Multicore Accounting status

Alessandra Forti
On behalf of the multicore TF
WLCG MB
18 November 2014

Layout

- Main problem
- APEL
- Ways of publishing
- Accounting portal status
- Summary
- Backup

Main problem

- Batch systems report (correctly)
  - CPU time as the sum of the CPU used by each thread in the job
  - Wallclock as elapsed time according to the clock on the wall
- Efficiency of a job in the accounting is calculated as CPU time/Wallclock
  - Clearly resulting in >100%
- Need to introduce the cpucount in the accounting and define a new quantity wallclock*cpucount
  - David Groep put it all down nicely see backup slides

APEL status

- EMI3 APEL client is ready to do multicore accounting.
  - The schema of the APEL DB has all the necessary fields
  - Still work to do on the DB migration due to schema changes
- Sites have to publish this fields in particular the cpucount is not published by default
- Several ways to publish depending on what the site is using
  - Most had to be configured to insert the cpucount not all sites are ready

Way of publishing
(John Gordon)

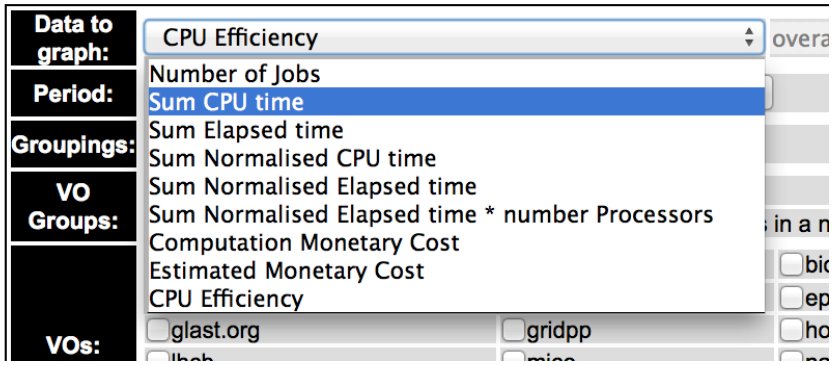
- The APEL client parsers gathers data on number of cpus and cores from the batch systems (except GE) provided an option is switched on.
  - This option is off by default so multicore sites need reminding to turn it on. If they want to backdate their publishing they will need to re parse their batch logs.
- ARC CE. NDGF sites publish via SGAS are publishing correctly.
- Other ARC CEs use JURA which publishes direct to APEL from each CE so there is no site database. Ncores and ncpus are published.
- OSG planning about the change to SSM2. They hope to have it done by Xmas.
- Italian sites have all (?) migrated from DGAS to use the standard APEL client so they (can) now publish cores. In the dev portalthere are only Frascati, Roma1 and Roma3 publishing though.

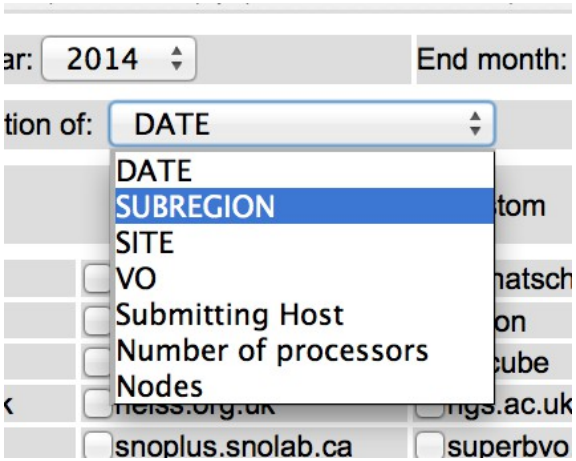
Accounting Portal

- The current accounting portal doesn't report things correctly.
  - Should publish both wallclock and wallclock * cpucount
    - As intermediate step to correct the efficiency it should have published wallclock*cpucount
      - Not working yet several sites have efficiency >100%
- There is a new development portal with several more selections that is what we want in the long term.
  - Not usable yet, but promising
  - Timeline to have it in production depends on all things falling in place
  - If you want to check it
    - http://accounting-devel.egi.eu/show.php

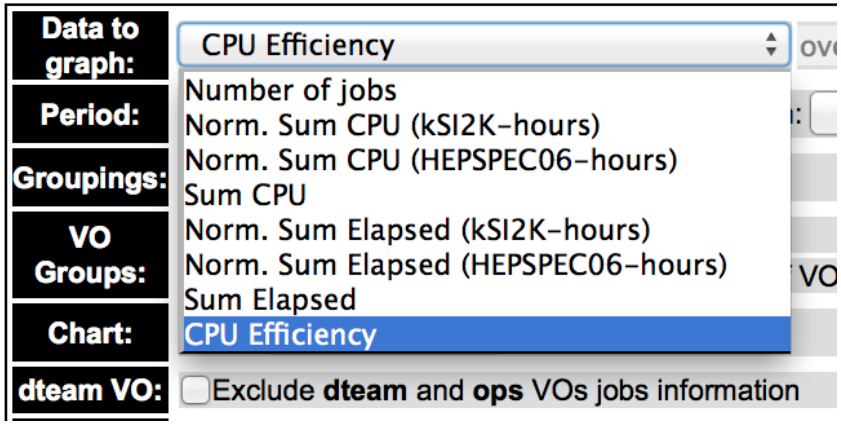
New accounting portal

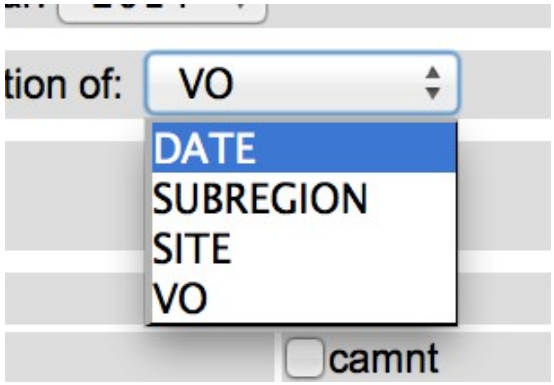
NEW





OLD





Summary

- APEL 3 can do accounting for multicore and the numbers in there are correct when sites publish the parameters
- Sites have to publish things correctly
  - EMI-3 CREAMs have to enable multicore support
    - Could be enabled by default in the next release (?)
  - Sites using SSM1.2 should move to SSM2
  - Sites using DGAS should move to use the APEL client
- Development portal promising need more work
  - and feedback

Acknowledgements

- Jeff Templon
- Manfred Alef
- Thomas Hartman
- Stefano Del Pra
- John Gordon
- Ivan Diaz Alvarez
- Rob Quick
- Stephen Jones
- Andrew Lahiff
- Dan Traynor

Backup slides

Definitions

(David Groep)

- WallDuration: the actual *wall* clock time used (so a job using 20 cores all at the same time, starting at 10:00:00Z and finishing at 10:01:00Z will report 60 seconds WallDuration
- CpuDuration: the value of cput reported by Torque, which is summed over all cores. As per GFD.98 "summed over all processed in the job". So the job above, it run with 100% efficiency and no IO latency, would report 1200 seconds
- NodeCount and CpuCount: as taken from batch system accounting data
- ServiceLevelType "Si2k": where the value of this service level is derived from HEPSPEC06 as "HS06*1000/4"
- ServiceLevel: CPUcount * (weighted average of the <ServiceLevelType> of the participating nodes)the participating nodes)
- So if the job above used 15-cores on a machine with a HEPSPEC rating of 16, and the other 5 cores ended up ona machine with a HEPSPEC rating of 12, the service level "Si2k" would be ((15*16)+(5*12)) * (1000/4), so 75000.
- In the end, WallDuration * ServiceLevel determines the 'price' of the system. Efficiency is harder, but in case all participating nodes in a job have the same performance level, it would be (CpuDuration/CpuCount)/WallDuration.
- In case of mixed nodes contributing to the job, the true efficiency cannot be
- reconstructed from APEL data (nor even from Torque accounting logs)