Accounting and Benchmarking

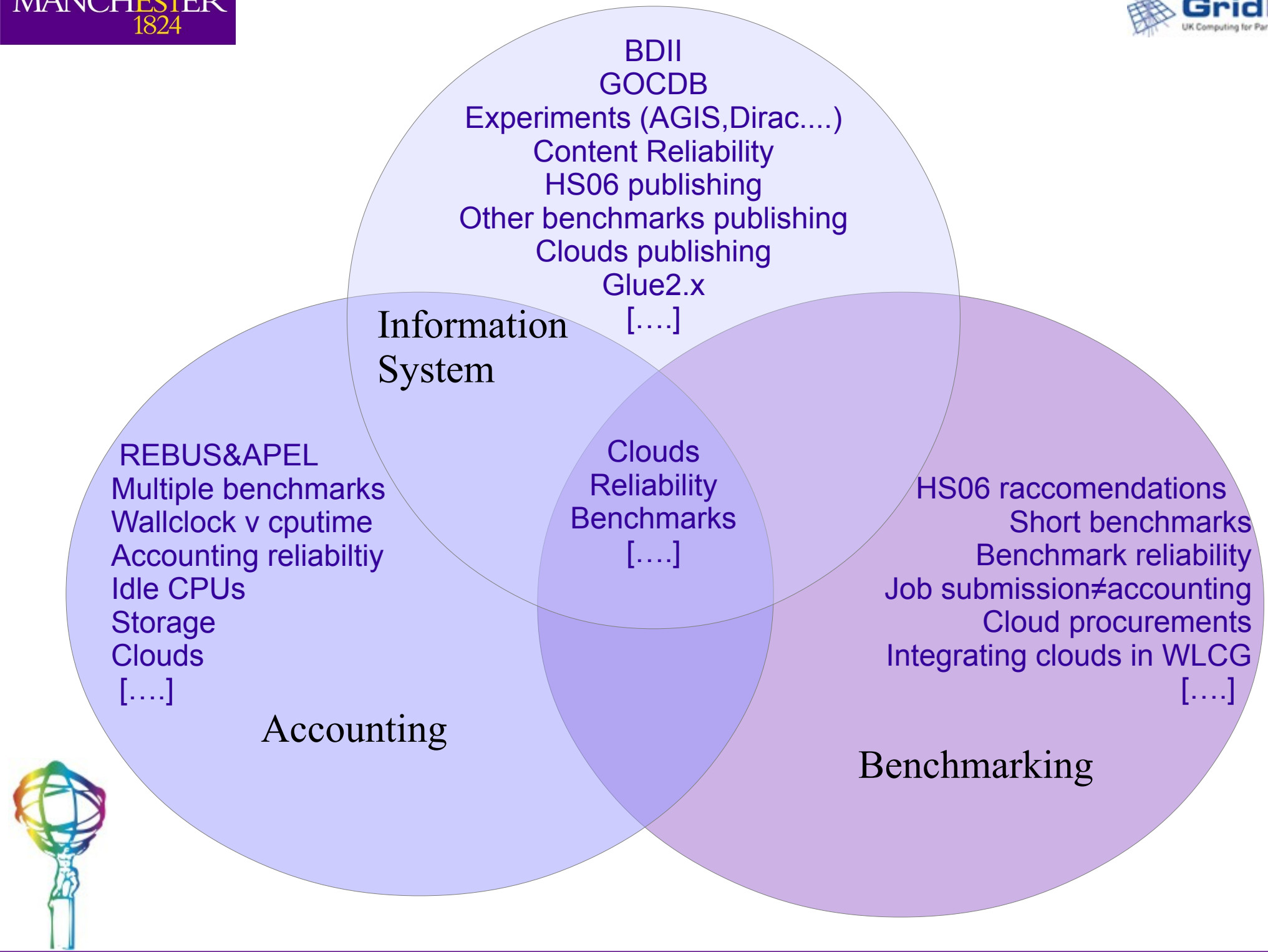
Alessandra Forti

Jamboree

20 January 2017

Some history

- Accounting long standing problem for WLCG and experiments
  - Sites often misconfigured
  - System complicated and inflexible
  - Sites and experiments measuring different things
- APEL is WLCG official accounting
  - HS06 is the benchmark it is based on
- Experiments put their numbers in the reports
- Accounting TF was created to follow up


BDII
GOCDB
Experiments (AGIS,Dirac....)
Content Reliability
HS06 publishing
Other benchmarks publishing
Clouds publishing
Glue2.x
[….]
Information System
REBUS&APEL
Multiple benchmarks
Wallclock v cputime
Accounting reliabiltiy
Idle CPUs
Storage
Clouds
[….]
Accounting
Clouds
Reliability
Benchmarks
[….]
HS06 raccomendations
Short benchmarks
Benchmark reliability
Job submission≠accounting
Cloud procurements
Integrating clouds in WLCG
[….]
Benchmarking

Decisions

- **Move WLCG to use wallclock as a metric in the reports**
- **Coherent naming and definition of fields**
- **Systematic review of discrepancies with experiments accounting for all the experiments**
- Push to move: Old → New APEL portal
  - WLCG view reflecting sites and experiments requirements
  - Redesign of monthly accounting reports for both T1&T2

Using wall clock

- Sites've asked for wall time for many years
  - CPU time legacy of when sites efficiency was reduced by bad IO.
    - Jobs hanging waiting for input or trying to storing output
  - Storage and network got better
    - Waste now mostly on experiments side
- WLCG introduced “corrected CPU time” to account for the inefficiency to avoid sites losing out on funding
  - This year experiments and sites agreed to move to walltime
- What wall clock? Who measures what?

What wall clock

- Raw wall clock:
  - Job end_time-start_time=job raw wallclock
  - Experiments measure the payload raw wall clock
- Scaled wall clock: sites use scaling factors for cpu and wall clock to compensate for the different power of the machines.
  - Batch system jobs wall time incorporates this factor
  - APEL “Elapsed time” is the scaled batch system wall time

Wall clock Problems

- Scaled wall clock for different sites not comparable
  - Different scaling factors and not all sites scale
  - Many sites “elapsed time” APEL view meaningless
- Experiments payload raw wall clock v sites pilots scaled wall clock
  - For multi payload pilots discrepancy is NOT negligible
  - Scaled wall clock ≠ raw wall clock
    - APEL “elapsed time” not comparable with experiments measures
  - Scaling factor allegedly in the BDII
    - Not all sites report correctly
    - Some sites have also inconsistent scaling on their nodes

Wall clock problems

- Experiments want also raw wall clock reported in APEL
  - Requires to change the parsers, APEL schema, summaries sent to portal → **need to understand how much effort**
- Does it include processors?
  - APEL "Elapsed" doesn't since the introduction of multicore which has affected efficiency numbers >100%
    - Need to introduce a new metric including the processors last year. Painful getting all sites updated.
  - "Elapsed" with no processors deemed "confusing"
    - On top of not being comparable because "scaled"
    - Will be dropped by the WLCG view

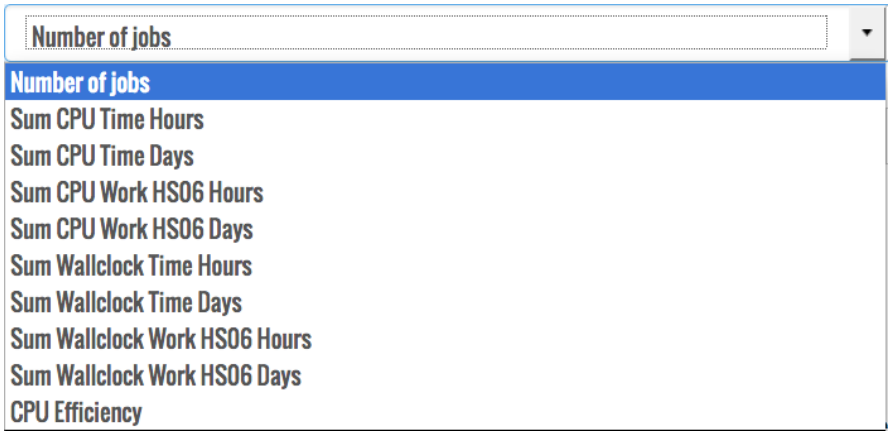
Is it really “normalised”?

- Benchmark value is a measure of power
  - HS06 = power/processor (in other contexts core/logical cpu/slot)
    - Total/Pledged HS06 = Total/Pledged power
  - Power*time is a measure of work
  - What we call “Normalized CPU time” is really CPU work
    - Power*CPU time
  - Though not really work for symmetry “Normalised Elapsed time” becomes Wall clock work
    - Power*Wall clock time
      - Wall clock time: scaled in apel, raw in exp dashboards
  - Wall clock work/time interval = Delivered power
    - In experiments reports, dashboards now also in WLCG reports
      - To be compared with pledged power

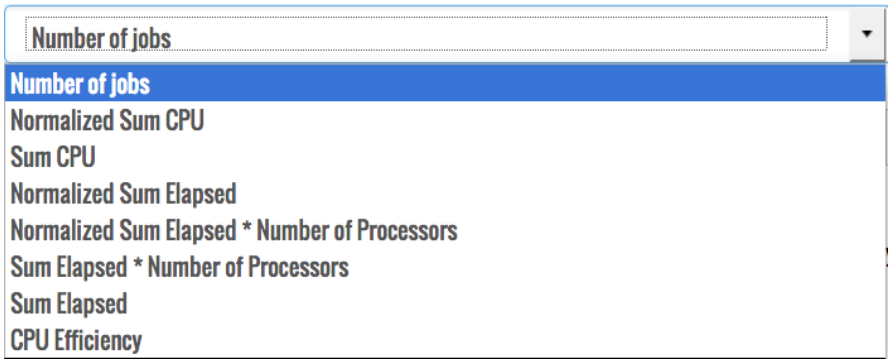
New names

- SUM CPU → SUM CPU time
- SUM Normalized CPU → CPU work
- SUM Elapsed → Dropped
- SUM Normalised Elapsed → Dropped
- SUM Elapsed*processors → SUM wall clock time
- SUM Normalised Elapsed*processors → SUM wall clock work
- Total/Pledged HS06 → Total/Pledged Power
- Wall clock work/time interval → Delivered Power

WLCG view



EGI view



ATLAS v APEL

- In ATLAS dashboard
  - Raw wall clock (seconds)
  - Average Power (HS06)
    - Calculated from REBUS: Total power/#logical CPUs
  - Delivered power: Raw Wall clock*Average power/time interval
- In SSB to compare to APEL
  - APEL wall clock work
    - Scaling is compensated when multiplied by power
  - ATLAS delivered power*time interval hours
  - If all bits and pieces are accounted in the BDII
    - ATLAS wall clock work ~ APEL wall clock work

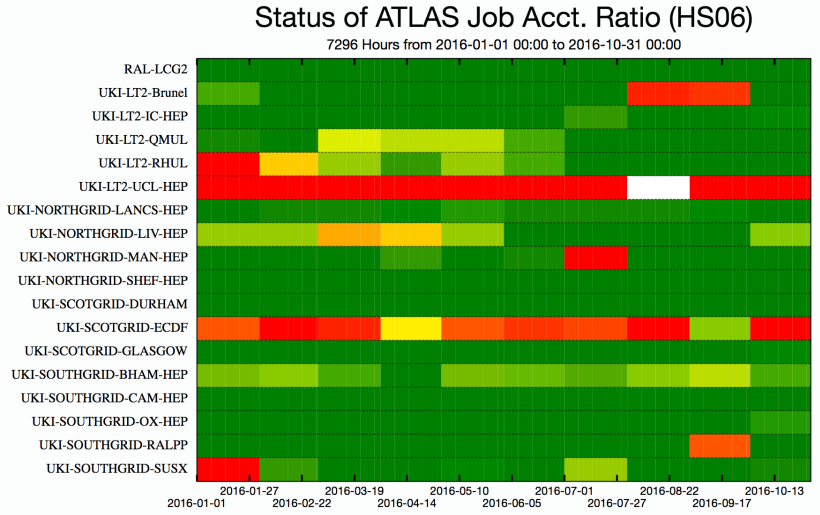
Why things may go wrong

- System is too complicated
  - Number of services involved is big
    - ATLAS: panda, dashboard summaries, REBUS, BDII
    - APEL: batch systems, CEs, BDII, GOCDB, parsers or SSMsend (or other methods), APEL, summaries for portal
  - Anything wrong in one of these places and the comparison if off
  - Sometimes problems maybe hidden and appear or disappear depending on the running resources in that period
- Can we simplify?
  - We hope to remove at least the BDII in the future but it still remains complicated.
  - Dashboard → kibana can be occasion to review the experiment side

Problems

- ARC/HTcondor parser: when flocking is active the current version doesn't see all the jobs anymore → under reporting in APEL
- Wrong scaling on a big portion of resources → over reporting in APEL
- Wrong HS06 in the BDII → wrong reporting in APEL maybe ok in ATLAS if averages out
- Resources not reported in the BDII → work done disappears from ATLAS
  - VAC sites affected if they don't adjust the BDII to include them
- Wrong DN/missing service in GOCDB → under reporting in APEL
- Site capacity misreported in REBUS → ATLAS is wrong
- New workflows in ATLAS → ATLAS under reporting
- APEL clients stops publishing → under reporting in APEL
- ……..

How to check



- SSB latest results
  - http://tinyurl.com/hevnfz5
- SSB metric history
  - http://tinyurl.com/pq72raq
  - Updated monthly

- Accounting FAQ
  - http://tinyurl.com/zn9lhhe
- Script to do more frequent comparisons
  - http://tinyurl.com/hsxrzef

```
aforti@vm7>time python get-atlas-accounting-data.py -s UKI-NORTHGRID-MAN-HEP -m6
Date,ATLAS work,EGI work,(wE-wA)*100/wA,ATLAS wc,EGI wc,(wcE-wcA)*100/wcA
2016-07,11206488,20367844,81.8%,1136975,2050129,80.3%
2016-08,17677190,17678383,0.0%,1793472,1798241,0.3%
2016-09,20989582,21853048,4.1%,2129536,2182444,2.5%
2016-10,18245809,19316185,5.9%,1851162,1947290,5.2%
2016-11,22827795,22321641,-2.2%,2316036,2261186,-2.4%
2016-12,24884445,23447118,-5.8%,2524697,2369523,-6.1%
2017-01,12144308,10142761,-16.5%,1232123,1036432,-15.9%
```

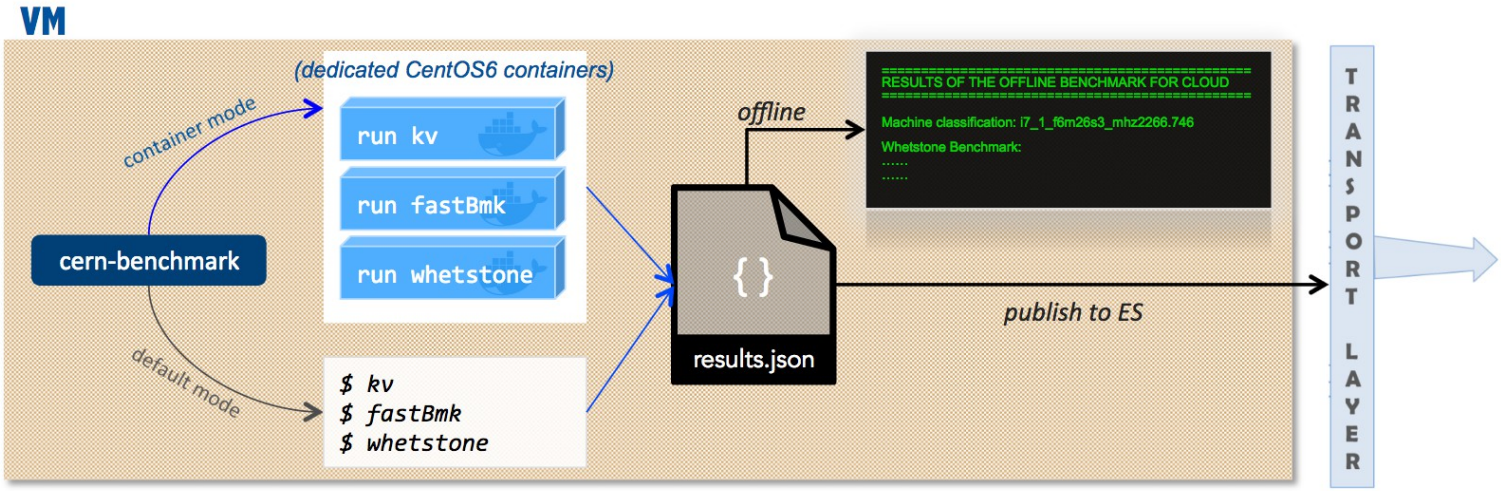
Fast Benchmarking

Why a fast benchmark

- Some applications don't scale well with HS06 anymore
  - HS06 still 32bit application
  - Compilation flags
  - Enhanced hardware features of new processors
- Some resources cannot be benchmarked
  - Commercial clouds
  - HPC
- Payloads brokering
- Fast Benchmarks can run at the start of any pilot or any VM
  - They last from 30s – 4mins

CERN benchmark suite

- The CERN-benchmark suite is available in the in GitLab
- The benchmark suite provides the ability to run one or more benchmarks with the option to publish (or not) the final results to ES (Elastic Search) at CERN
  - Sites are encouraged to participate
- The currently available benchmarks are: ATLAS KV, LHCb Fast Benchmark and Whetstone
  - Different benchmarks show different correlations with HS06 and applications



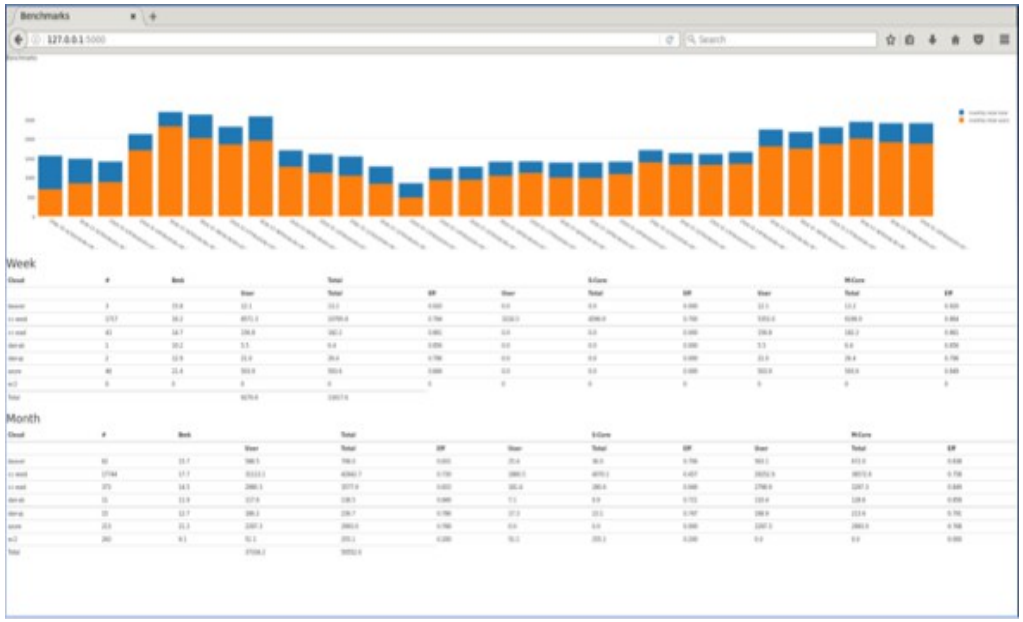
Hepix WG

- Most current work carried out by the Hepix WG
  - LHC experiments and sites
  - If you are interested in the details indico category
    - https://indico.cern.ch/category/1806/
- Next pre-GDB in February will be dedicated to Benchmarking
  - https://indico.cern.ch/event/578967/

Ongoing work in ATLAS

- R. Sobie carrying out tests using Canadian IAAS resources.
  - 2940 cores (80% private resources, 20% azure)
  - EC2 to be added but may cause problem with eviction
- Benchmark run when the VMs boot
  - VM: CernVM, 8 cores
  - Benchmark: atlas version from CVMFS
  - Tests with fastBmk and whetstone
- Goal to measure cpu benchmark and VM lifetime
  - VM lifetime measured logging /proc/stat every 15 minutes on a server. Log parsed on the server. For now.

Some results

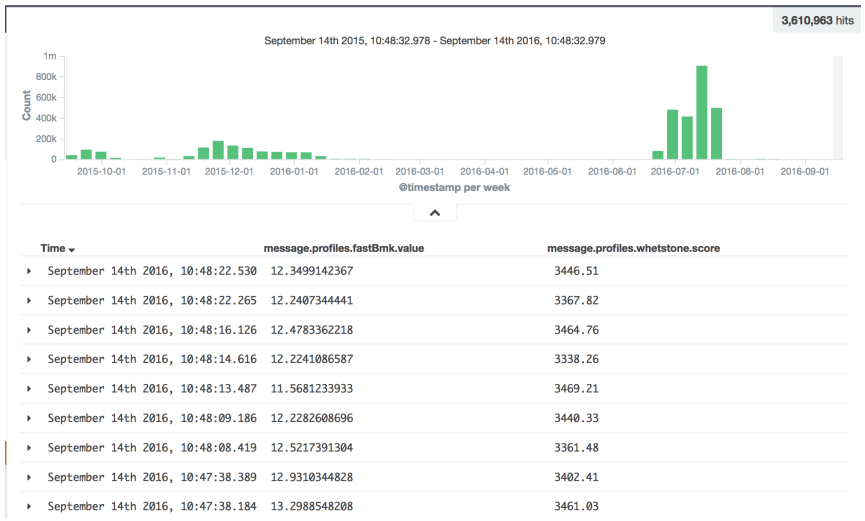


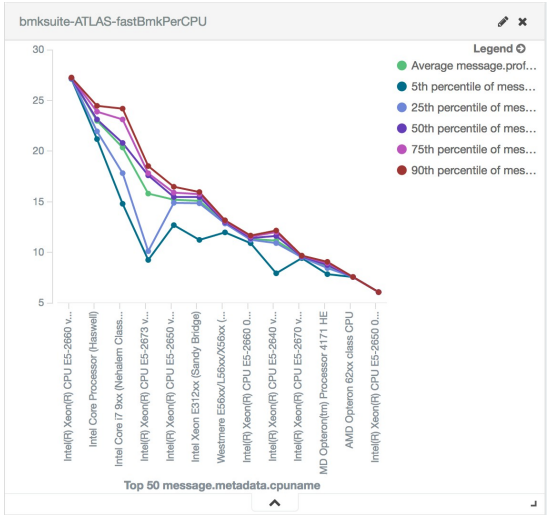
- Plot of user (orange) and total (blue) CPU-hours per day
- Weekly and monthly summaries provided below

- Cloud benchmarks and accounting (UVictoria)
- Measure Fast-HS benchmark at the VM boot-time
- Operational since mid-November
- Currently improving information display and adding plots/histograms

Elastic Search cluster

- The long term plan is to use the common ES hosted at CERN for all the resources
  - Results transported via message broker and stored in ES
  - Currently used in the cloud tests
  - Pilot needs certificate access to be enabled
    - Should work by next week





Future work

- Continue with the cloud work
  - Adding metrics such as walltime*benchmark
  - Analyse different workloads (mcore, himem, score)
- Enable the benchmarks in the standard pilot
- Publish results in ES

Conclusions

- Accounting is rationalised and cleaned up
- Most of all experiments and sites are agreeing on a common language
- It is important to investigate why there are discrepancies between experiments accounting and APEL
  - Even if it is a lighter shade of green
- It is important to be able to easily compare the resources accounting and the APEL one.
  - Though it might not mean everything is ok
- Fast benchmarking work ongoing