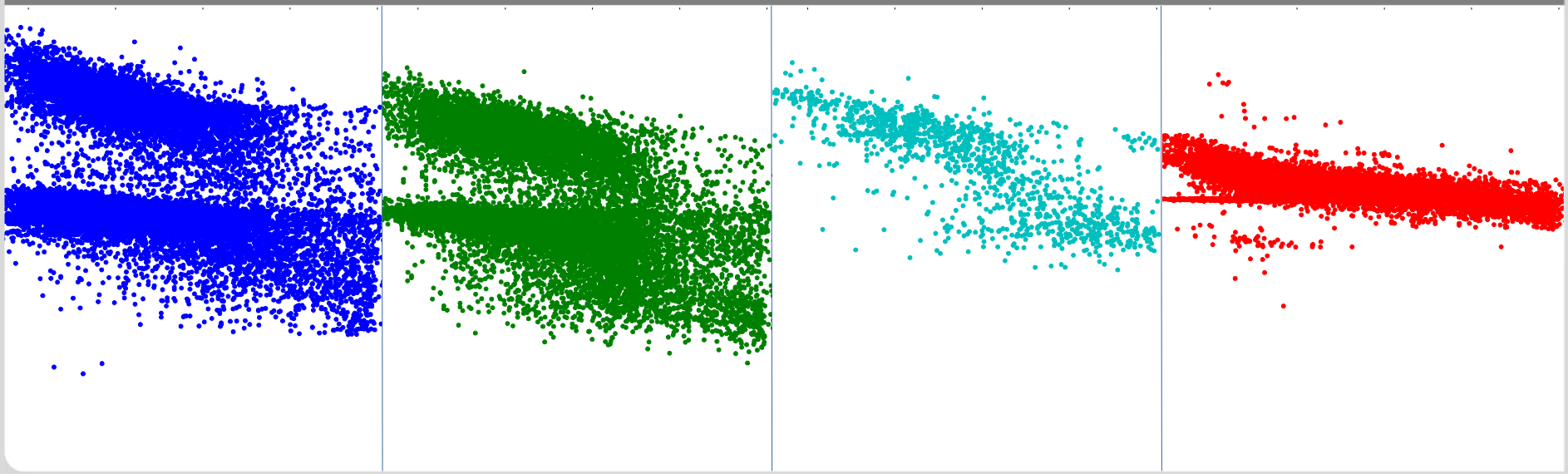
Karlsruhe Institute of Technology

First Suggestions for a WLCG Fast Benchmark

Manfred Alef

WLCG GDB Amsterdam 2016-03-07

Steinbuch Centre for Computing (SCC)



Use Cases and Requirements

- Use cases:
  - Estimate the performance of the provided job slots in case of anonymous hardware (e.g. in commercial clouds)
  - Check the announced performance of the job slots (in batch farm) or a cloud VM

Use Cases and Requirements

- Requirements:
  - As fast as possible
    - ~1 min runtime?
  - As exact as possible
    - Low spread of results
    - Be aware: the shorter the runtime of a fast benchmark the higher the probable inaccuracies
      - What are the neighbors doing?
      - Regular reassessments to iterate
  - Freeware (e.g. GPL), no licensing costs

Benchmark Candidates

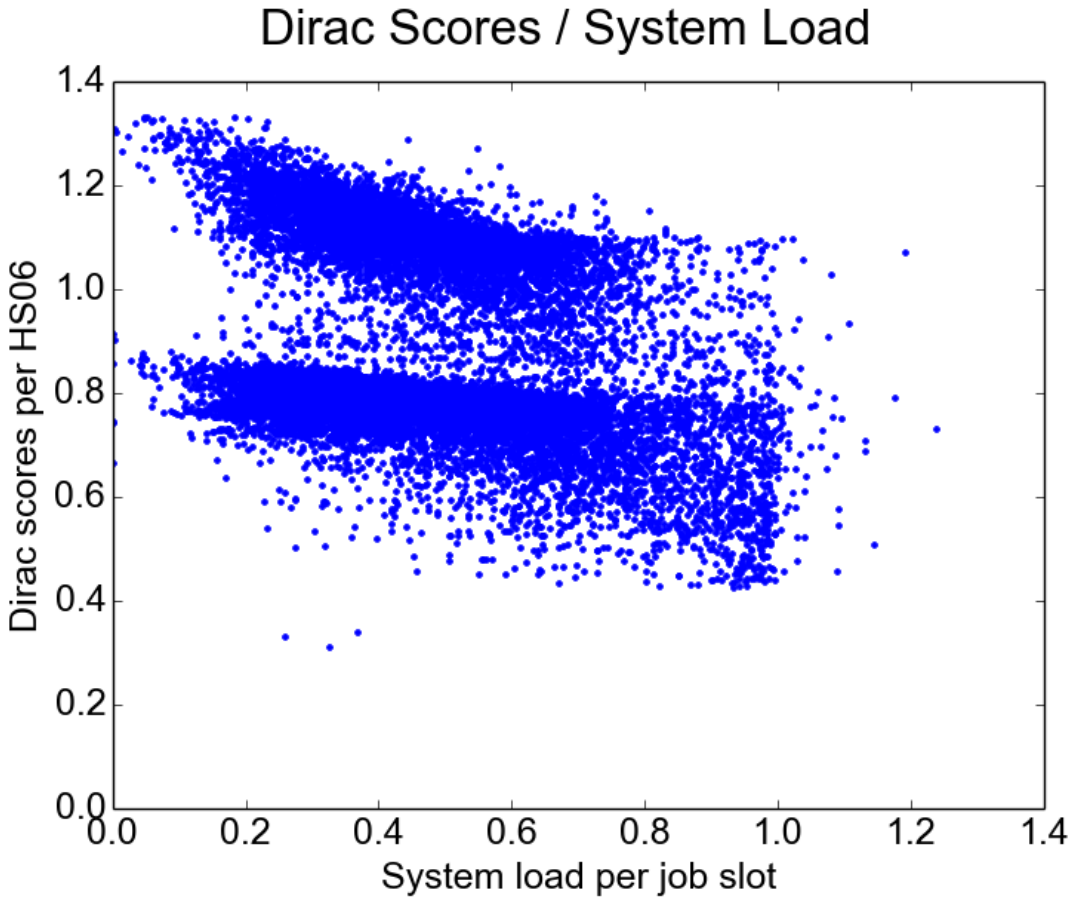
- 5 fast benchmarks related to WLCG use cases
- Analysis:
  - ➔ ~20.000 single-core batch jobs at GridKa measuring
    - The 1min system load at job start time
    - HS06 (from MJF)
    - Results of 4 fast benchmark candidates
      (+ ~2.500 jobs running candidate #5)
  - ➔ Time period: 2 weeks
    - GridKa farm utilization level varying between 50 and 100%

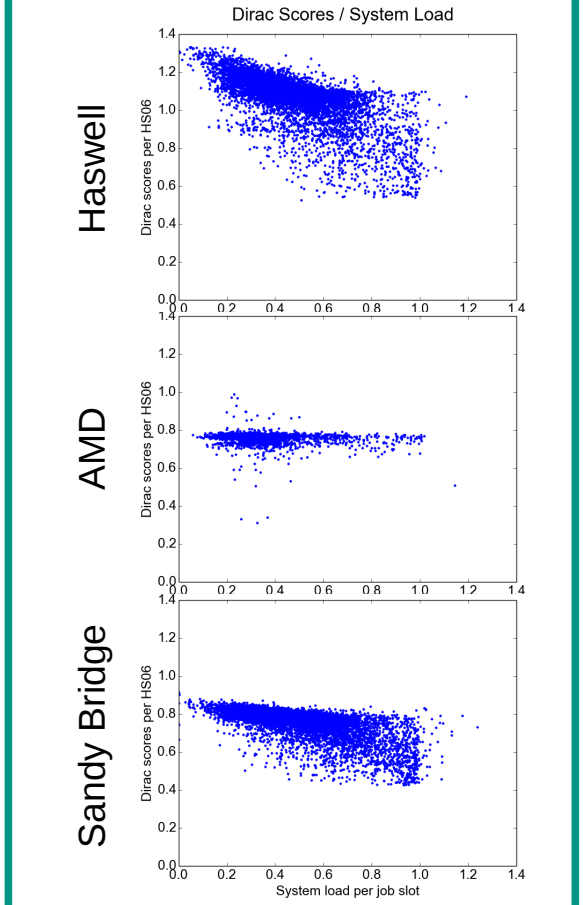
Benchmark Candidate "A"

- Dirac fast benchmark
  - ➔ Used by:
    - LHCb
    - Belle II, ...
  - ➔ Python script
    - Compiler flags are hidden to the user
  - ➔ Fast
    - Runtime: 0:30 ... 2:30 minutes
  - ➔ Correlation with job performance and with HS06:
    - Well scaling with LHCb workload [1,2]
    - Mismatch with HS06 of up to ~40% [3,4]

Benchmark Candidate "A"

- Dirac fast benchmark





Benchmark Candidate "B"

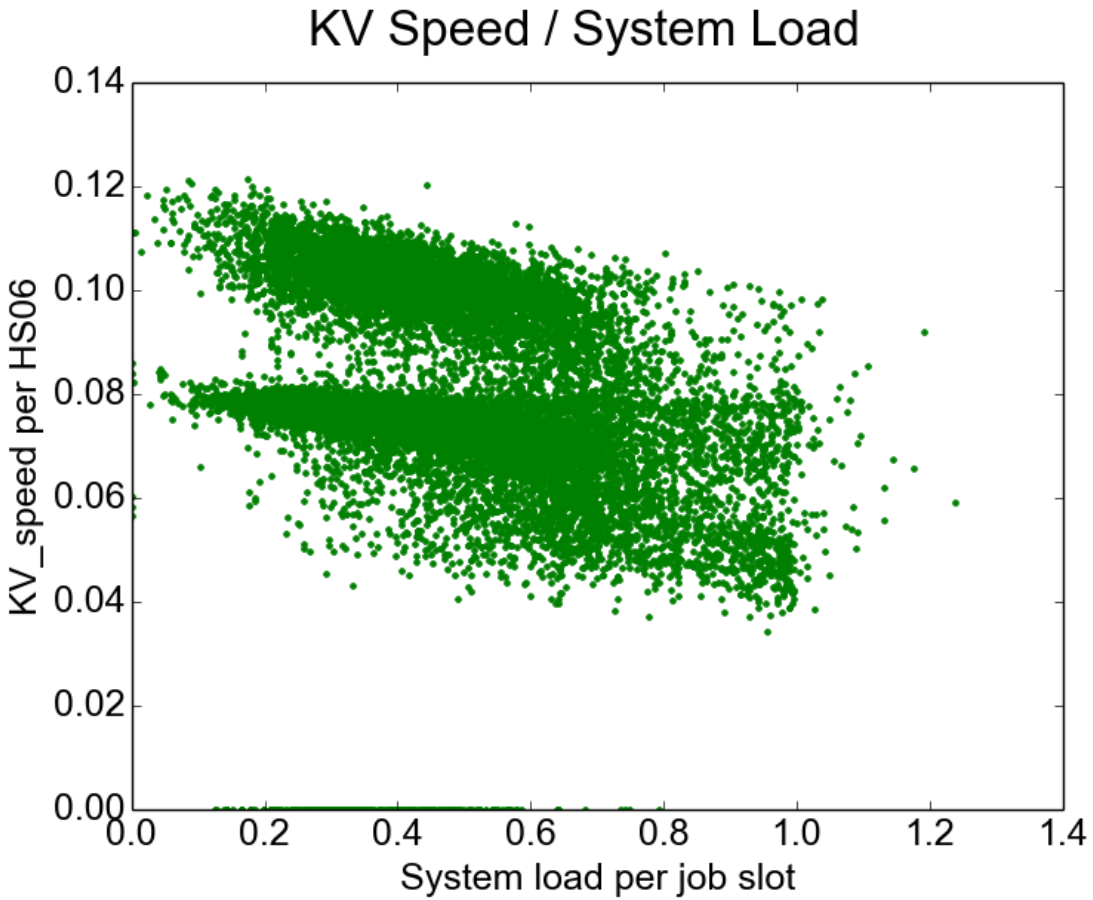
- Atlas KV (KitValidation) benchmark
  - ➔ Used by:
    - Atlas
  - ➔ Framework for benchmarking, various benchmarks can plug-in
    - Wrapper script "kv-script.sh" downloaded from [5]
      - Somewhat modified to run in batch environment instead of VM
    - Tested with default workload (Geant4: 100 Single Muon Events)
    - Compiler flags are hidden to the user

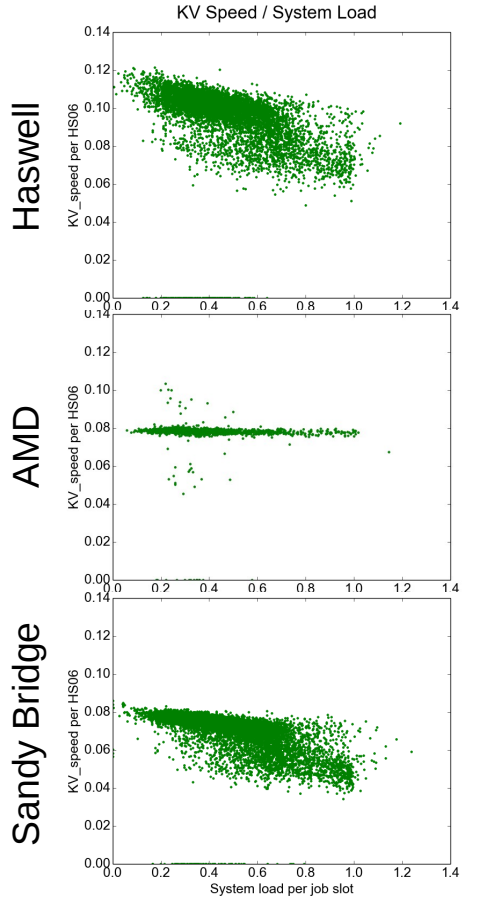
Benchmark Candidate "B"

- Atlas KV (KitValidation) benchmark
  - ➔ Very slow
    - Runtime: 3:45 ... >10:00 minutes
      (average: 6 ... 7 minutes)
    - Includes at least 3 minutes for pre- and post processing and registering with external DB
  - ➔ Correlation with job performance and with HS06:
    - Scaling well with Atlas simulation jobs [6]
    - Probable mismatch with HS06 of up to 40%?
    - *KV score is in units of seconds/event, in this talk: converting to inverse results ("KV speed")*

Benchmark Candidate "B"

- Atlas KV (KitValidation) benchmark



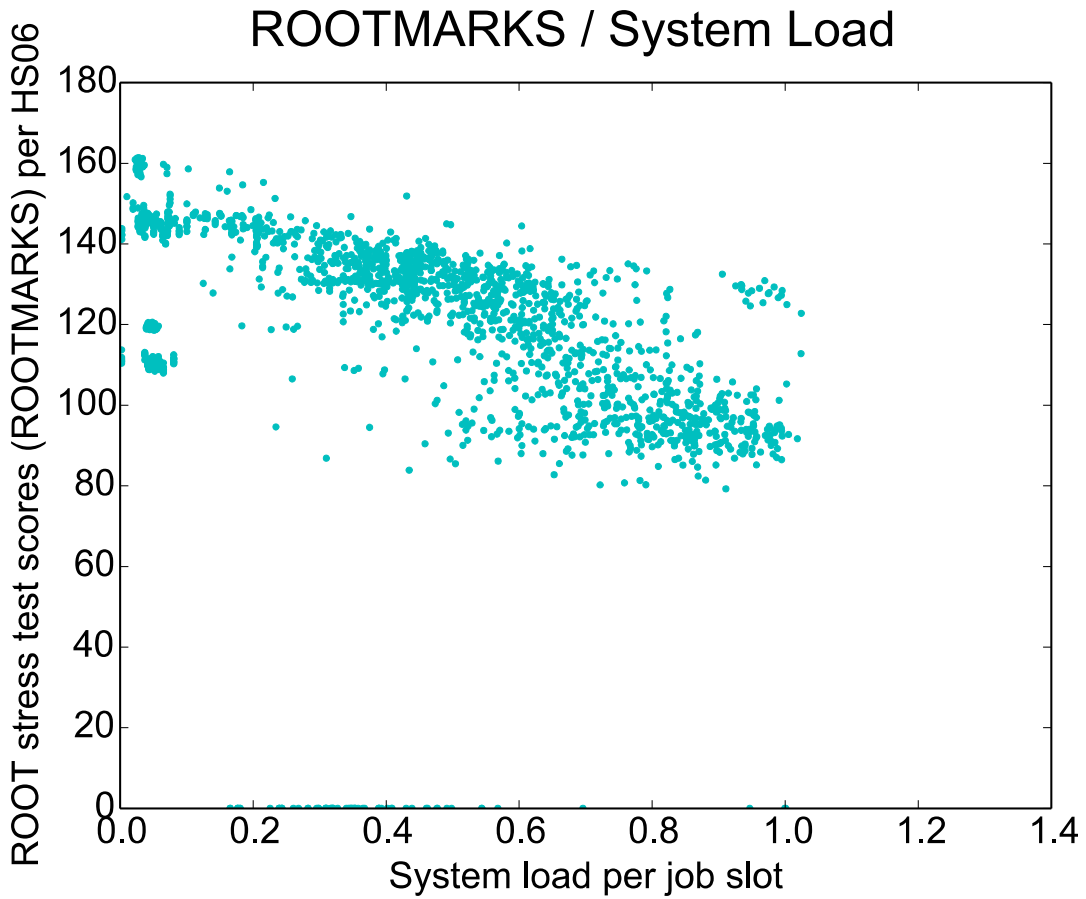


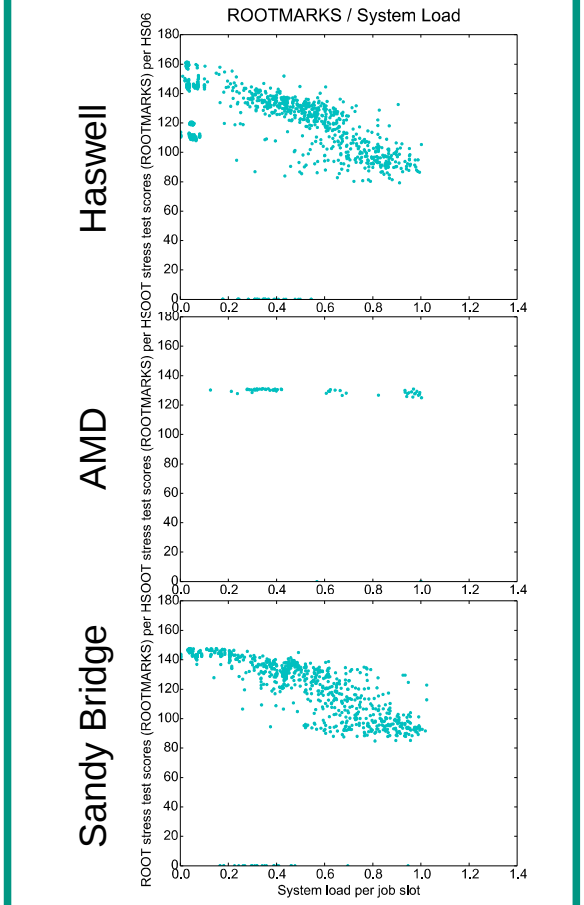
Benchmark Candidate "C"

- ROOT stress test
  - Used by:
    - Alice
  - By default using pre-compiled binary
  - Very fast
    - Runtime: 0:30 minutes
  - Correlation with job performance and with HS06:
    - Scaling linearly with Alice simulation efficiency [7]
    - Better scaling with HS06 than Dirac and KV (AMD's about 30% faster than Intel's if WN busy)

Benchmark Candidate "C"

- ROOT stress test





Benchmark Candidates "D+E"

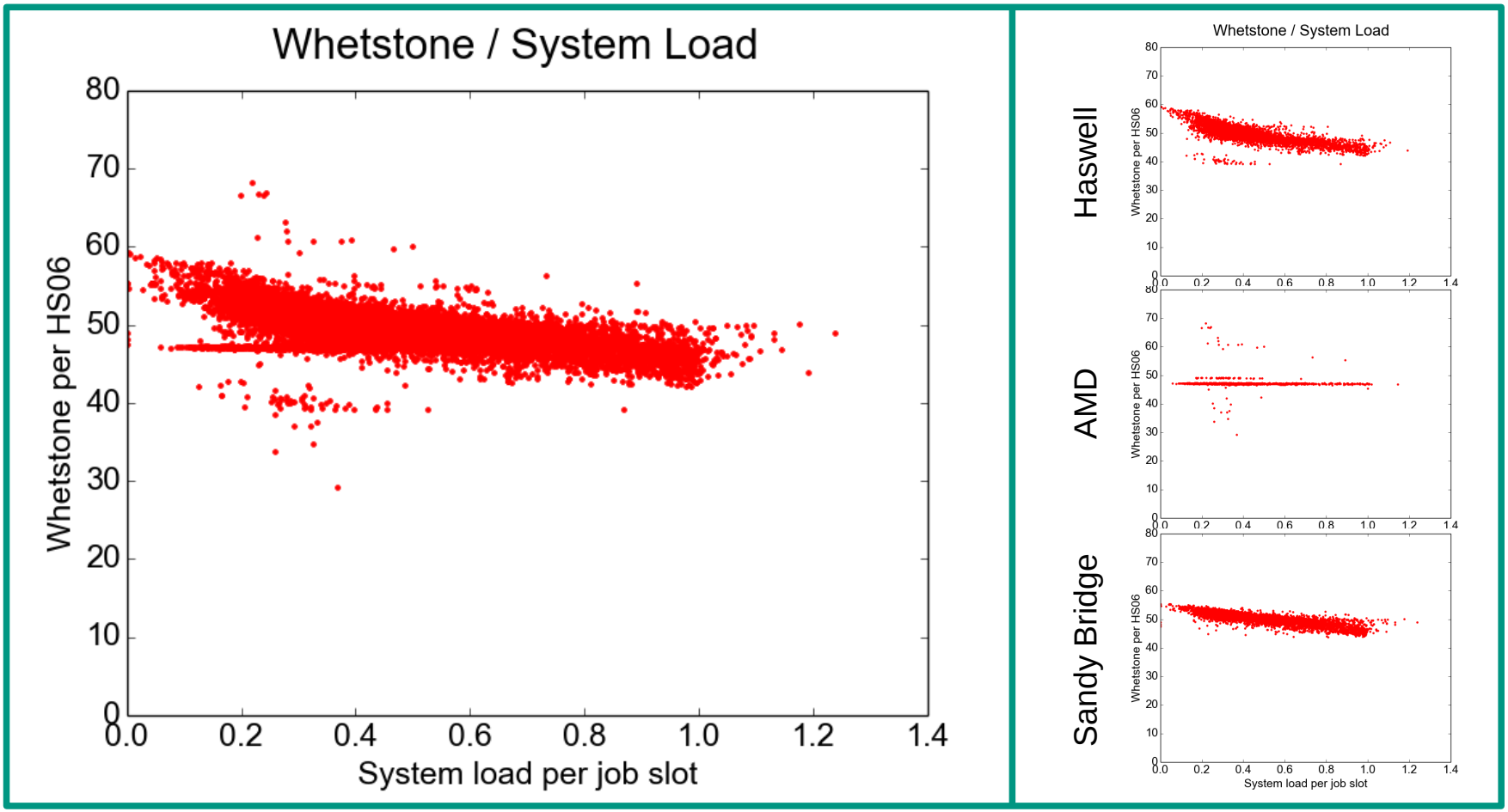
- Whetstone, Dhrystone
  - Used by:
    - Alice, Atlas, ...
    - HTCondor (condor_kflops, condor_mips), Boinc, ...
  - I have used the unmodified makefile (from UnixBench package [8])
  - Very fast
    - Runtime: 2:00 ... 3.00 minutes per benchmark for 10 iterations (default in UnixBench package)
      - Can run less than 10 iterations
  - Very small memory footprint (less than L3 cache size)

Benchmark Candidates "D+E"

- Whetstone
  - ➔ Correlation with job performance and with HS06:
    - Scaling very well with HS06 [3,4]
    - Probable mismatch with WLCG application performance
  - ➔ Not only the dispersion of the HS06-normalized results, but also their dependency on the system load, are very low
    - Most suitable candidate to estimate HS06 in cloud VM's?

Benchmark Candidates "D+E"

- Whetstone

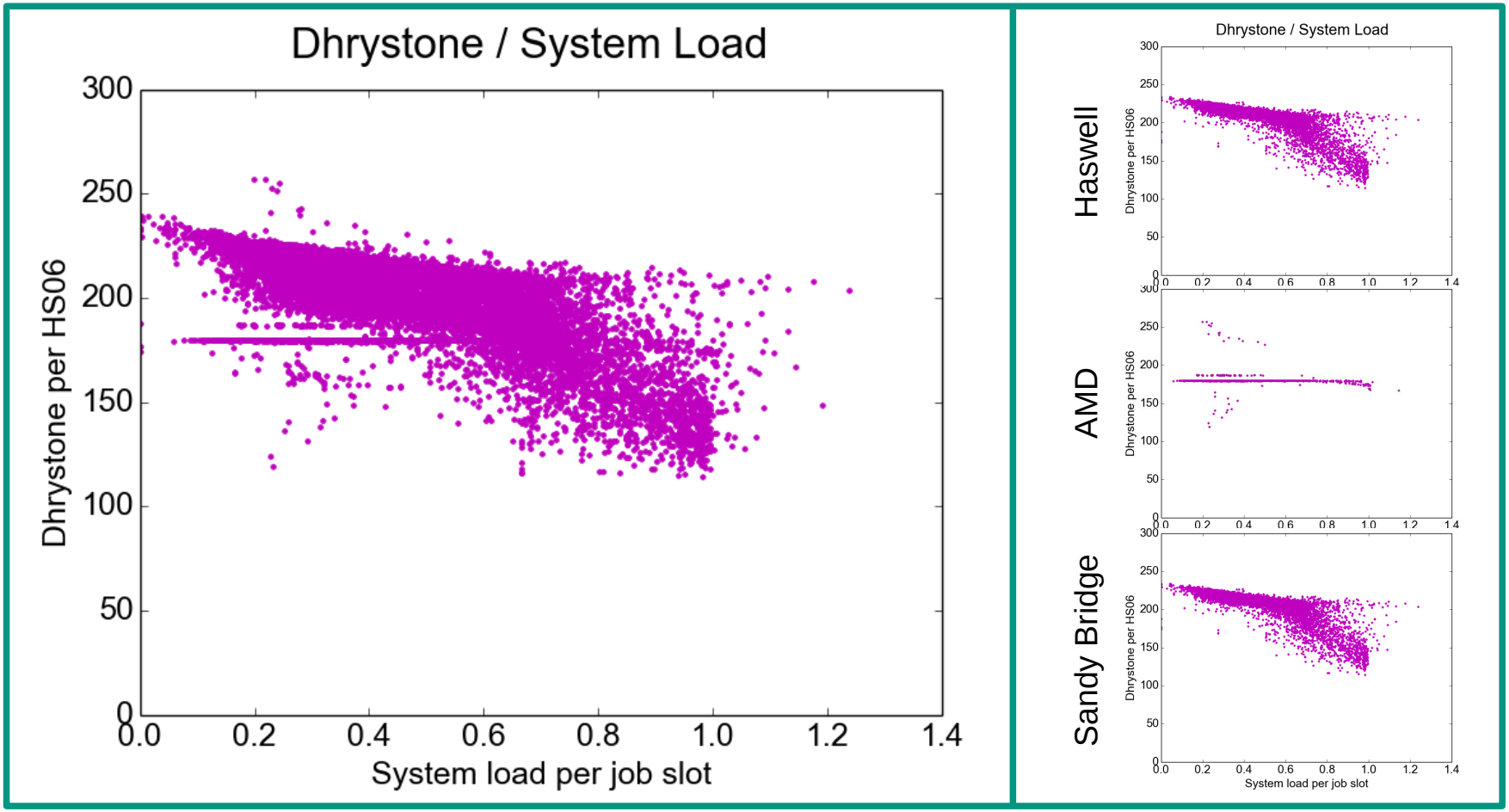


Benchmark Candidates "D+E"

- Dhrystone
  - ➔ Benchmark results very similar to ROOTMARKS

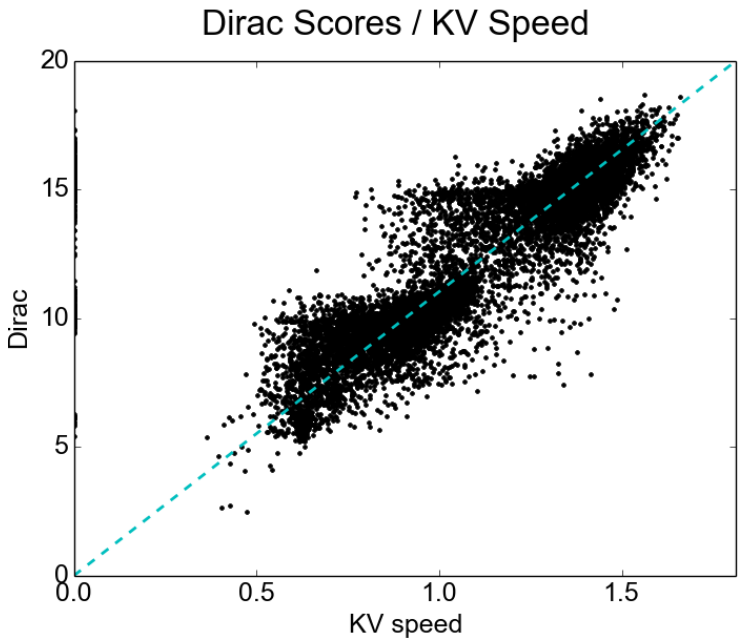
Benchmark Candidates "D+E"

- Dhrystone



Conclusions, Next Steps

- Suggesting 5 fast benchmark candidates (related to WLCG business)
  - ➔ Quite good correlation of **Whetstone** with HS06, suitable candidate for fast HS06 estimate?
  - ➔ **Dirac fast benchmark** and **KV** Single Muon Simulation are scaling well with LHCb and with Atlas simulation jobs
    - Good correlation of both fast benchmarks



  - ➔ **ROOT stress test** doesn't correlate well with (most of) the other fast benchmarks, however it's linear scaling with Alice jobs

Conclusions, Next Steps

- Performance measurements shown in this talk are the results of more than ~20.000 single-core batch jobs at GridKa, at global farm utilization level of around 50 ... 100%
- Cross-checks with other sites, with multi-core jobs, and also in relevant cloud environments, still to be done
  - ➔ Will relaunch the HEPiX Benchmarking Working Group soon

References

[1] https://indico.cern.ch/event/319751/session/0/contribution/6/attachments/1153280/1656518/150909-MJFandBenchmarking-LHCb.pdf

[2] https://indico.cern.ch/event/319754/session/0/contribution/8/attachments/1202029/1749779/151209-MJFUpdate-LHCb.pdf

[3] https://indico.cern.ch/event/319754/session/0/contribution/9/attachments/1202373/1750492/Results_of_HS06_Scaling_Studies_at_KIT_2015-12-09.pdf

{4] https://indico.cern.ch/event/433164/session/2/contribution/9/attachments/1220374/1783838/Results_of_HS06_Scaling_Studies_at_GridKa_2016-02-01.pdf

[5] https://sdccloud.web.cern.ch/sdccloud/KV/DO-29401/index.html

[6] https://indico.cern.ch/event/319751/session/0/contribution/8/attachments/1151865/1653919/gdb-20150910.pdf

[7] https://indico.cern.ch/event/319751/session/0/contribution/9/attachments/1151876/1654130/ALICE_benchmarks.pdf

[8] https://github.com/cloudharmony/unixbench