The INFN COSA project: Benchmarking

Michele Michelotto

Benchmarking

Main activities of HEPIX Benchmarking Group
Status Report of Manfred Alef at HEPIX LBL meeting

→ **Fast benchmark**
to estimate the performance of the provided job slot (in traditional batch farms) or VM instance (in cloud environments)

- Job matching (e.g. “can a pilot run another payload with the resources left?”)
- Accounting if HS06 score is not available

→ **Next generation of long-running benchmark**
for installed capacities, accounting, procurements aso. in WLCG (successor of HS06)

Benchmarking

Organization:

- **Mailing list (hepix-cpu-benchmark@hepix.org):**
  - 49 subscribers
- **Meetings:**
  - Kick-off at HEPiX Zeuthen
  - 5 Vidyo meetings so far (biweekly)
  - 6 … 16 attendees per meeting

Benchmarking

Fast benchmarks:

→ **Started with 5 candidates:**

- 3 benchmarks relevant to HEP workload:
  - DIRAC Benchmark 2012 (Python script) [1] (DB12 – as yet named 'fastBmk', 'LHCbMarks', …)
  - Atlas Kit Validation (KV) [2] (default workload: Geant4 single muon event generation)
  - ROOT stress test [3]
- 2 benchmarks widely used in common workload management tools (HTCondor, Boinc, ...):
  - Whetstone, Dhrystone [4]

Benchmarking

Fast benchmarks:

→ **Tools:**

- New release of the CERN Cloud Benchmarking Suite (Domenico Giordano, Cristovao Cordeiro) supports not only cloud but also batch environments [5]
- Now it provides a "hyper-benchmark" which reports:
  - 1-min load of the system under test
  - HS06 score from MJF store (if available)
  - DB12
  - Whetstone
- Can run KV as well (if available in CVMFS)

Benchmarking

Fast benchmarks:

- Indepentent investigations by Domenico Giordano (CERN) [5] and Manfred Alef (KIT) [6] have demonstrated good correlation between DB12 and KV when executed in batch jobs as well as in clouds
  - Work done by Domenico Giordano shows an outlier when running in VM with many cores – caused by much shorter runtime?
- LHCb and Alice have demonstrated good scaling of their applications with DB12 [7,8]
- Latest investigations by Costin Grigoras (Alice) have demonstrated better scaling of Alice software with DB12 than ROOT stress [8]

Benchmarking

Fast benchmarks:

→ Differences:

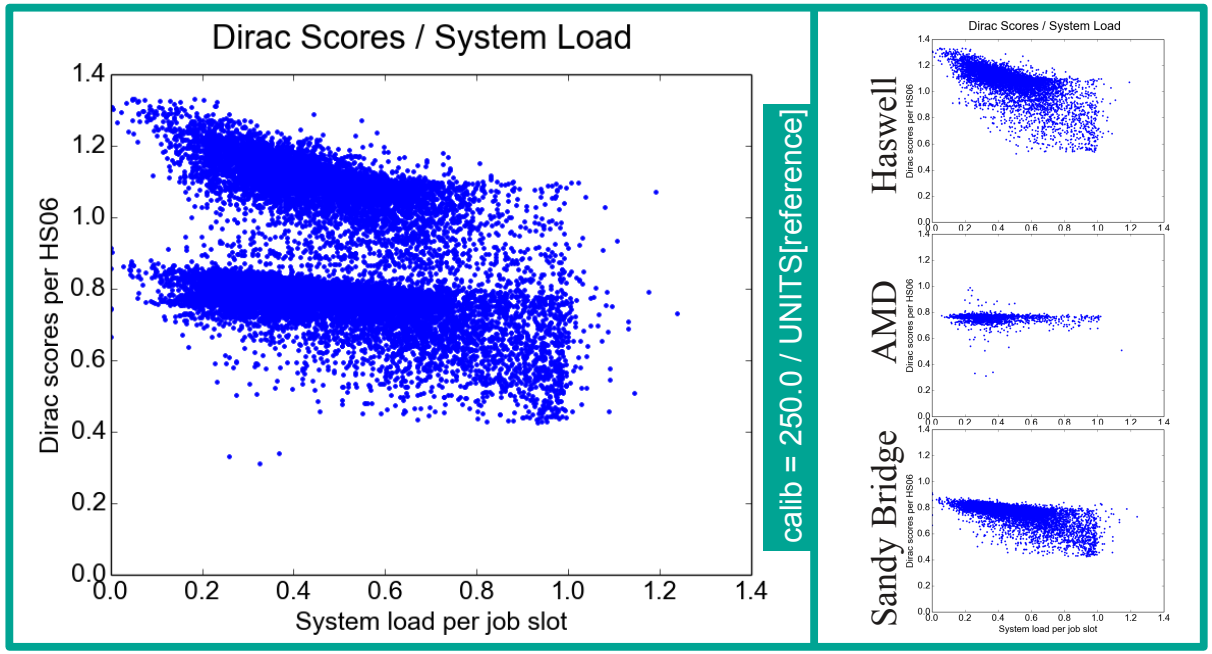
- Runtime:
  - DB12 (< 1 minute) is much faster than KV (~ 5 minutes)
- Licensing:
  - DB12 is open source
  - Geant4 (default workload of KV) is open source
  - Athena (wrapper used by KV) is closed source

→ **DB12 seems to be the most suitable candidate in the first group of fast benchmarks (related to WLCG workloads)**

Benchmarking

Fast benchmarks and HS06 score:

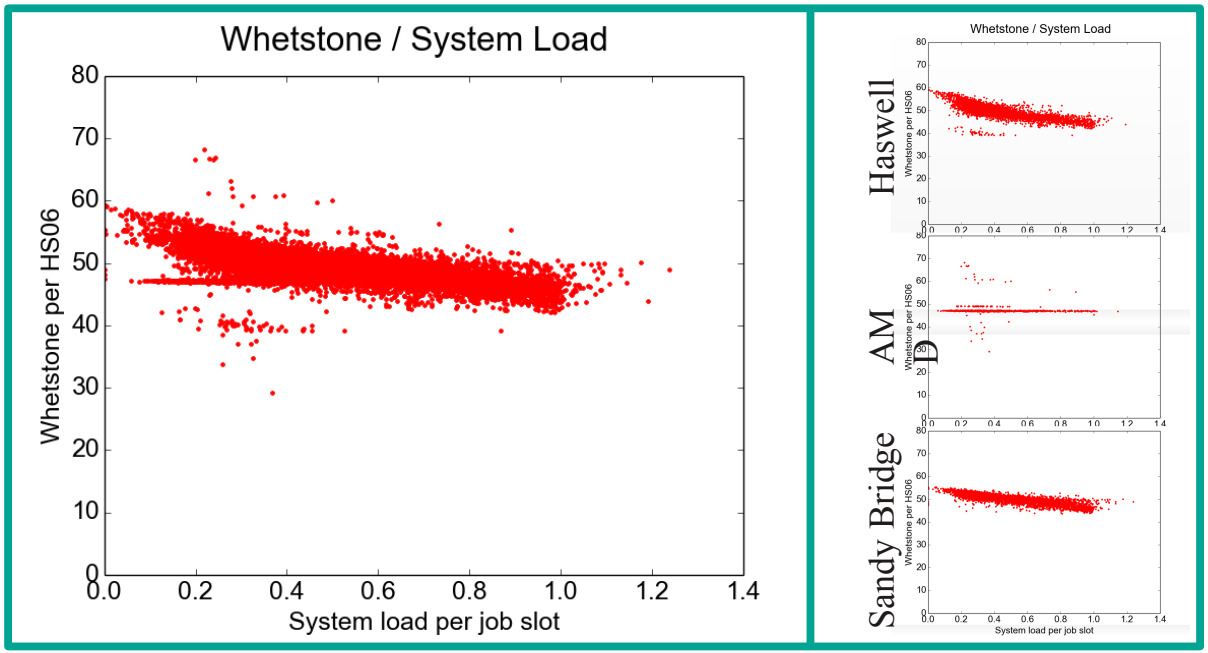
- ➜ At the present time HS06 is still the official metric for installed capacities (pledges, accounting, …)
- ➜ Unfortunately DB12, KV, and ROOT don't scale well with HS06 [6]



Benchmarking

Fast benchmarks and HS06 score:

- ➔ Are we happy with DB12 also for predicting HS06 score (e.g. in anonymouns environments like clouds)?
- ➔ Whetstone might be a better choice to estimate HS06 [6]



Benchmarking

Fast benchmarks and HS06 score:

- Scaling of Whetstone with HS06 has been demonstrated so far only by Manfred Alef as the results of single-core batch jobs at GridKa [6]
  - There are currently no other results known, for instance from multi-core or cloud VMs
- Dhrystone scales similar to ROOT stress and is therefore not of the first choice

Benchmarking

An important open question is why DB12 and KV are scaling well with WLCG applications, contrary to HS06

→

- Where is the magic boost of DB12, KV, and WLCG applications coming from e.g. when running on latest chip generations

- Haswell?

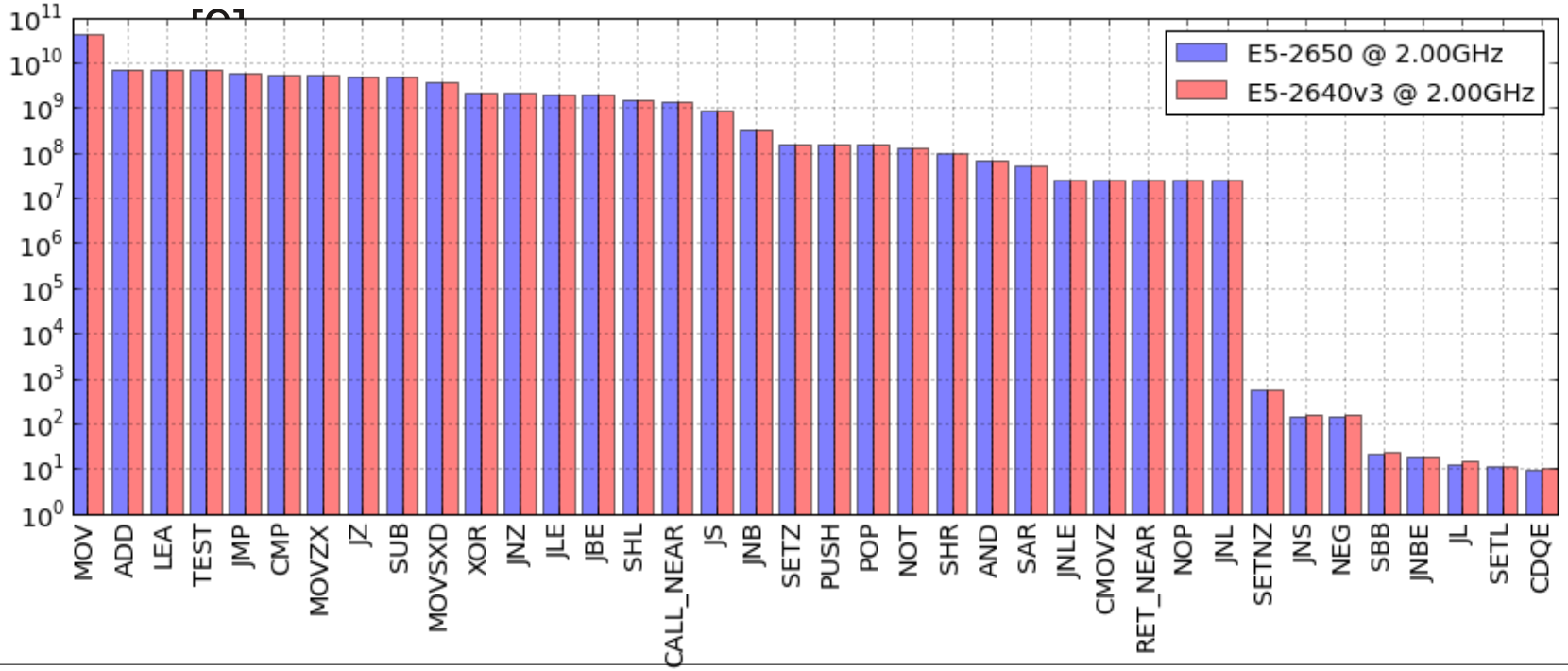
→

- Investigations by Marco Guerri [9] have shown that the hot-spot in DB12 is a huge switch statement which benefits from the improved Branch Prediction Unit of modern processors

Benchmarking

Evolution of processors:

→ **No general speedup in typical instructions**
between Sandy Bridge (blue bars) and Haswell hosts (red bars) running at the same clock speed, e.g. PyEval_EvalFrameEx



Benchmarking

Evolution of processors:

→ Boost of up to 45% in DB12 is caused by **improvements in a single type of instructions** (Branch Prediction Unit) [9]

Benchmarking

Evolution of processors:

- ➔ Until ~10 years ago:
  Clock speed racing, small steps of enhancements
- ➔ Multiple cores Inflating cache size
- ➔ Peripheral, special purpose hardware extensions, for instance:
- ➔
  - Vector engines Graphics
  - hardware Video processing
  - Encryption
  - Random number generation
  - Branch prediction
  - ...
  -

Benchmarking

Evolution of processors:

- The hot-spot in the CPU consumptions of the WLCG benchmarks should be the same as in the most relevant HEP applications

Else high risk that the new fast benchmarks will break as soon as one of the next chip generations will boost only the benchmark, or only the HEP applications

Benchmarking

Plans:

- Proposal of fast benchmark Q1/2017
- Development of long-running benchmark starting in Q2/2017

Benchmarking

References:

[1] http://diracgrid.org
[2] http://iopscience.iop.org/article/10.1088/1742-6596/219/4/042037/pdf
[3] https://root.cern.ch
[4] https://github.com/cloudharmony/unixbench
[5] https://indico.cern.ch/event/535458/contributions/2176092/attachments/1284582/1909948/ CERNCloudBenchmarkSuite_HEPiXBmkWG_giordano.pdf
[6] https://indico.cern.ch/event/394780/contributions/1832628/attachments/1238976/1820833/ Fast-benchmarks-2016-03-07.pdf
[7] https://indico.cern.ch/event/394786/contributions/2298897/attachments/1335785/2011087/ 20160914-mcnab-lhcb-benchmark.pdf
[8] https://indico.cern.ch/event/394786/contributions/2298897/attachments/1335785/2010771/ ALICE_update_on_fast_benchmarking.pdf
[9] https://mguerri.web.cern.ch/mguerri/Benchmarking/SandyBridgeVSHaswell.html