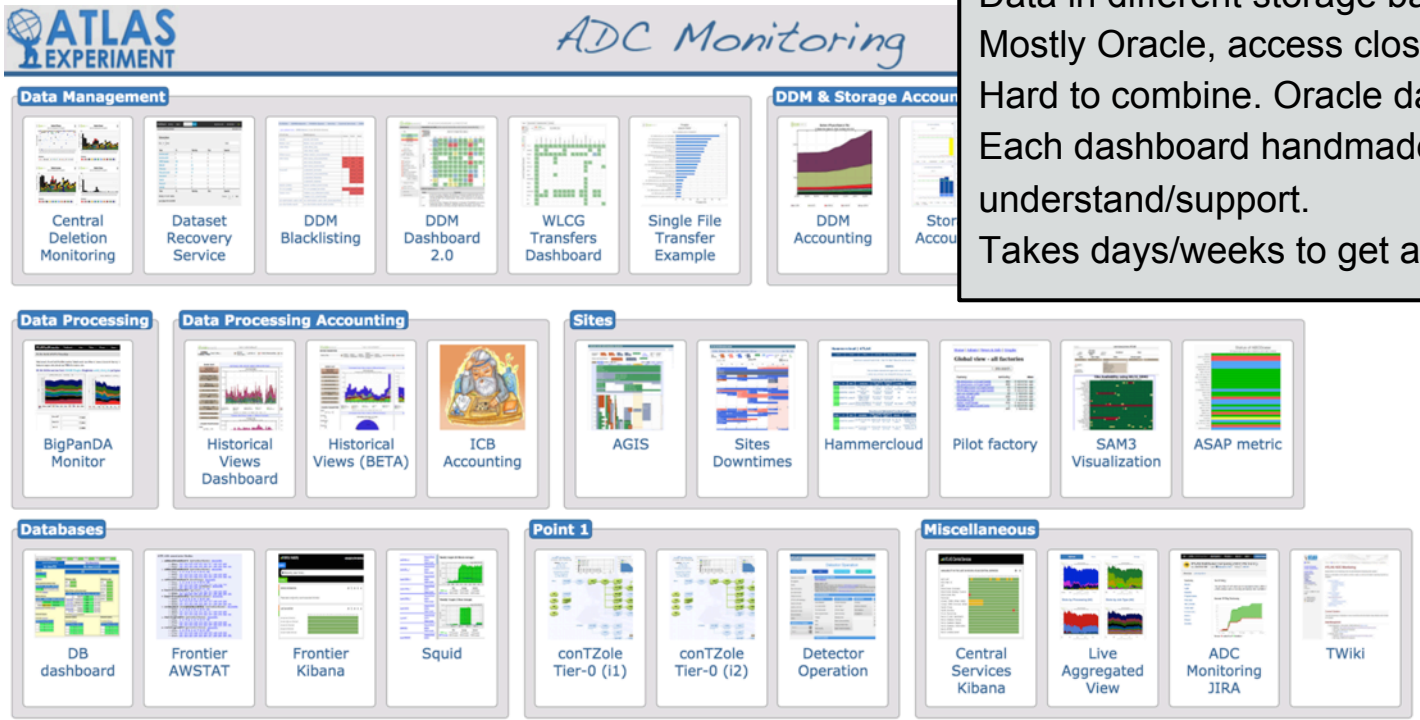
Ilija Vukotic
for the ATLAS collaboration

ICHEP 2016, Chicago, USA

Getting the most from distributed resources

What we want	What we need
To understand the system	A way to easily get global picture
To understand interplay of different systems and services	Collect all the data at one place. Be able to cross-reference.
Debug systems	Ability to drill down to the most detailed information
Run simulation, test models	Programmatic access to all the data
Alerts, sensing services	Continuously / periodically running services operating on raw / derived information fast enough for a real time feedback

Year ago

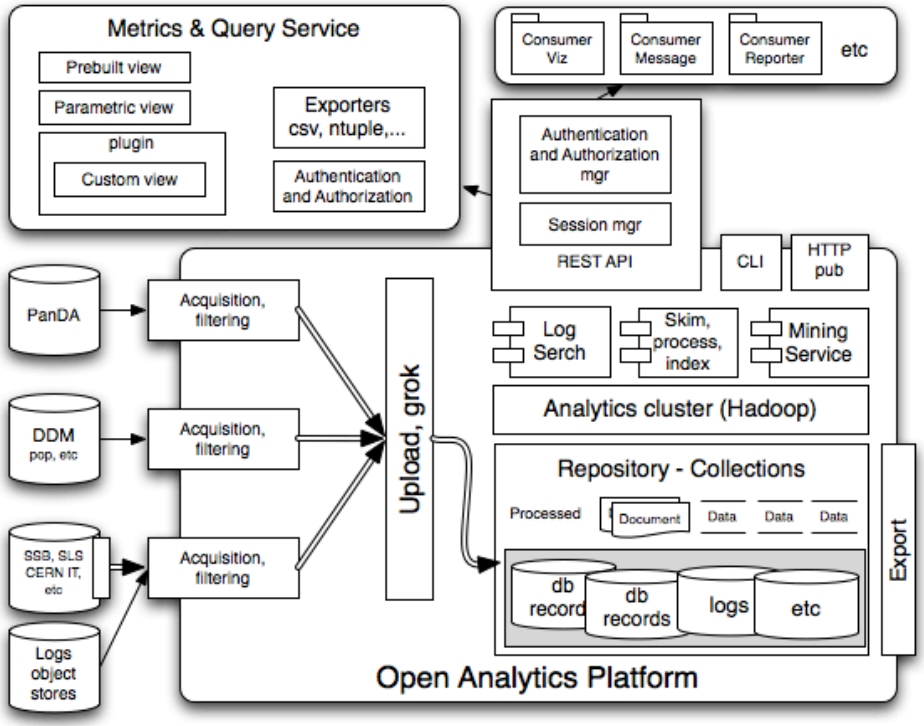


Data in different storage backends.
Mostly Oracle, access closed off.
Hard to combine. Oracle data mining tools not used.
Each dashboard handmade, each different. Hard to understand/support.
Takes days/weeks to get a plot added.

Architecture

Main functions

- Acquisition, filtering and upload of data sources into a repository
- Hadoop cluster for analysis of multiple data sources to create reduced collections for higher level analytics
- Serve repository collections in multiple formats to external clients
- Makes collected sources available for export by external users
- Host analytics services on the platform such as ElasticSearch, Logstash, Kibana, etc.



Data sources

PanDA - a data-driven workload management system for production and distributed analysis processing
Rucio - a Distributed Data Management system used to manage accounts, files, datasets, and distributed storage systems.
FAX - Federated ATLAS storage system using XRootD protocol. Provides a global namespace, direct access to data from anywhere.
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.
FTS - File Transfer Service - the lowest-level data movement service doing point-to-point file transfers.
xAOD - primary analysis data product.

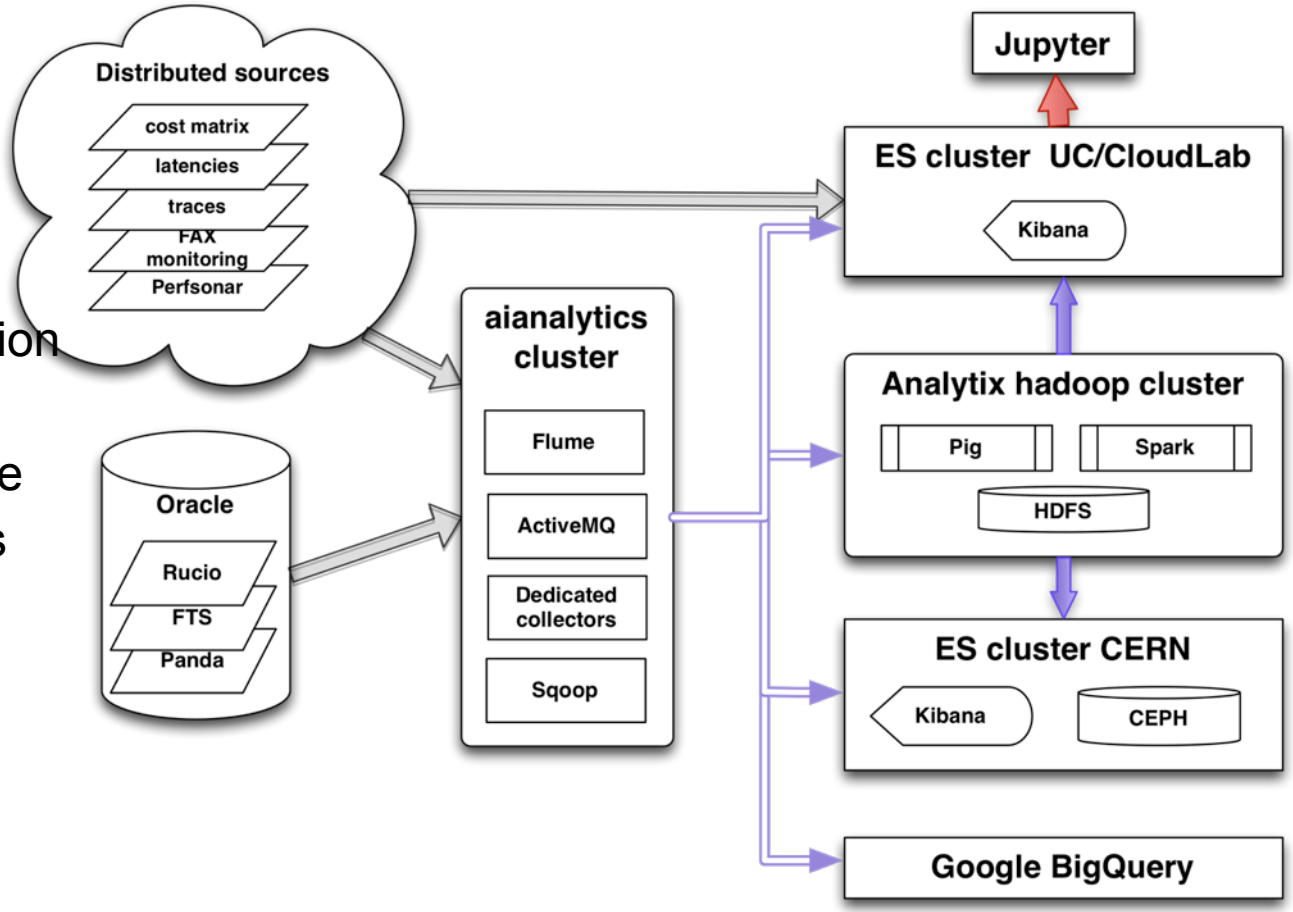
Resources

aianalytics cluster
- 5 VM nodes
- Mainly for data ingestion

Analytix Hadoop cluster
- Base load map-reduce
- Periodic indexing jobs

Elasticsearch clusters
- Runs Kibana
- Backend for Jupyter

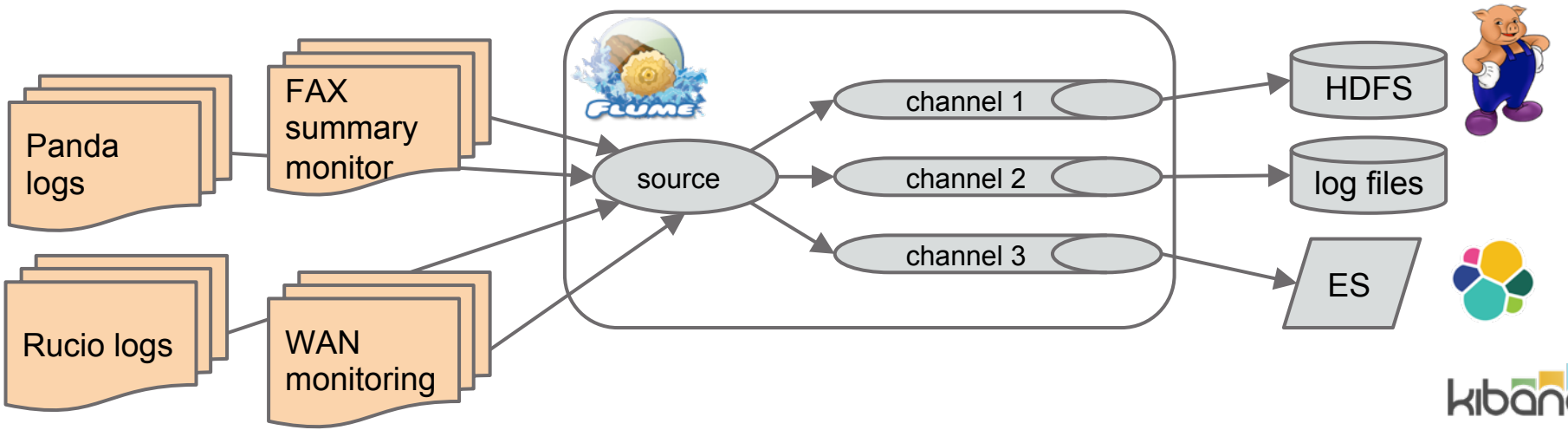
Google BigTable
- Extreme performance
- Investigating cost



Central Flume Collector

Listens for JSON messages (from multiple WLCG or ATLAS services)

"events" multiplexed into different memory channels based on header content, and sent to log files and/or HDFS for analysis and/or ElasticSearch for indexing



elastic @ CloudLab & THE UNIVERSITY OF CHICAGO

Clemson

1 indexing node
5 data nodes

- 20 cores per node (2 CPUs)
- 256 GB RAM
- 2x10 Gb/s Ethernet card
- 40 Gb/s Infiniband
- 2x1 TB disk drives in each node.

University of Chicago

3 master nodes (VMs 2 cores, 8 GB RAM)
1 client node (VM 2 cores, 16 GB RAM)
5 data nodes

- 24 cores per node (2 CPUs)
- 48 GB RAM
- 1 Gb/s Ethernet card
- 3x1 TB disk drives in each node.

Jupyter

an interactive computing environment that enables users to author notebook documents that include: live code, plots, interactive widgets, documentation

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 /

At CERN

- SWAN (Service for Web based Analysis)
- Currently in Beta
- Code in CERNBox

Full set of ML packages installed

Root numpy
Numpy
Scipy
Matplotlib
Pandas
XGboost
Scikit Learn
Scikit images
Theano
TensorFlow
Keras
H5py
BLAS / ATLAS / LAPACK libraries
Cuda and CuDNN
ZLib

Data sizes

Hadoop-based collections

- Rucio (42 TB)
- PanDA Job Archive (2.4 TB)
- PanDA State change logs (0.5TB)
- PanDA Logs (4.5 TB)
- Network data(2-3 GB/hour)
- FAX cost matrix, traces (20 GB)

In Elasticsearch

Indices: **1767**	Memory: **259GB / 433GB**	Total Shards: **15733**	Unassigned Shards: **0**	Documents: **8,509,799,053**	Data: **9TB**	Uptime: **3 months**

Analytics Infrastructure - lessons learned

Need really good hardware - lot of RAM, SSD caching, large disks

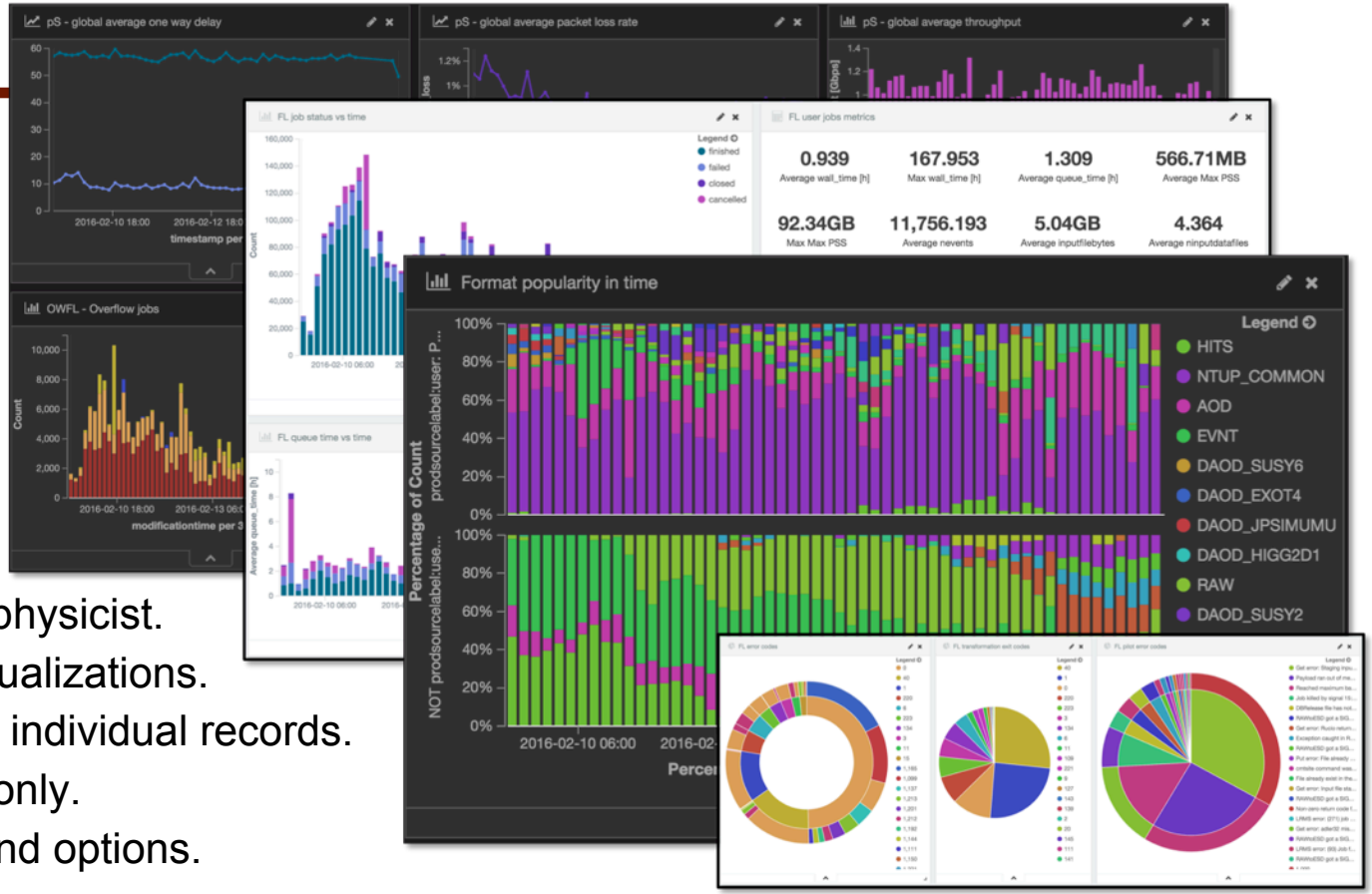
Elasticsearch backups

while most of the data could be re-indexed, some exists only in ES.

Non-negligible learning curve (MR, pig, java, jython) - need a lot of documentation, support, education.

It is clear that Elasticsearch indexed data have much more use. Try to index as much as possible of the data.

Monitoring
Kibana

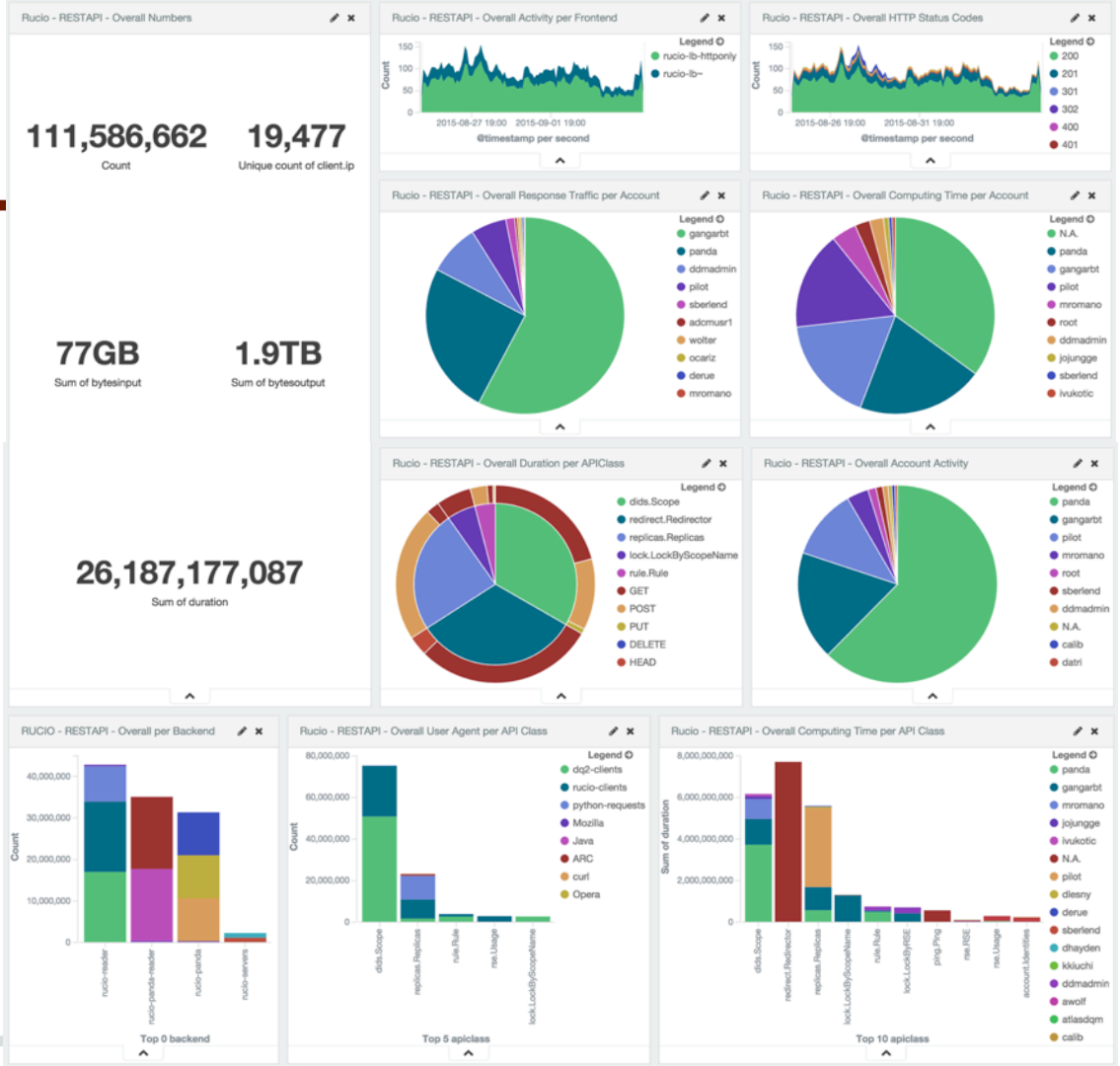


Very fast, easy to use.
A bit counterintuitive to a physicist.
Easy to create custom visualizations.
Easy to drill-down to even individual records.
Can do simple arithmetic only.
Limited set of plot types and options.
Embeddable visualizations.

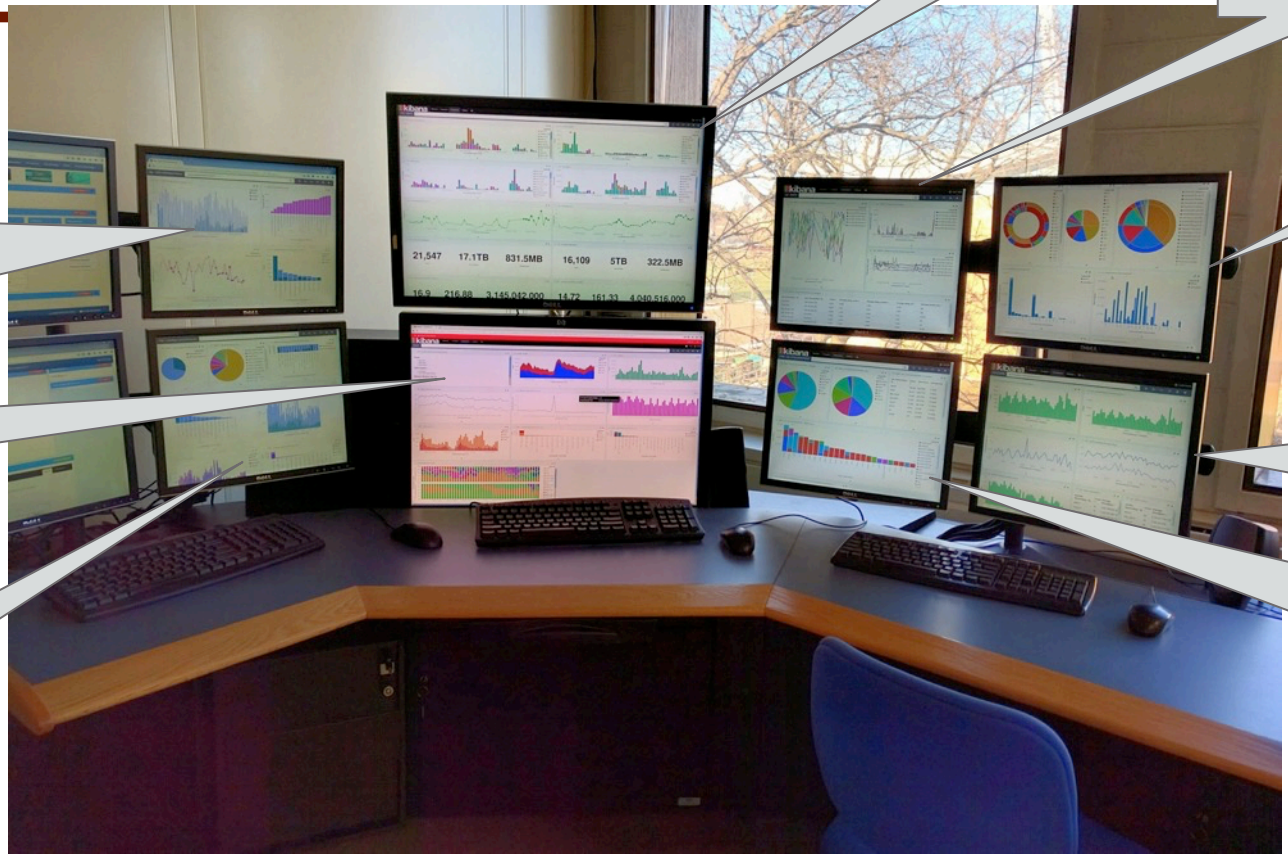
Monitoring

Kibana

Rucio account activity



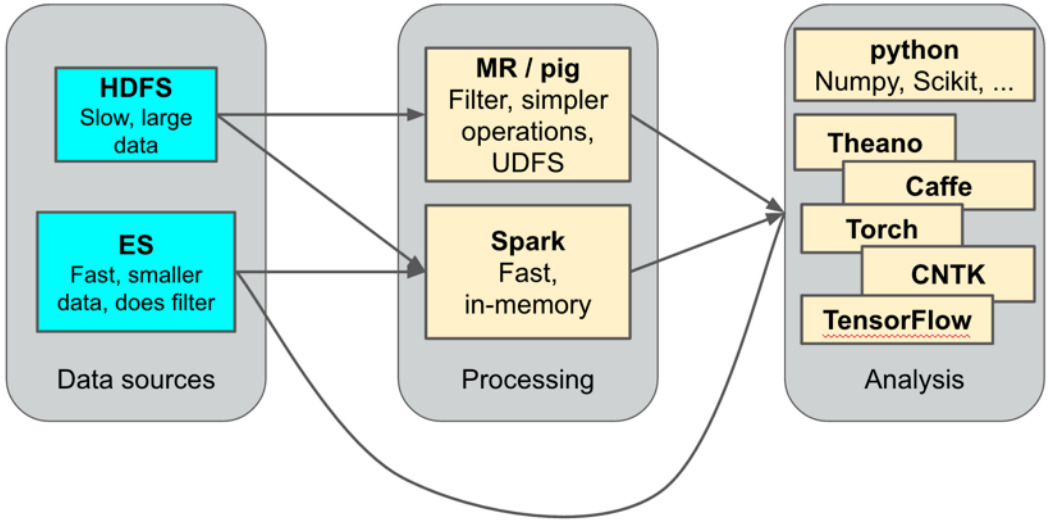
Analytics control center



In-depth analysis

Examples:

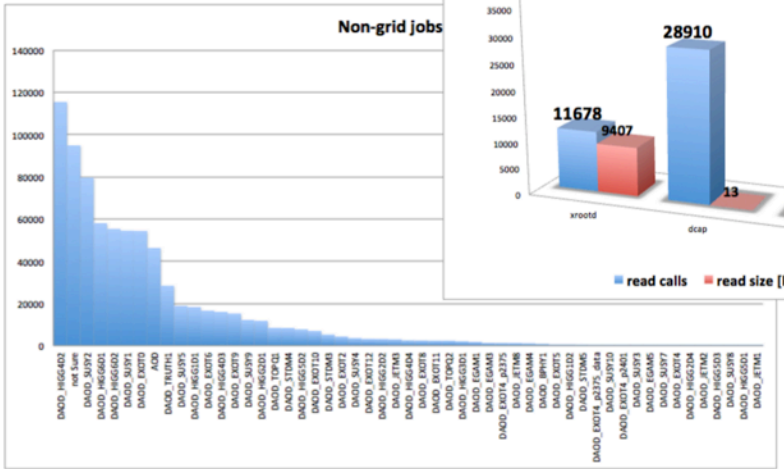
- Predicting a network link performance
- Testing different alerting models
- Modeling data replication algorithms
- Testing caching models

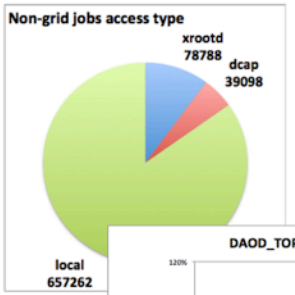


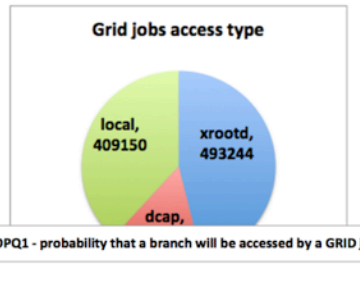
Requirements:

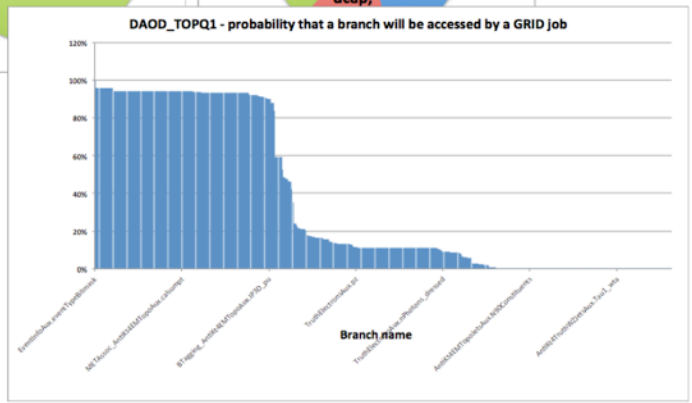
- order of ten users
- no expectation of immediate results
- need to go through much more data than monitoring/alerting
- new tools/frameworks
- different hardware (more memory, GPUs)

In-depth analysis example





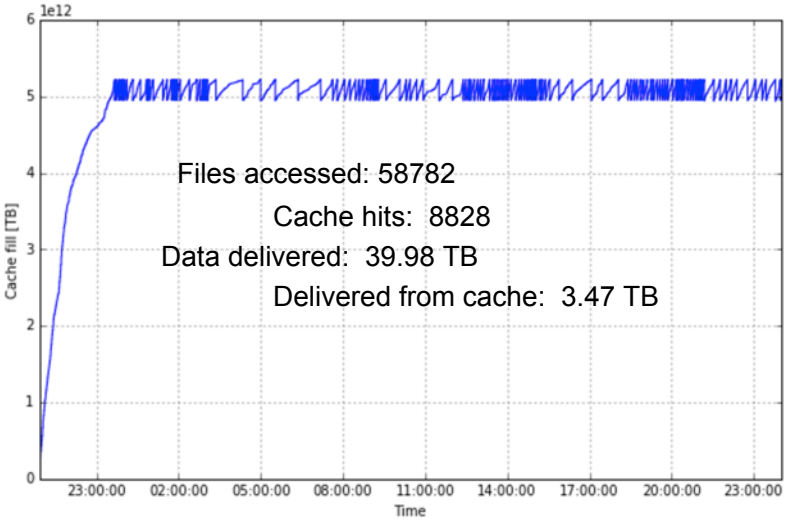


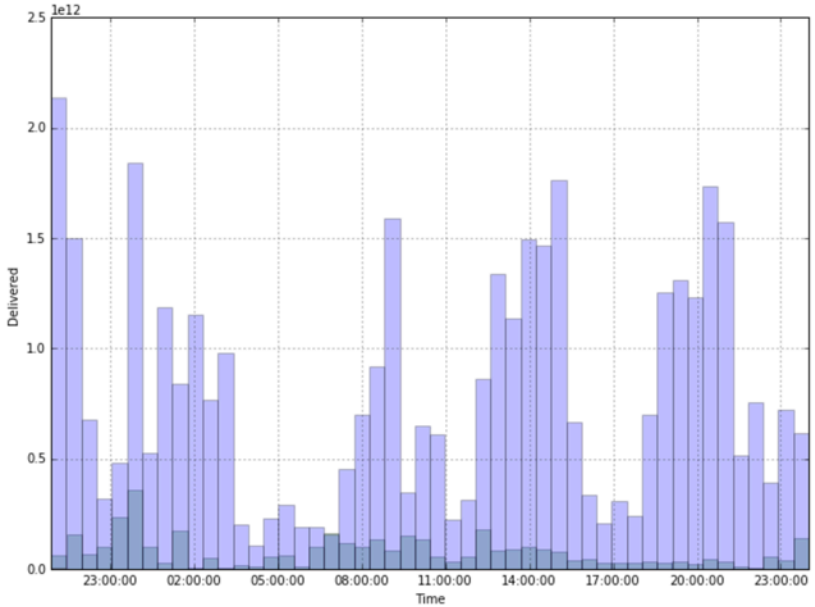


xAOD accesses analysis

Run a model

In ES we have detailed monitoring data for all remote data accesses: what file was accessed from where, by whom, how much was read.
We use historical data to run different models and parameters of data caching (cache size, cache block size, high/low watermarks, …) and calculate cache hit rate, turnover, etc.





Jupyter-based analyses

Understanding event timings for different

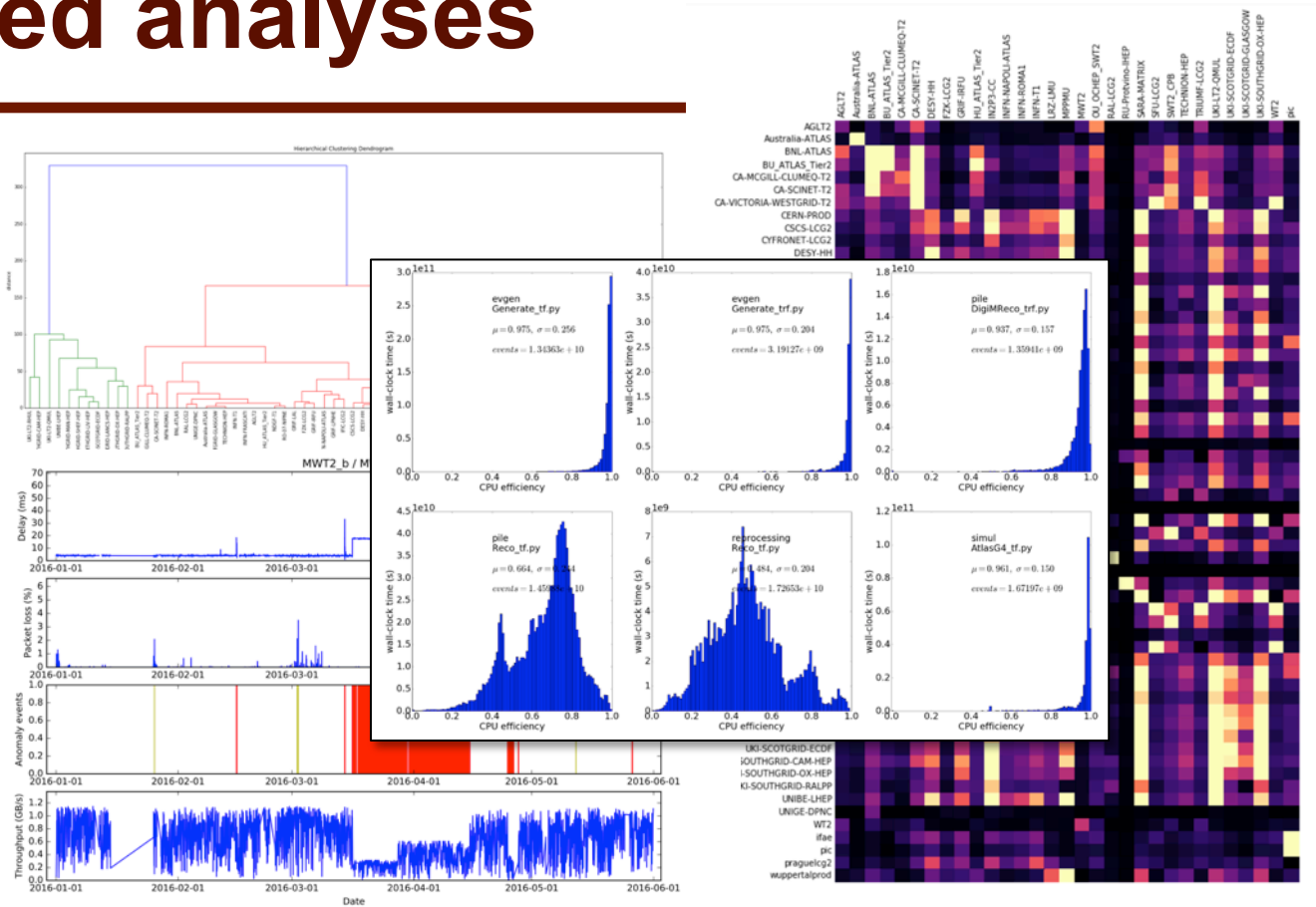
- Processing steps (generation, simulation, reconstruction,…)
- CPU types
- Sites

Memory usage

CPU efficiencies

User analysis patterns

Network closeness



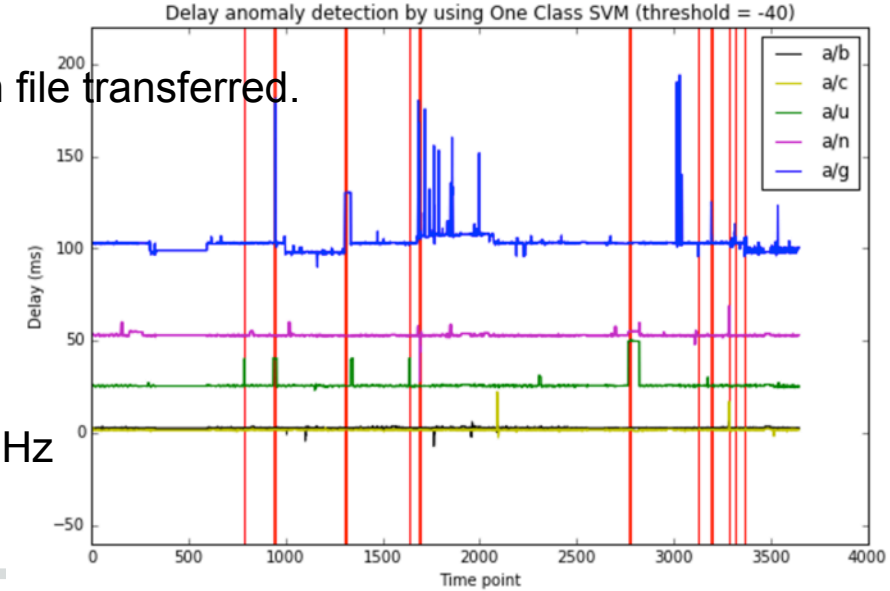
A network weather service using ES

Data sources:

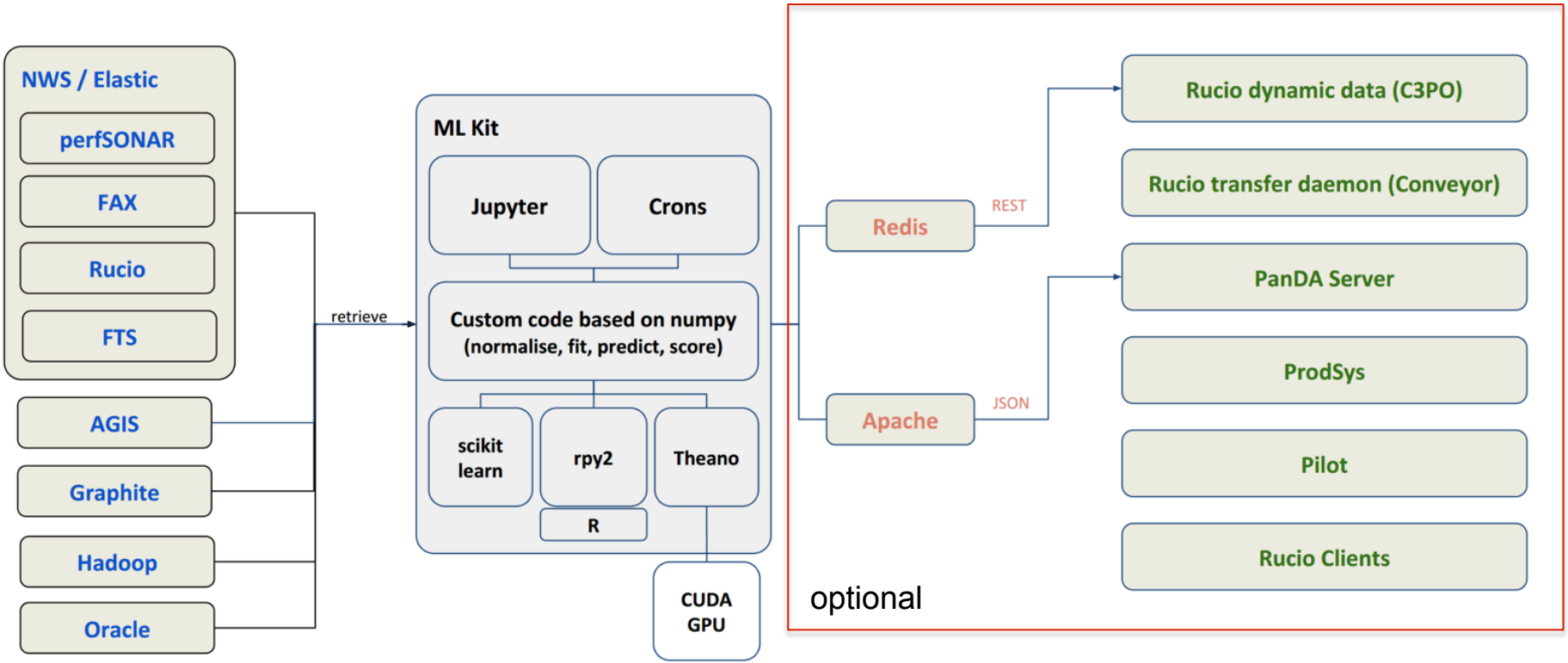
- **PERFSonar** - Throughput, latencies, and packet loss data come from OSG network datastore, collected from AMQ.
- **FAX cost** – Measures remote data access from remote storages to computing site's worker nodes.
- **FTS** – File transfer service gives all info for each file transferred.

Services:

- Alerts on changes in network performance
- Prediction of future network performance
- Simple REST interface and python API
- ES is able to deliver searches in <100ms @ 100Hz
- Not limited to ATLAS



Machine learning for network awareness



Conclusions

The new stack of Big Data tools (Hadoop, Flume, Sqoop, Logstash, pig, ES, Kibana, Jupyter) provide a great platform for ATLAS Distributed Computing analytics tasks:

- Horizontally scalable
- Performant
- Simple to develop for
- Easy to use for an analyzer
- Fast to make custom dashboards, GUIs
- Can replace other full fledged services (alerting, reporting,…)
- Great backend for in-depth analysis, ML