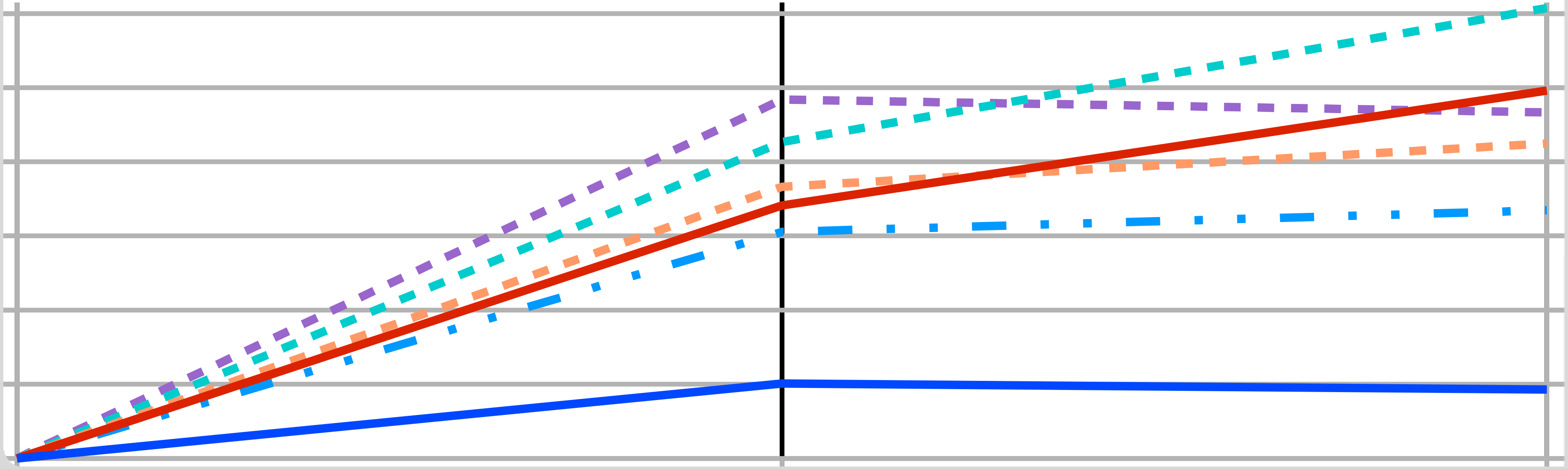
HEPiX Benchmarking Working Group
Status Report Oct 2017

Manfred Alef (KIT), Domenico Giordano (CERN), Michele Michelotto (INFN)

STEINBUCH CENTRE FOR COMPUTING (SCC)



Mandate

HEPiX

Benchmarking Working Group

- **Fast benchmark**
  - → Estimate performance of provided job slot or VM instance
- **Next generation of long-running benchmark**
  - → For installed capacities, accounting, procurements aso. (successor of HS06)

Organization

- 60 subscribers of mailing list (hepix-cpu-benchmark@hepix.org)
- Biweekly Vidyo meetings
  - Kick-off at HEPiX Zeuthen (Apr 2016)
  - ~10 attendees per meeting
    - Site admins
    - Experiment representatives (Alice, Atlas, CMS, LHCb)

Status of the Working Group

- Today:
  - Status update since last HEPiX meeting (Apr 2017), and April GDB
    - Talks by Domenico Giordano *

* https://indico.cern.ch/event/595396/contributions/2558270/attachments/1449936/2235385/HEPiX_April_2017_benchmarking_giordano.pdf
https://indico.cern.ch/event/578985/contributions/2529527/attachments/1443884/2223904/GDB_April_2017_benchmarking_giordano.pdf

* https://indico.cern.ch/event/595396/contributions/2558270/attachments/1449936/2235385/HEPiX_April_2017_benchmarking_giordano

Fast Benchmark

- DIRAC Benchmark 2012 (DB12) is an attractive fast benchmark
  - → Python script running for around 1 min
  - → Very good correlation with Alice and LHCb jobs when running 1 benchmark copy ('DB12-in-job')
    - However, DB12 doesn’t show the stability and characteristics to probe all components of the CPU potentially used by HEP workloads; e.g. the limited instruction mix doesn’t stress the memory subsystem

Next-Generation Long-Running Benchmark

Purpose of the 'long-running' benchmark is to measure installed and pledged compute capacities.

Hence it must scale (with a certain accuracy*) with the average WLCG job mix, but it will probably not scale with any individual job type (simulation, event generation, reproduction, ...)

***** Initial objective of HS06: spread ≤ 10%

Next-Generation Long-Running Benchmark

- Current HS06 benchmark built on SPEC CPU2006
- New SPEC CPU2017 has been released Jun 20
  - ➔ Volunteering sites have already purchased the new benchmark suite, and they are now warming up
- Packaging Alice and Atlas reference workloads in Docker containers *
- HS06 scaling issues have been investigated in more detail
  - ➔ 64bit temporary workaround?

* https://indico.cern.ch/event/653573/contributions/2700565/attachments/1513184/2360433/HEPiX-workload-on-docker-container.pdf

SPEC CPU2017 Benchmark Suite

- Website: www.spec.org/cpu2017/
- 43 single benchmarks
  - Integer, and floating point
  - Speed, and rate metric
  - Many benchmark names are already known from CPU2006, but CPU2017 is coming with new releases, and running improved workloads

SPEC CPU2017 Benchmark Suite

- SPECspeed and SPECrate metrics as before
  - → Now different branches within the benchmark suite
    - SPECspeed: 20 benchmarks (10 integer + 10 fp)
    - SPECrate: 23 benchmarks (10 integer + 13 fp)
  - → Memory requirements
    - SPECspeed benchmarks very memory-hungry (up to 16 GB), that’s far too much for parallel copies as in HS06
    - SPECrate requires only 2 GB RAM per copy
- Current status:
  - → Volunteering sites are warming up
  - → First results at next HEPiX

Scaling Issues of HS06 vs. HEP Applications

- 64 bit interim solution?
  - → HS06 runs with mandatory -m32 compiler flag
  - → Improved scaling with -m64?
    - Nearly linear increase by around 10...20% of 64bit benchmark scores
    - Double-checked SL6 + CentOS7
    - AMD Epyc: + ~33% (when running 1 benchmark copy per core)
  - → Conclusion: migration to 64 bit doesn’t fix the scaling issues

Scaling Issues of HS06 vs. HEP Applications

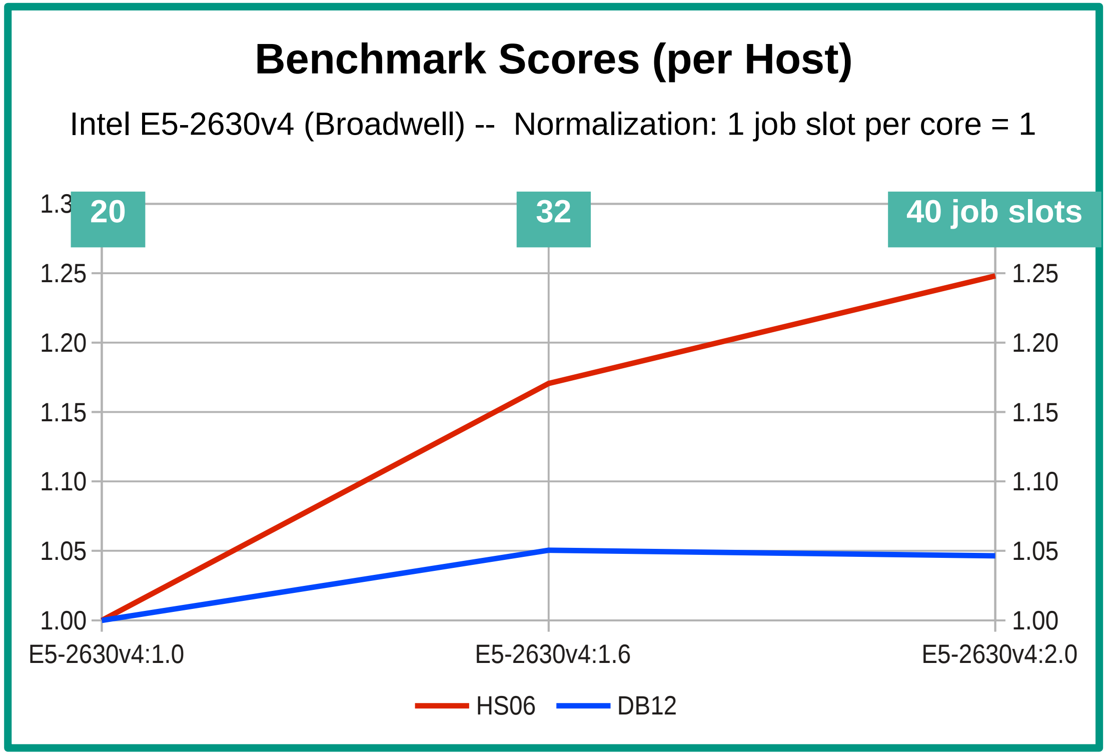
- Expanding to second dimension
  - HS06 had been developed by the HEPiX Benchmarking Working Group from 2007 to 2008
  - Typical WN hardware at that time without Hyperthreading feature:
    - Intel: quad-core CPUs Xeon E53xx or E54xx
    - AMD: 8...16-core CPUs Opteron 23xx or 61xx
  - First servers with Hyperthreading feature (Intel E55xx) appeared on the market at the end of the project
  - Variety of WN configurations at sites
    - HT disabled
    - HT enabled, more than 1 job slot per physical core
      - E.g. ~1.5, or 2 job slots per core

Scaling Issues of HS06 vs. HEP Applications

- Expanding to second dimension
  - → Experiment reports, for instance at several GDB meetings, have compared different hardware models
  - → Only few reports taking into account the individual WN configuration, especially the number of job slots
  - → Indications that this is important too

Scaling Issues of HS06 vs. HEP Applications

- Expanding to second dimension
  - ➔ Discrepancies in static benchmark scores (HS06, DB12-at-boot)



  - ➔ What about HEP applications?

Scaling Issues of HS06 vs. HEP Applications

- Expanding to second dimension
  - Deeper analysis at KIT and at PIC
    - GridKa compute farm has been reconfigured
      - Default configuration: 1.5 (or 1.6) job slots per core
      - Latest hardware model (Intel Xeon E5-2630v4, Broadwell) with 3 different configurations:
        - 1.0 job slots per core (20 slots)
        - 1.6 job slots per core (32 slots)
        - 2.0 job slots per core (1 per logical processor) (40 slots)
      - Correlations between job performance (events/s) and benchmark scores?
    - Dedicated benchmarking hosts at PIC

Scaling Issues of HS06 vs. HEP Applications

- Expanding to second dimension
  - → Deeper analysis at KIT and at PIC
    - Performance results:
      - Benchmark scores (# copies == # job slots)
        - HS06
        - DB12-at-boot (MJF package)
      - Further benchmarks compared at PIC:
        - Atlas KV
        - CMS ttbar sim.

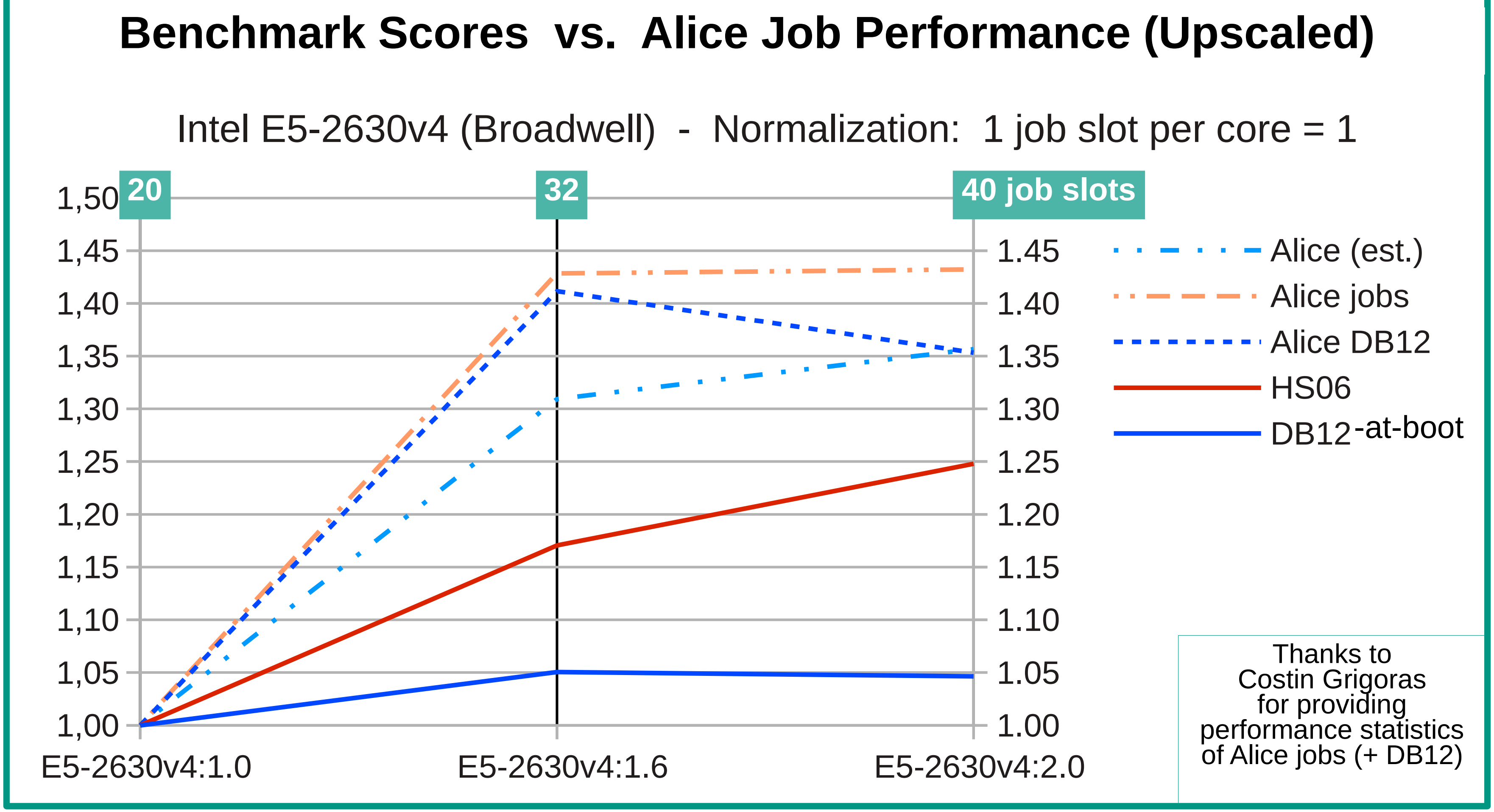
Scaling Issues of HS06 vs. HEP Applications

- Expanding to second dimension
  - ➔ Deeper analysis at KIT and at PIC
    - Performance results:
      - Performance of jobs run at GridKa (everyday job mix)
        - Alice (thanks to Costin Grigoras)
        - Atlas (values downloaded from Bigpanda, Tasks: simul=10944000, recon=11323845, evgen=11330855)
        - LHCb (thanks to Philippe Charpentier)
        - CMS: n.a.
      - Alice and LHCb have also reported corresponding DB12-in-job scores (running 1 benchmark copy)
        - LHCb: DB16-in-job which is the same Python script as DB12 but with a modified internal calibration factor

Scaling Issues of HS06 vs. HEP Applications

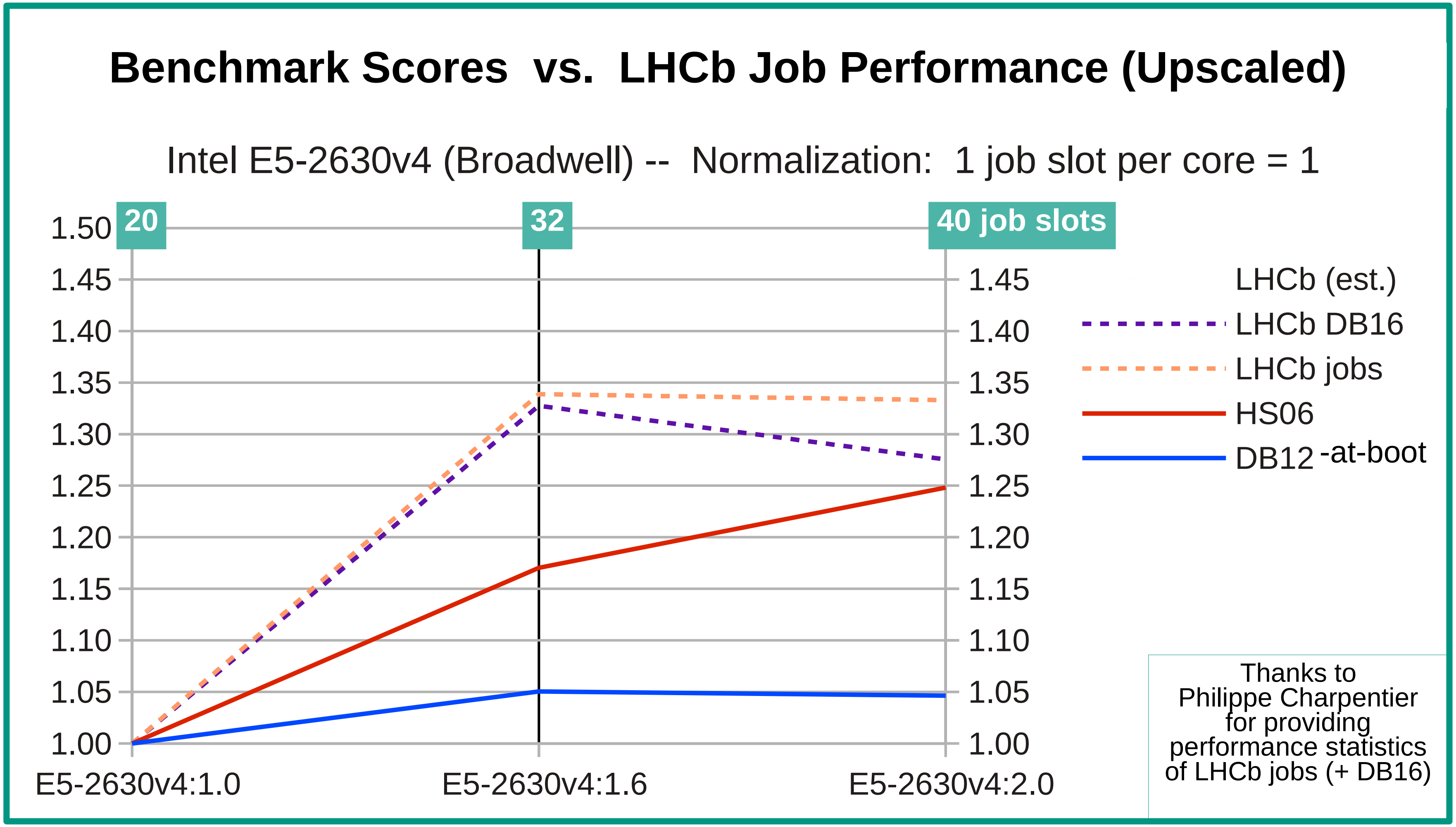
- Expanding to second dimension
  - ➔ Deeper analysis at KIT and at PIC
    - Performance results:
      - Job performance estimated by comparing runtime of top processes
        - Rough estimates, no high-precision accounting scores!
        - LHCb: n.a. (sophisticated autocalibrations)

Scaling Issues of HS06 vs. HEP Applications



Thanks to
Costin Grigoras
for providing
performance statistics
of Alice jobs (+ DB12)

Scaling Issues of HS06 vs. HEP Applications

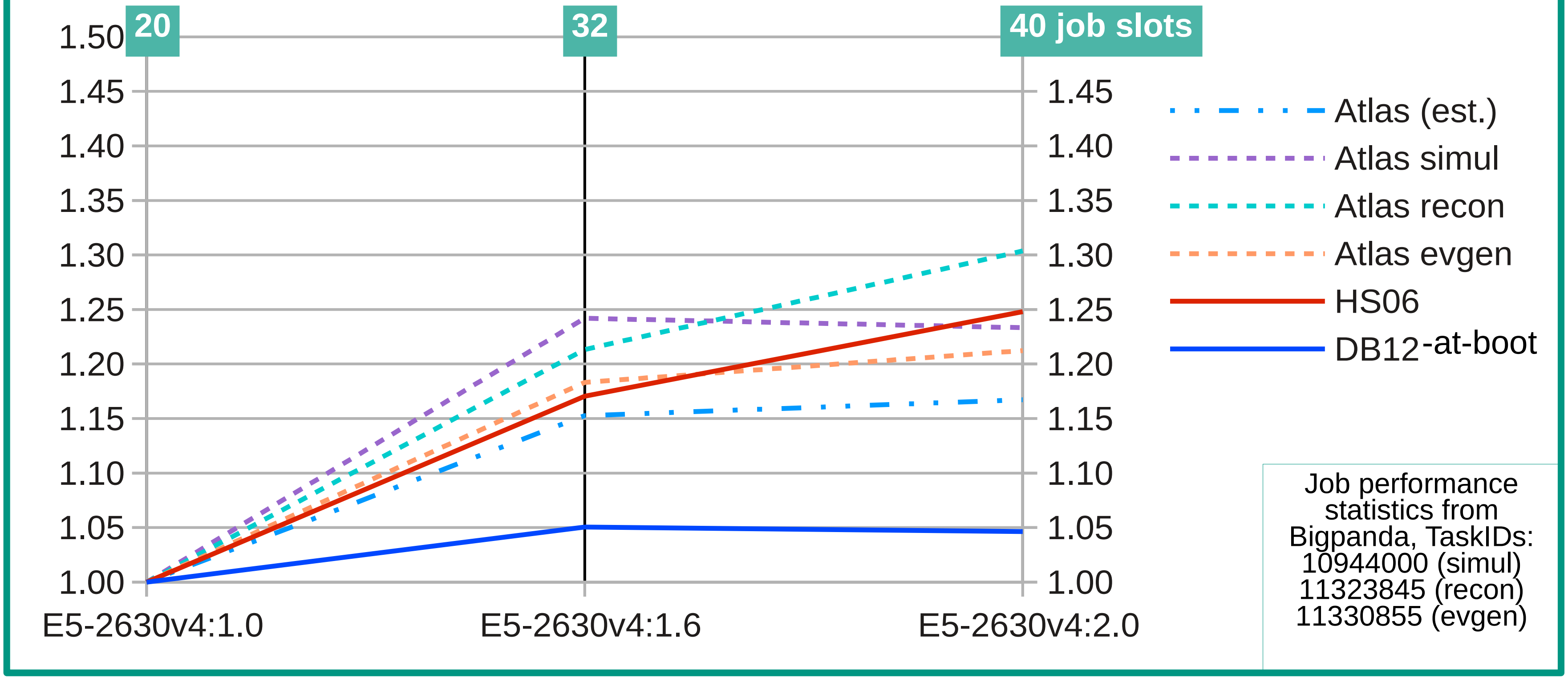


Thanks to Philippe Charpentier for providing performance statistics of LHCb jobs (+ DB16)

Scaling Issues of HS06 vs. HEP Applications

Benchmark Scores vs. Atlas Job Performance (Upscaled)

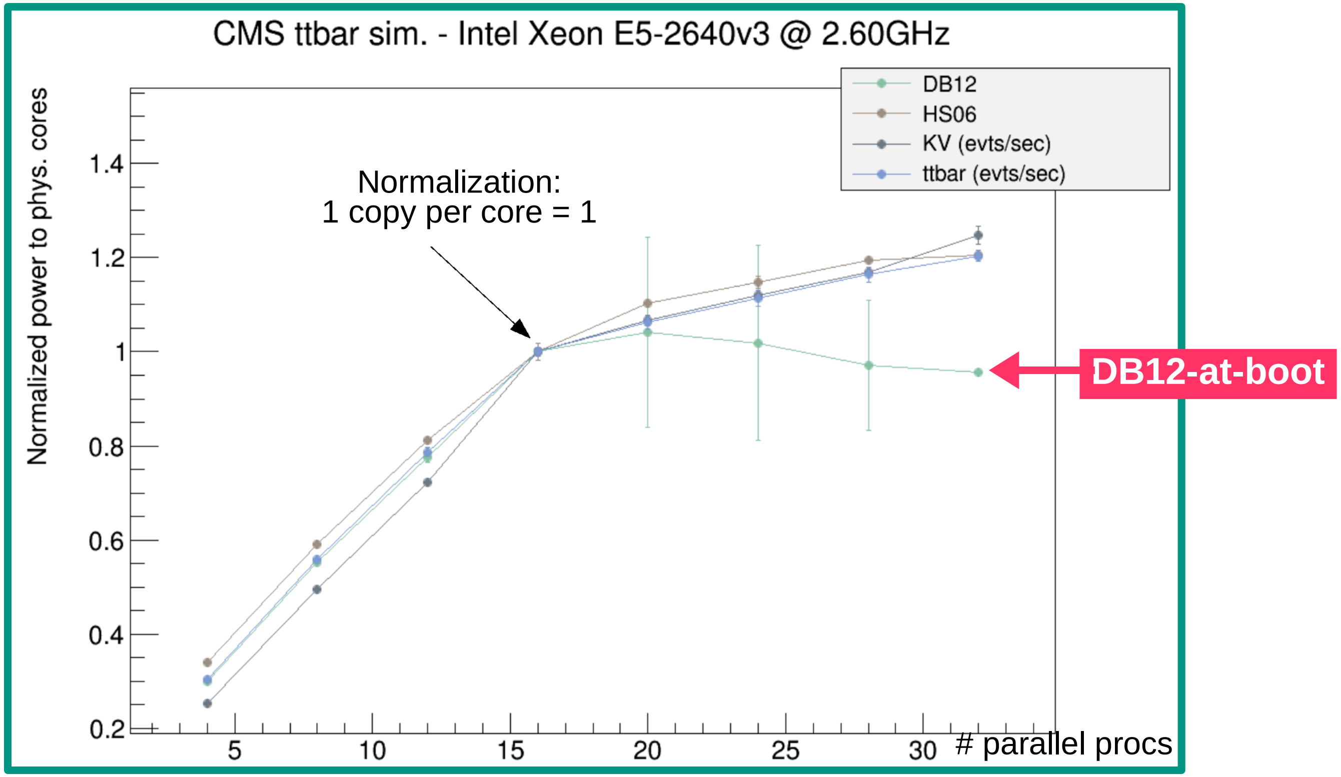
Intel E5-2630v4 (Broadwell) - Normalization: 1 job slot per core = 1



Job performance statistics from Bigpanda, TaskIDs:
10944000 (simul)
11323845 (recon)
11330855 (evgen)

Scaling Issues of HS06 vs. HEP Applications

- CMS:
  - ➔ ttbar sim. at PIC on Haswell host (J. Flix et. al. *):



  - ➔ Estimates at GridKa similar to the Atlas ones

* https://indico.cern.ch/event/624830/contributions/2576000/attachments/1454803/2244865/20170505_CMS_Benchmarking_JFlix.pdf

Summary

- Fast benchmark:
  - → DB12 (in-job) scales with Alice and LHCb jobs
    - Runtime ~1 minute
- Long-running benchmark (HS06 + successor):
  - → Not only the hardware model but also the configured number of job slots per physical core are important
  - → Migration to HS06 64bit doesn’t solve the issues
  - → DB12-at-boot (multiple copies) is <u>not</u> a suitable candidate
  - → Containerising reference workloads (Docker, CVMFS)
  - → Investigating SPEC CPU2017

?