
CERN

Benchmarking and accounting for the (private) cloud

Jerome Belleman, Daniel Pek, Ulrich Schwickerath,

Special thanks to the CERN Cloud team

Outline

- CERNs batch farm

Outline

- CERNs batch farm
- Schema to classify worker nodes by performance

Outline

- CERNs batch farm
- Schema to classify worker nodes by performance
- Benchmarking

Outline

- CERNs batch farm
- Schema to classify worker nodes by performance
- Benchmarking
- Accounting

Outline

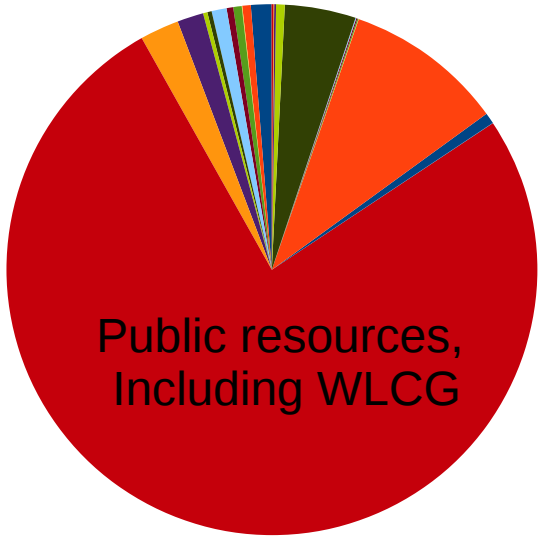
- CERNs batch farm
- Schema to classify worker nodes by performance
- Benchmarking
- Accounting
  - Traditional batch accounting

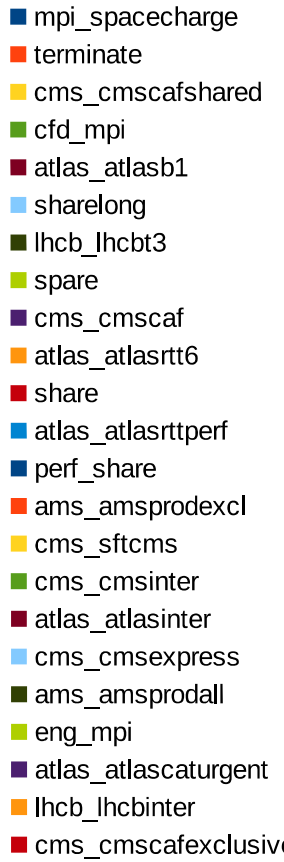
Outline

- CERNs batch farm
- Schema to classify worker nodes by performance
- Benchmarking
- Accounting
  - Traditional batch accounting
  - Cloud accounting

CERNs LSF batch farm

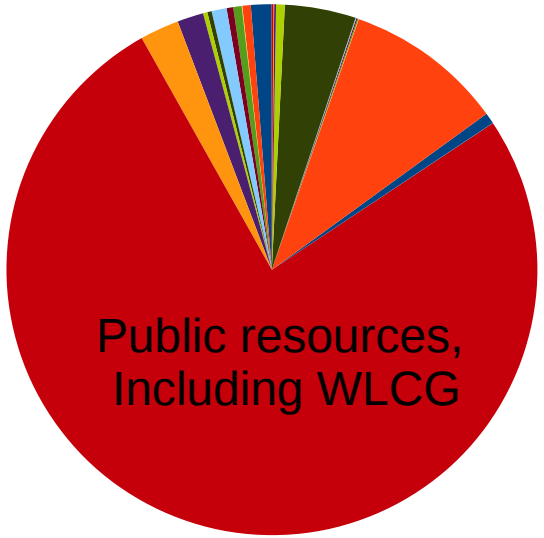
- CERN LSF batch farm:
  - About 4300 nodes in total, ~3700 VMs
  - About 3600 in public resources
    - Got rid of old physical worker nodes
    - 93% on virtual machines now
  - Traditional GRID worker nodes
  - Traditional APEL based accounting (HS06)
- In addition dedicated IaaS projects for the experiments





CERNs LSF batch farm

- CERN LSF batch farm:
  - About 4300 nodes in total, ~3700 VMs
  - About 3600 in public resources
    - Got rid of old physical worker nodes
    - 93% on virtual machines now
  - Traditional GRID worker nodes
  - Traditional APEL based accounting (HS06)
- In addition dedicated IaaS projects for experiments

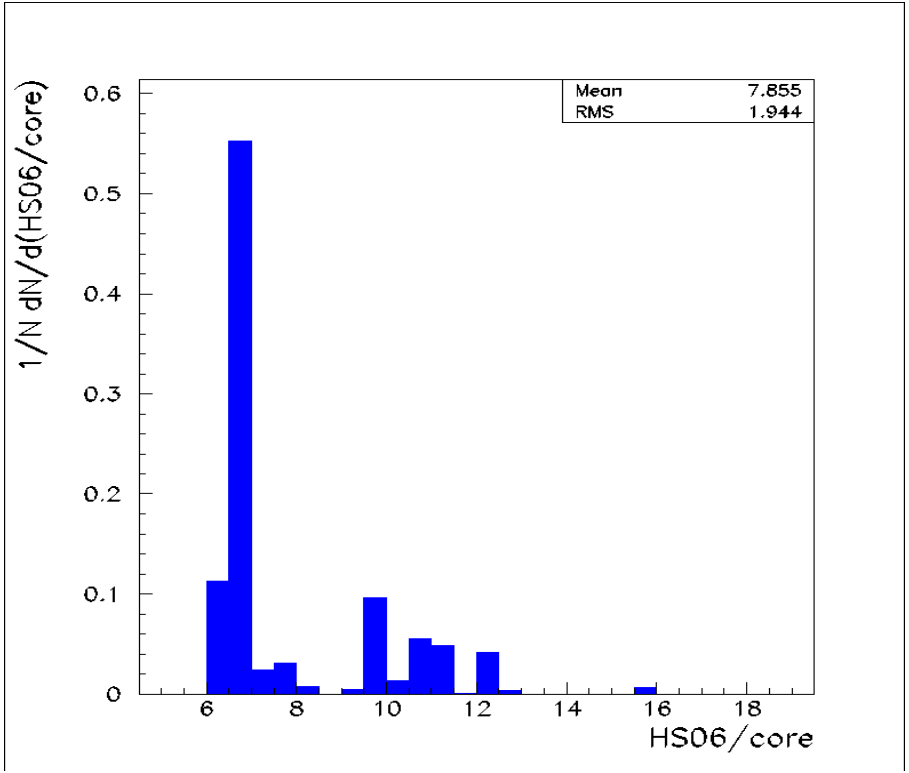


- mpi_spacecharge
- terminate
- cms_cmscafshared
- cfd_mpi
- atlas_atlasb1
- sharelong
- lhcb_lhcbt3
- spare
- cms_cmscaf
- atlas_atlasrtt6
- share
- atlas_atlasrttperf
- perf_share
- ams_amsprodexcl
- cms_sftcms
- cms_cmsinter
- atlas_atlasinter
- cms_cmsexpress
- ams_amsprodall
- eng_mpi
- atlas_atlascaturgent
- lhcb_lhcbinter
- cms_cmscafexclusive

CERNs LSF batch farm

Heterogenious hardware

- Complexity partly hidden by virtualization
- Hypervisor and its performance is hidden
- Still large spread of per core performance



Classification of worker nodes

- Bare metal times
  - Procurement of chunks of identical machines
  - Classify by procurement (vendor, procurement time, sub-class ...)
  - Benchmark one or few sample machines
- Virtual worker nodes
  - No information available about the hypervisor
  - VMs can change name
  - Benchmarking each of them every time is expensive
  - **Need a new way to classify machines by performance**

Classification of worker nodes

Example: a6_8_1512h23_266
AMD based virtual machine

- SLC6
- 8 cores
- CPU-ID 1512h, see below
- CPU speed 2300 MHz
- Default memory speed 266

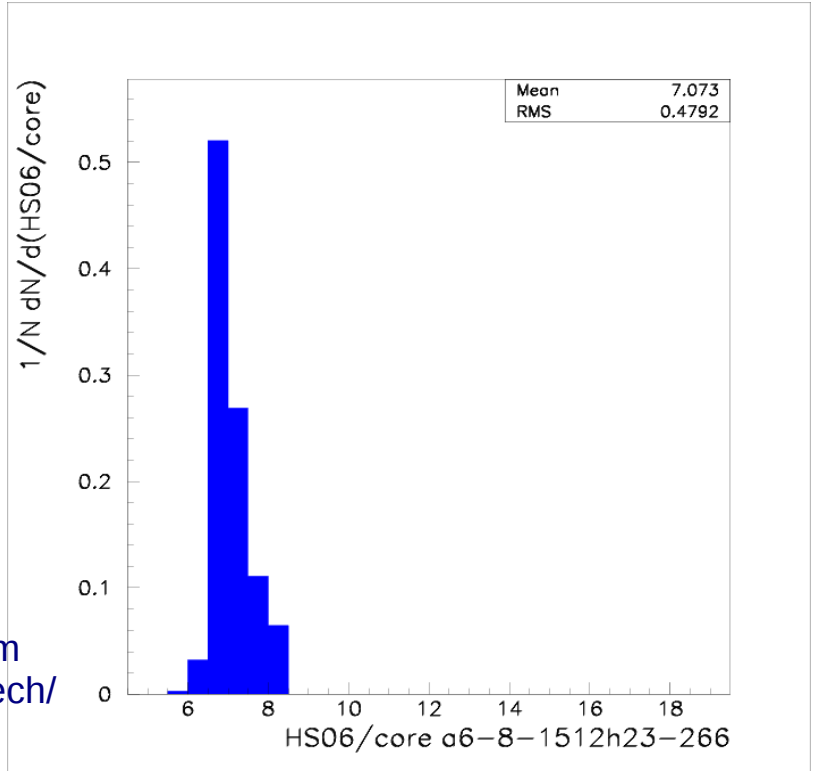
Remark : Details of the machine:

Memspeed = unknown
Cpuspeed = 2.3MHz
Cpuvendor = AMD
Cores = 8

CPUfamily = 21 =0x15h
CPUmodel = 1 = 0x1h
CPUstepping = 2 =0x2h

=> **CPUID = 1512h**

http://world.std.com/~swmcd/steven/tech/cpu.html



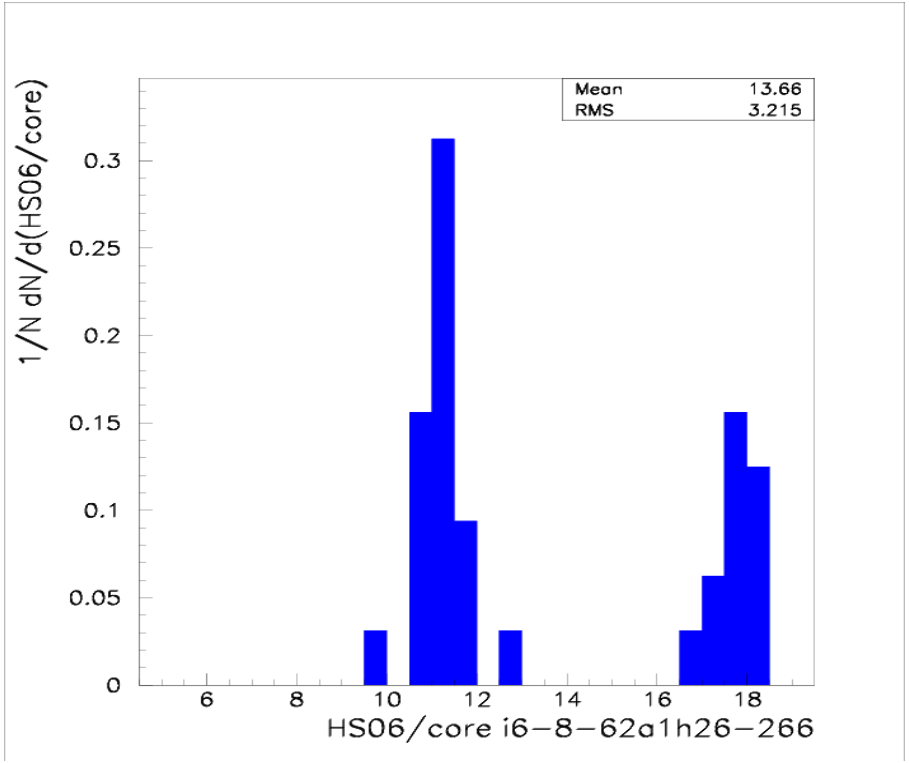
Benchmarking by class

- Pre-requisites:
  - Enable CPU pass-through (else different classes are mapped to the same class)
  - Don't over-commit CPU resources
  - Tune KVM for best CPU performance
- Benchmark
  - Ensure the hypervisors are fully loaded
    - Easy for new batches of hardware coming in
    - Benchmark each VM to get statistics
  - Be pessimistic when interpreting the results

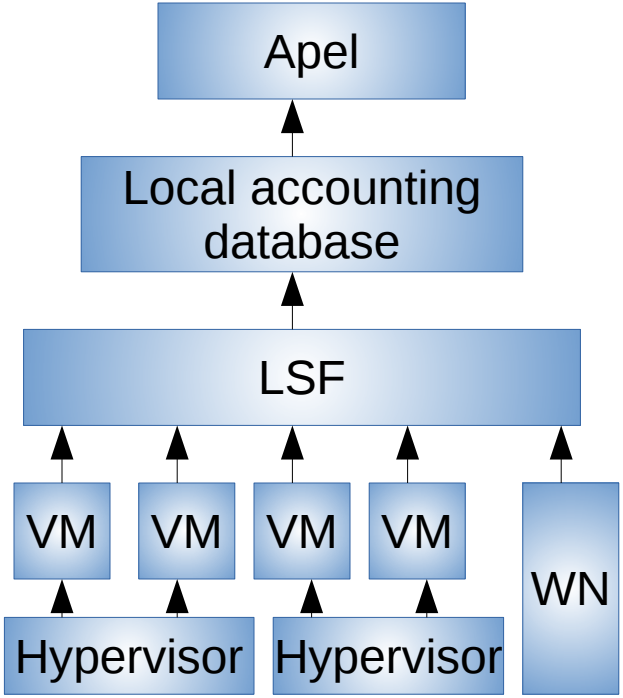
Benchmarking by class

Limitations:

- The memory speed is not passed to the VM by KVM
- A conservative default of 266MHz is assumed
- Different memory speeds yield to a double-peak structure

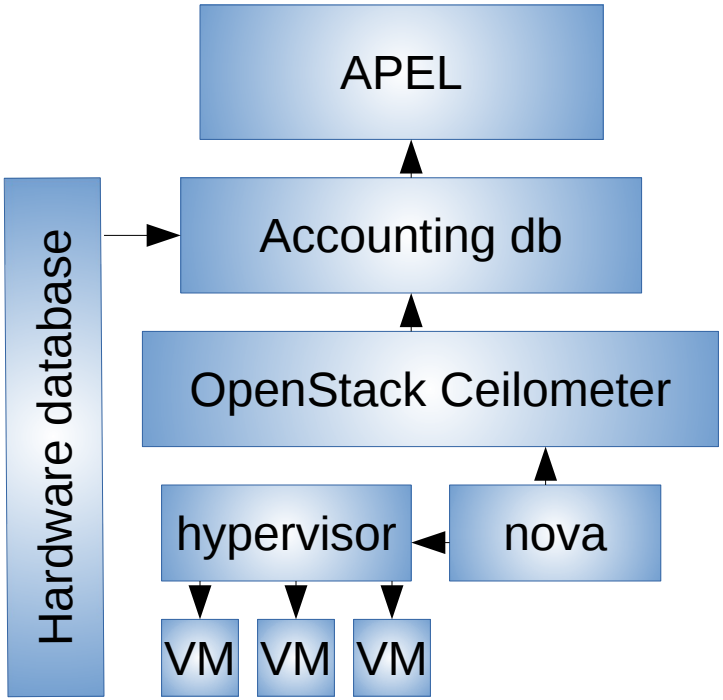


Traditional batch accounting



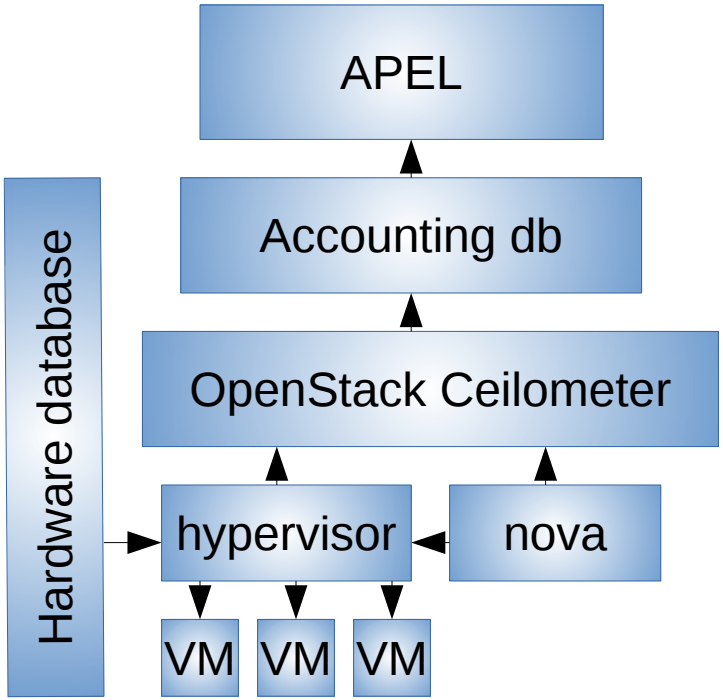
- WLCG - Accounting via APEL and SSM
- Local database holding accounting data
- LSF job_finish records are sent to the local database
- Virtual and physical worker nodes look the same

Cloud accounting: general case



- Still experimental!
- No access to the VMs by the site
- Classify by performance of the hypervisor for now
- Loss of information for short lived VMs (loss of link to hypervisor after the VM is gone)

Cloud accounting: general case



- Work in progress:
  - Inject performance info from the hypervisor to ceilometer while the VM is running
- Possible future work:
  - inject all information we need to do the classification as for the batch case
  - Unclear how to do this in a general case

Conclusions

- Established a new classification schema for batch worker nodes
  - Using only information available from the machine itself
  - Works reasonably well
  - Used in production both for physical and virtual worker nodes
- Extension to the general case
  - Non-trivial because it's not the VMs which report in this case
  - Requires additional configuration in OpenStack
  - Work in progress

www.cern.ch