Analysing perfsonar data with elasticsearch

Rolf Seuster
University of Victoria

TOC:
ElasticSearch at UVic
PerfSonar data into ES
What can we do with it now ?

Introduction

- work presented here based on new initiative between Canadian research network community:
  HepNet, Canarie, ComputeCanada + several NREN
- aim is to create automated warnings and alarms in case networking metrics indicate problems in Canadian research network
  - wide range of technologies: 2x100G nationwide to 1G local connection of some small universities
- should leverage as much as possible capabilites provided by FOSS as ElasticSSearch, Kibana / graphana

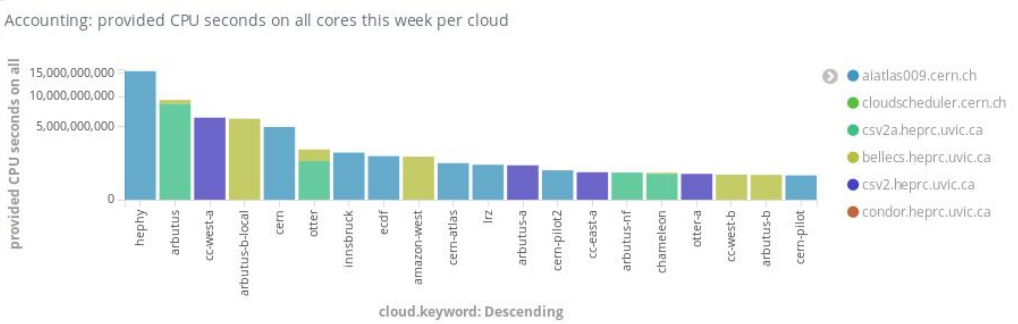
What is ElasticSearch Logstash / Kibana (ELK)?

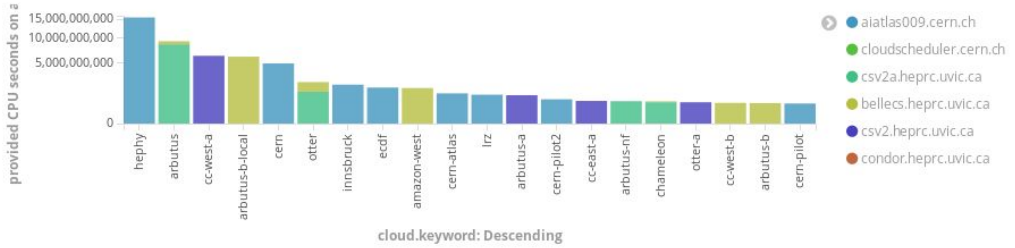
- From https://www.elastic.co/what-is/elk-stack:
  "ELK" is the acronym for three open source projects: Elasticsearch, Logstash, and Kibana. Elasticsearch is a search and analytics engine. Logstash is a server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms it, and then sends it to a "stash" like Elasticsearch. Kibana lets users visualize data with charts and graphs in Elasticsearch.
- Plenty of plugins available, often plugins / dashboard for common tasks already exists (mostly monitoring health of servers by looking at CPU/network/disk…)
- common now also in (LHC) experiments for job monitoring
  - CERN IT's monitoring of lxplus/lxbatch/ other servers in ELK first I saw many years ago

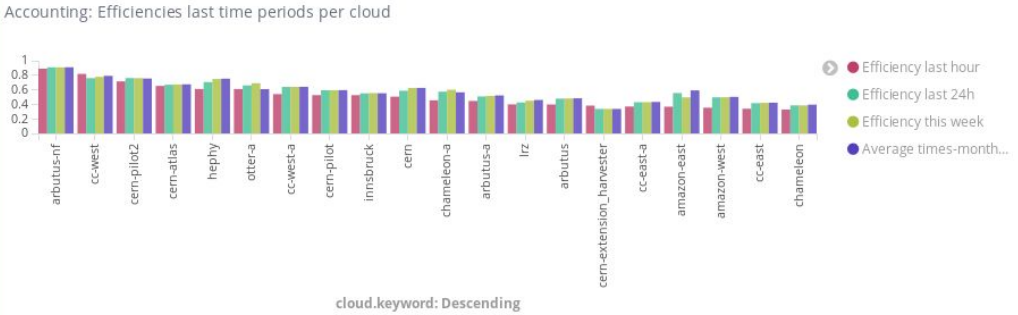
ElasticSearch at UVic

- Our first installation in 2016 or so. We use it for
  - Accounting (1st application, most important for us)
  - Job monitoring for ATLAS + Belle (our 2nd application)
  - General monitoring (logfiles, disk usage, etc.)
  - code re-usage (similar python sniplets ~ everywhere...)
- grew bigger over time, as it's still a very convenient way of displaying data which change frequently over time, however
  - sort of steep learning curve
  - things usually turn out way more difficult than they should be ...

Accounting

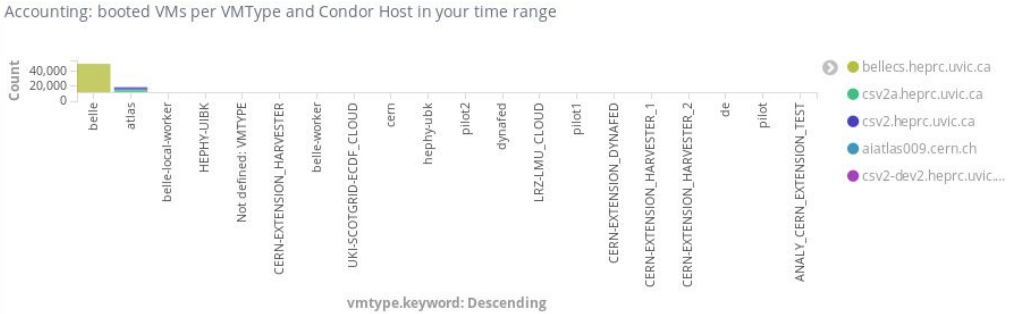


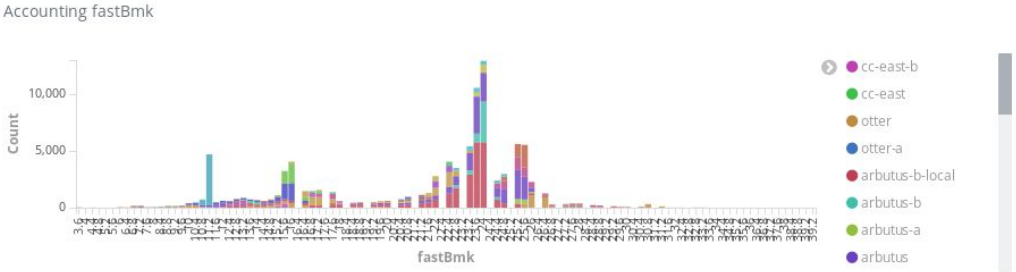




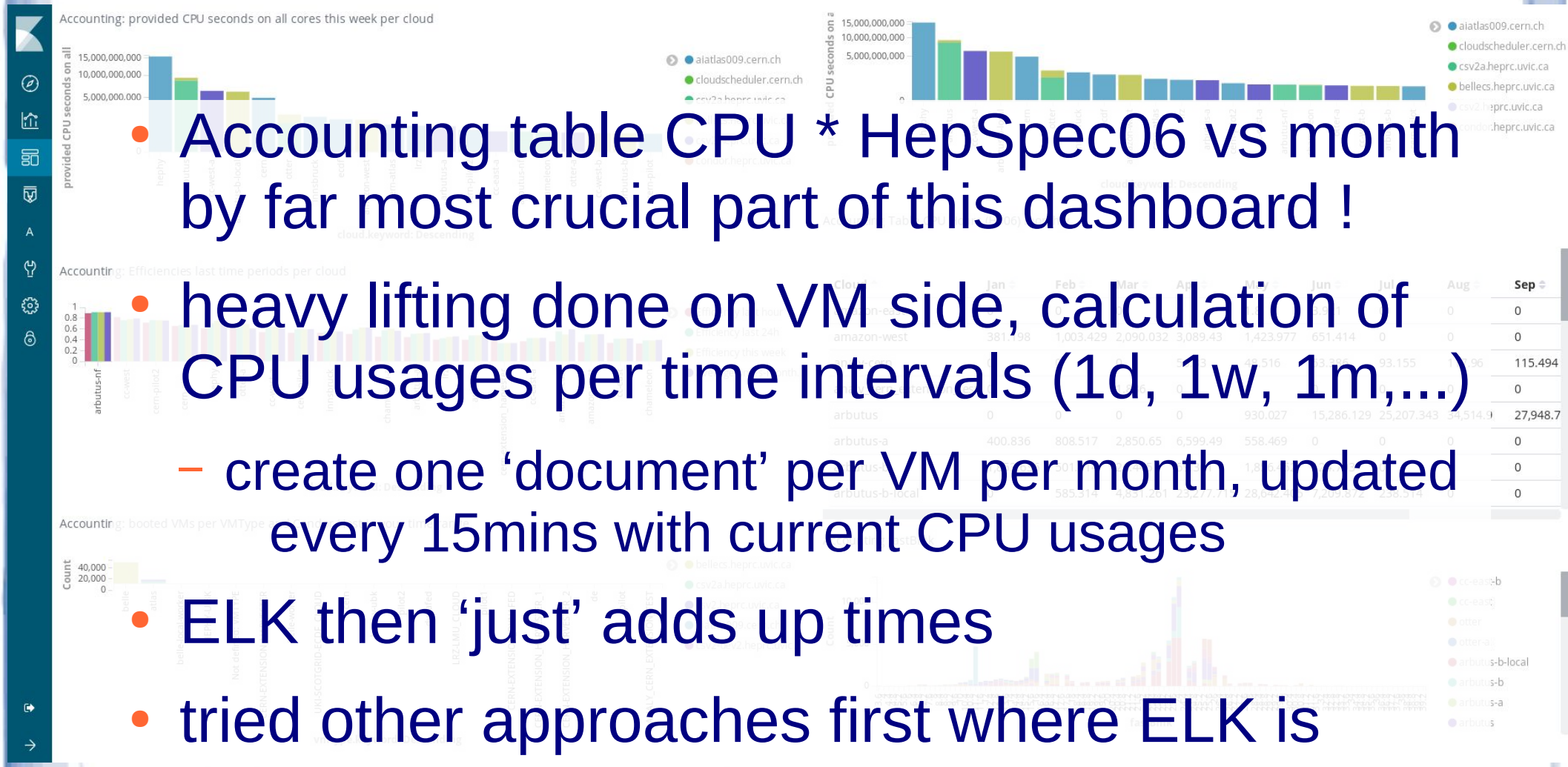
Accounting Table CPU Hours (HS06) monthly

Cloud	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
amazon-east	0	0	0	0	1.8	3.921	0	0	0
amazon-west	381.198	1,003.429	2,090.032	3,089.43	1,423.977	651.414	0	0	0
analy-cern	0	0	0	56.33	48.516	53.386	93.155	117.96	115.494
analy_cern_extension_test	0	0	1.836	0	0	0	0	0	0
arbutus	0	0	0	0	930.027	15,286.129	25,207.343	34,514.9	27,948.7
arbutus-a	400.836	808.517	2,850.65	6,599.49	558.469	0	0	0	0
arbutus-b	1,853.636	501.414	26.445	59.361	1,856.452	124.793	0	0	0
arbutus-b-local	0	585.314	4,831.261	23,277.715	28,642.405	7,209.872	238.514	0	0



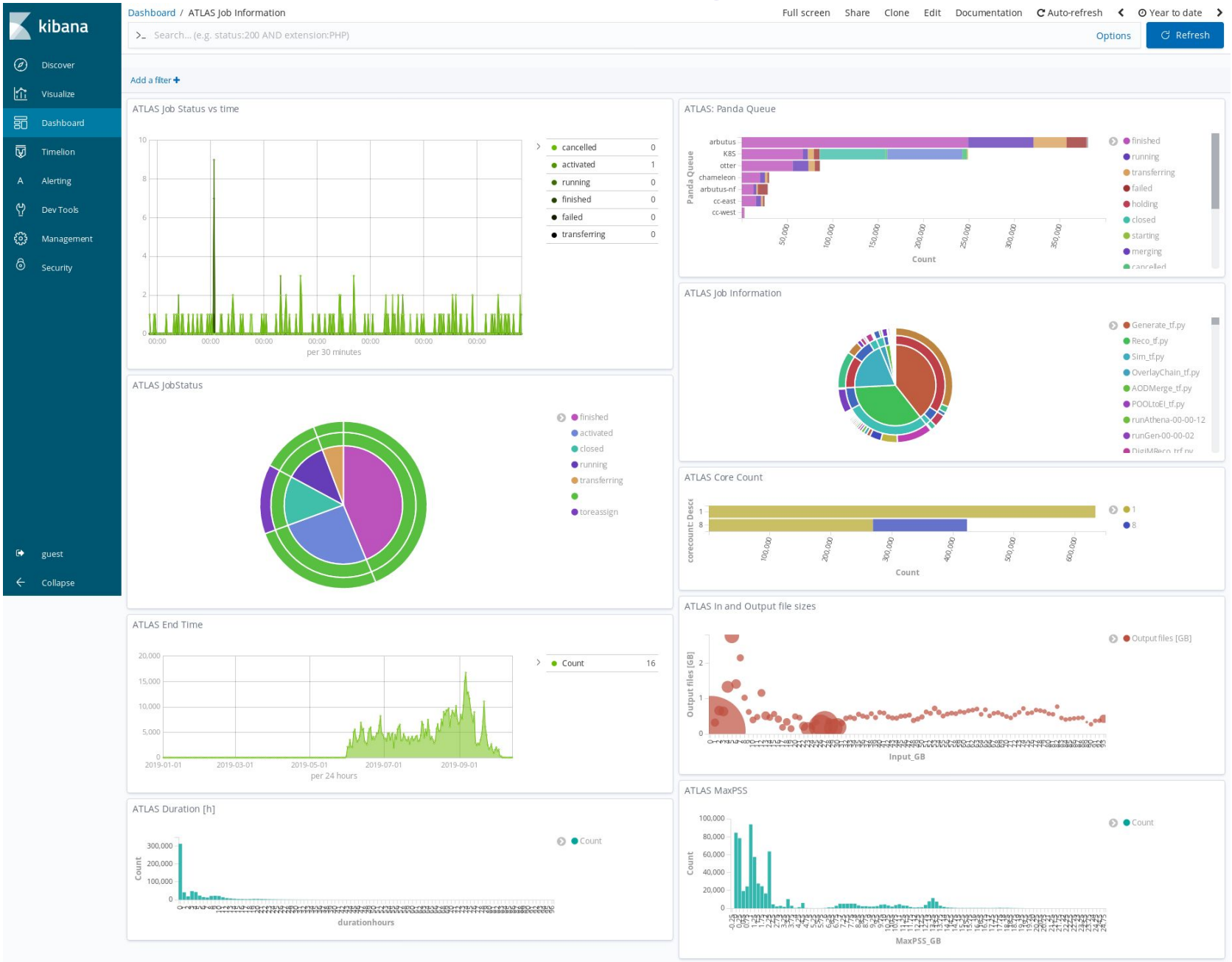


Accounting – some details

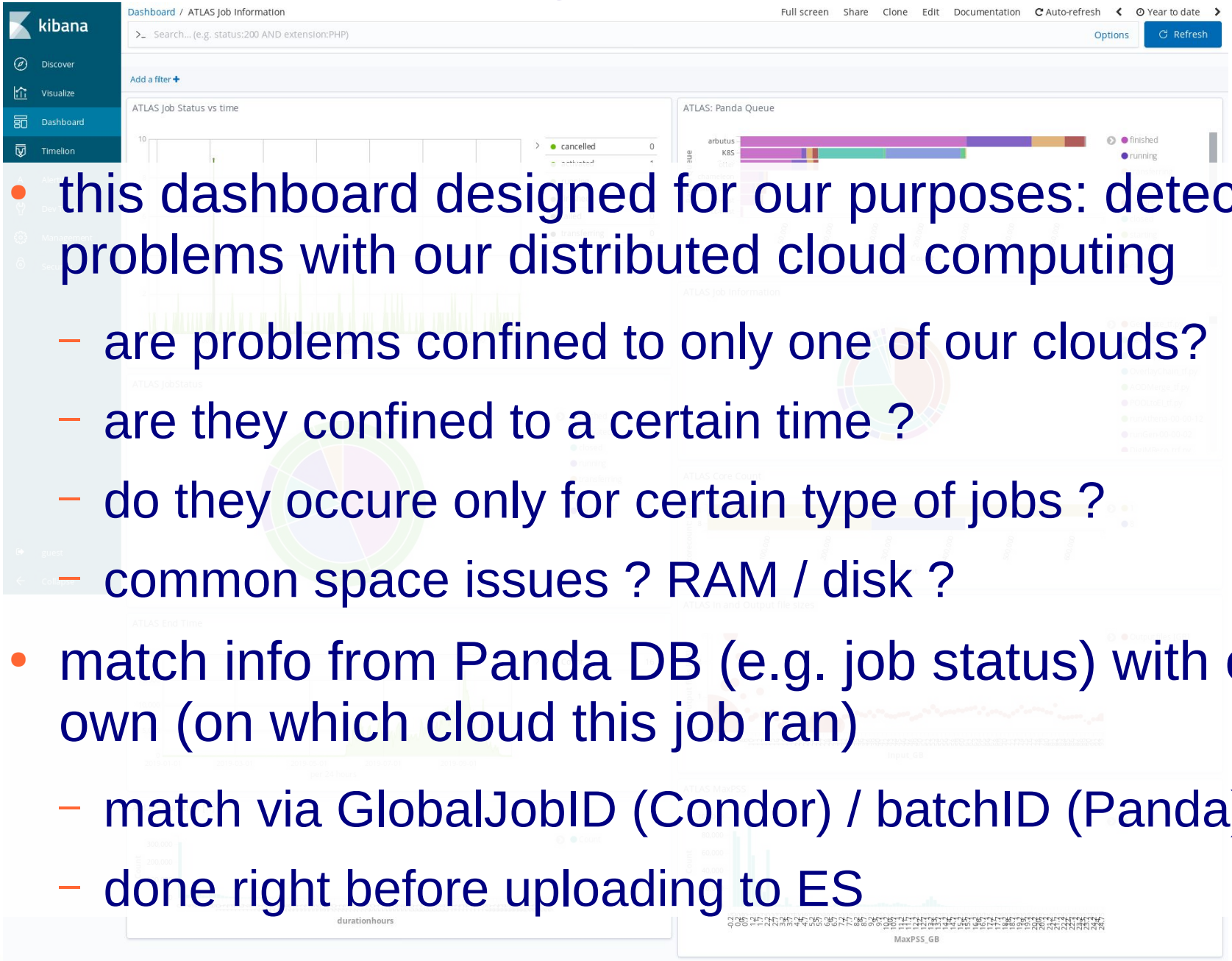


- Accounting table CPU * HepSpec06 vs month by far most crucial part of this dashboard !
- heavy lifting done on VM side, calculation of CPU usages per time intervals (1d, 1w, 1m,...)
  - create one ‘document’ per VM per month, updated every 15mins with current CPU usages
- ELK then ‘just’ adds up times
- tried other approaches first where ELK is doing most calculation
  - almost impossible ...

Job Monitoring – ATLAS

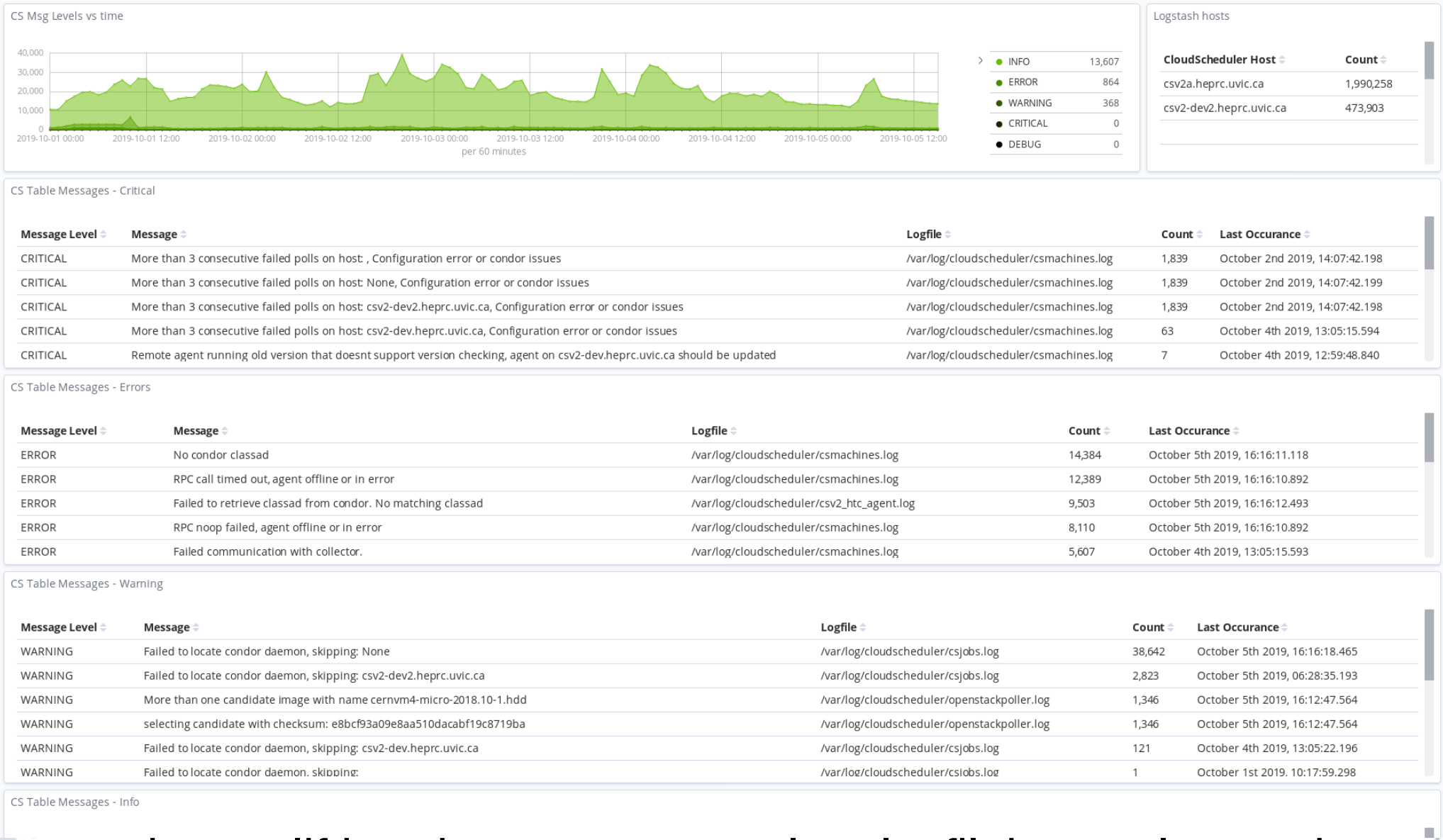


Job Monitoring – ATLAS – details



- this dashboard designed for our purposes: detecting problems with our distributed cloud computing
  - are problems confined to only one of our clouds?
  - are they confined to a certain time ?
  - do they occure only for certain type of jobs ?
  - common space issues ? RAM / disk ?
- match info from Panda DB (e.g. job status) with our own (on which cloud this job ran)
  - match via GlobalJobID (Condor) / batchID (Panda)
  - done right before uploading to ES

Logstash for our CloudScheduler



→ heavy lifting done my external tools: filebeat + logstash

my own Experience with ELK

- ES not good at doing heavy lifting, pre-process input data as much as possible to suit your use-case !
- ES/Kibana then for ‘just’ displaying the data, few and simple calculations still possible in this step
- some things are plain impossible in ES, others really difficult to achieve
- Disclaimer: I don’t claim to be a real ES expert, but I do have spent hours and hours googling for many ‘simple’ solutions … Maybe my physics background prevents me from seeing the ‘beauty’ of ES ...

PerfSonar Data into ElasticSeach

- custom python script using python ABI from perfsonar
- some 550 lines, ~23kb length
  - python, json + pycurl (no new installation needed)
- runs every ~15-30 mins for about 2mins to upload data (most time spending uploading, plain runtime w/o upload is ~20 secs)
- currently uploads data from single host to ES, plan is to run it on all our perfsonar servers

re-assembling Data from perfsonar esmond DB

- data from ps-tool iperf3:
  - throughput, packet-retransmits simple numbers
  - data from subintervals stored in two different arrays, re-assembling based on timestamps
  - kibana cannot easily access ‘nested arrays’
- data from ps-tool packet_trace stored in nested arrays, but data added:
  - missing entries/names/rtt, added RTT of last hop in separate variable
  - list of all AS numbers in string, then hashed

re-assembling Data from perfsonar esmond DB, cont'd

- data from ps-tool powstream (10Hz ping):
  - full histogram of results in esmond → compute average and RMS and only store that in ES
- data from all tools stored in ES with their timestamp from esmond
  - shorter summary also stored in"summary" table → later more
- code is in github repo:
  https://github.com/ComputeCanada/NREN-perfsonar/tree/master/electicsearch-scripts

Example PS Dashboard

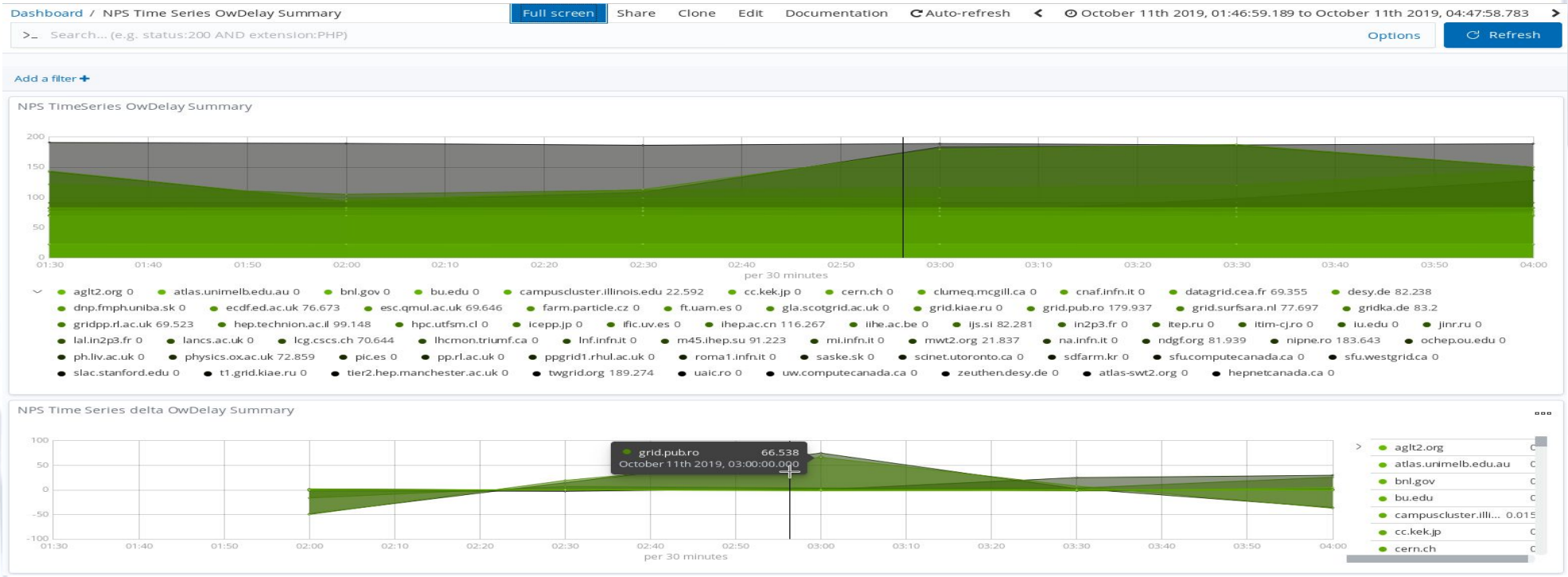
almost no pre-processing, only cuts applied in histograms

time series example: some pre-processing for OwDelay

- local time on some server seemed to be off ...

OwDelay,cont'd

- differential rate in 'visual builder', but now from a 'summary' (explained in next slides)



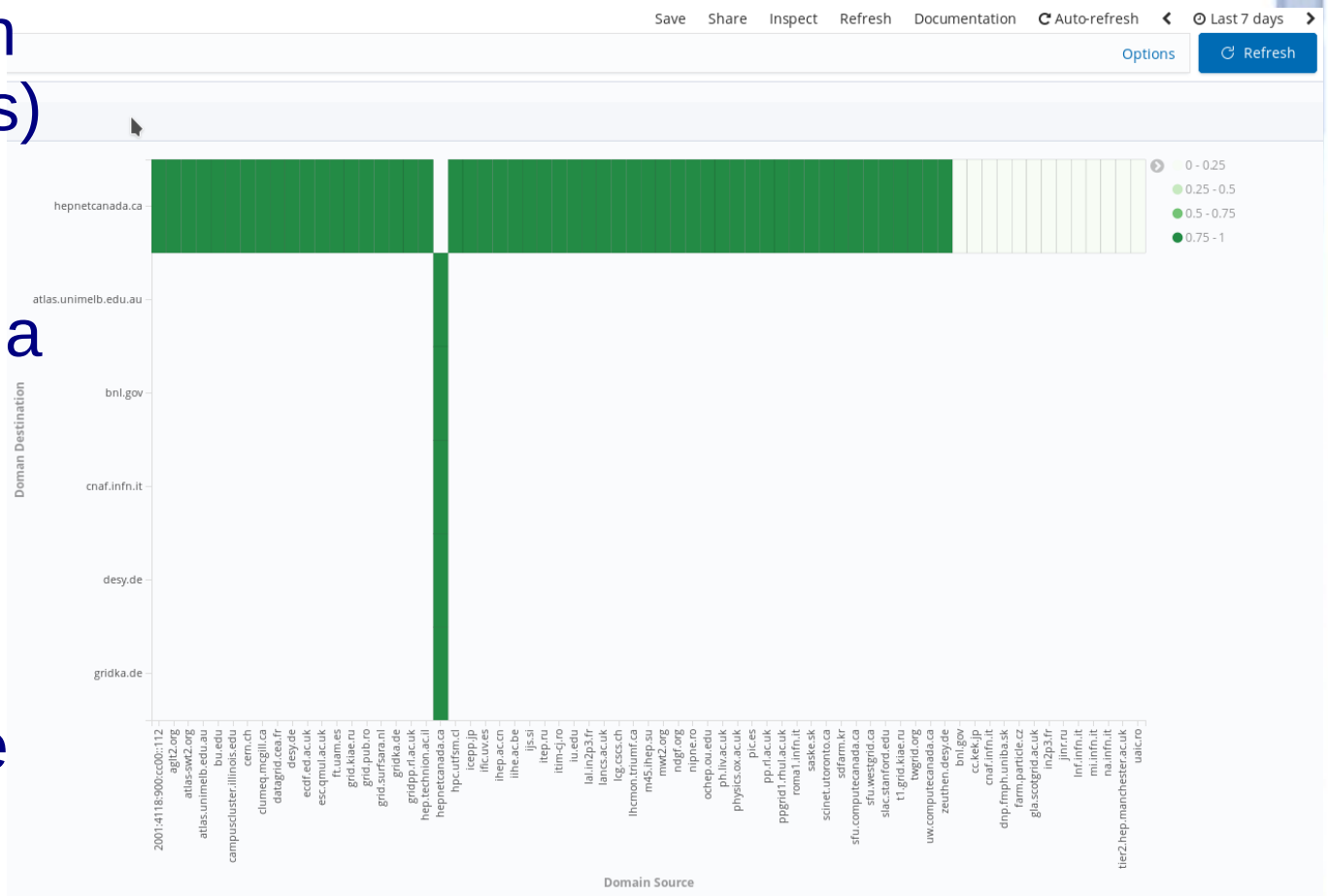
Creating ‘Summaries’

- uploading once per 15-30 mins allows one to create summaries: collect all measurements in this period from two ‘domains’ (source and destination) into one single ES document
- what does a domain mean here?
  - perfsonar runs two types of tests: latency and bandwidth tests, usually run from two different IPs(*) on same subnet
  - put results from both tests into same summary document
- e.g. allows for ‘simple’ IPv4/IPv6 comparisons...

(*) many modern server are capable of running both latency & bandwidth tests, both DB on same machine

Paths comparison IPv4 / IPv6

- green means path (from AS numbers) was identical at least once during last 7 days within a 30 min window (==time between uploads)
- Reminder: this is data from a single perfsonar server !



Combining measurements summaries

- similar plots could show e.g.
  - different RTT/latencies in IPv4/IPv6
  - throughput vs RTT / latency
- but still trying to figure out how to show this nicely with kibana …

First Look into Alarming

- our ELK installation (Amazon's opendistro) allows for some alarming capabilities:
  - able to sent alarms to slack, amazon chime and via simple web hooks
  - very first try with some simple tests and via web hook failed :(
  - ELK found plenty alarms but didn't seem to trigger the web hook …
  - needs certainly further investigation, no clear reason found in the various logfiles ...

my personal critics on ES / Kibana

- many ‘simple’ plots just not possible in kibana:
  - combination of variables from more than one document (except time delta of same variable)
  - real 2d (scatter) plots (the one they provide is not intuitive and not really working well ...)
  - difference between variable and its (time) average
- they have two plot-types for time series
- all plot-types are made differently, no uniq look & feel
  - some almost mouse only, some others text only …
- one scripting language is called ‘painless’, oh well ...

Future Plans

- many plots created within last few days, need to further explore possibilities now provided
- collect data from other perfsonar servers
  - bandwidth and latency in same domain
  - reverse path for traceroutes
- use bulk upload feature from elasticsearch python abi to speed up uploads
  - requires one pip install
- implement Alarms in Elasticsearch to really warn people of detected problems

Similar work

- ATLAS Analytics platform in Chicago Elasticsearch:
  - AFAIK, requires jupyter notebooks to analyse data
- Similar work started in US by Shawn McKee
  - would like to collaborate and share work/code and experience

Summary & Conclusions

- ES & Kibana are a beast to tame ...
  - … but can be a powerful & reliable tool once done !
- IMHO, in our field, pre-processing is key:
  - ES is just not made for our (un-)common use cases
  - maybe we also have way more intuitive tools at hand (?)

    e.g. $ python -c ‘import antigravity’

Questions ?

Backup: Collaboration in Canada

- new effort in Canada between Canarie, Compute Canada and several NREN's:
  - collect data centrally and analyse them to send out alerts / warning in case of new network problems occuring in Canadian R&E network,
    collect data from maybe 10-20 servers
  - effort started middle of 2019, want to have first system in place to further evaluate in 2020
  - looked at several solutions, but none seemed to do what we wanted to do (automated alerts etc..)

Backup: Security

- about same time when Ilija's ATLAS ES instance went down same thing happened to us, lost all data (don't really care, was of temporary interest anyway) but also all dashboards (hard work lost!)

  → spent time on backup data (only accounting data) + all dashboards (doesn't exist yet in Kibana !!)

- via npm package kibana4-backup, weekly crontask

- Installed opendistro's ELK stack, which comes with security modules by default (ES 7 now, too)

Time series: OwDelay

- busy plot with many endpoints

Time series: OwDelay

- selecting just one endpoint out of many…