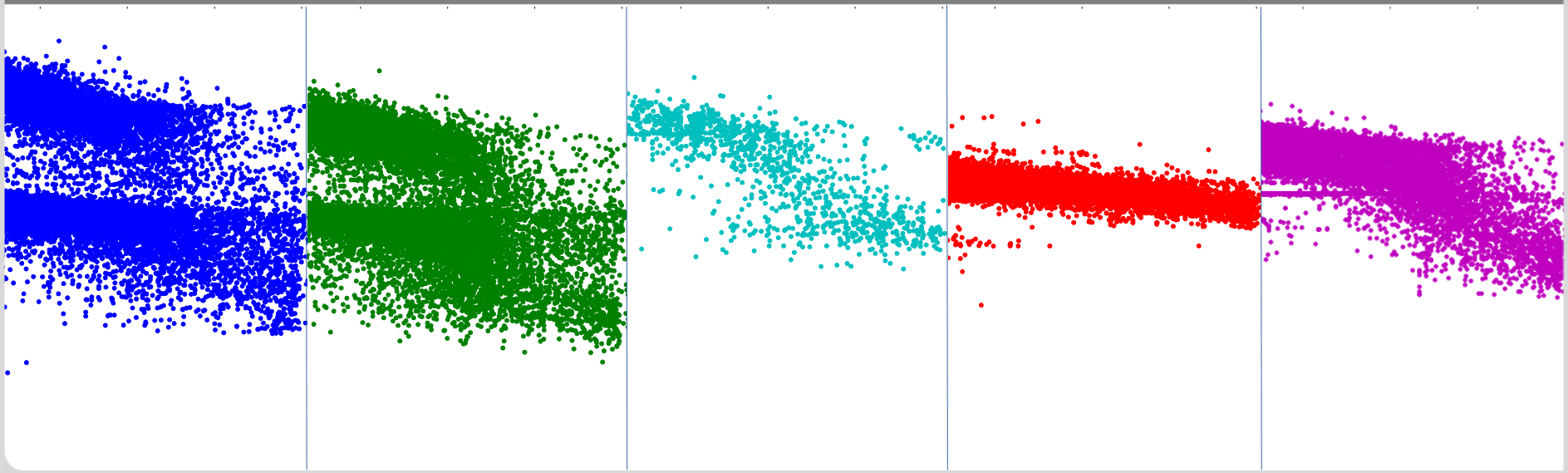
Karlsruhe Institute of Technology

CPU Benchmarking at GridKa
(Update April 2016, HEPiX, Zeuthen)

Manfred Alef

STEINBUCH CENTRE FOR COMPUTING (SCC)



General Remarks

General Remarks

- What's the execution time of a simple arithmetic expression like this:

```
a := b + c
```

  - Very easy exercise, isn't it?

General Remarks

- What's the execution time of a simple arithmetic expression like this:

```
a := b + c
```

  - Very easy exercise, isn't it?
    - Load "b" and "c" into registers
    - Add "b" and "c"
    - Store the result "a"

General Remarks

- What's the execution time of a simple arithmetic expression like this:

```
a := b + c
```

  - Very easy exercise, isn't it?
    - Load "b" and "c" into registers
      - Where are the most recent copies of "b" and "c"?
        - Cache (L1, L2, ...), connected to which processor?
        - Memory (RAM, swap)
        - ...
    - Add "b" and "c"
    - Store the result "a"

General Remarks

- What's the execution time of a simple arithmetic expression like this:

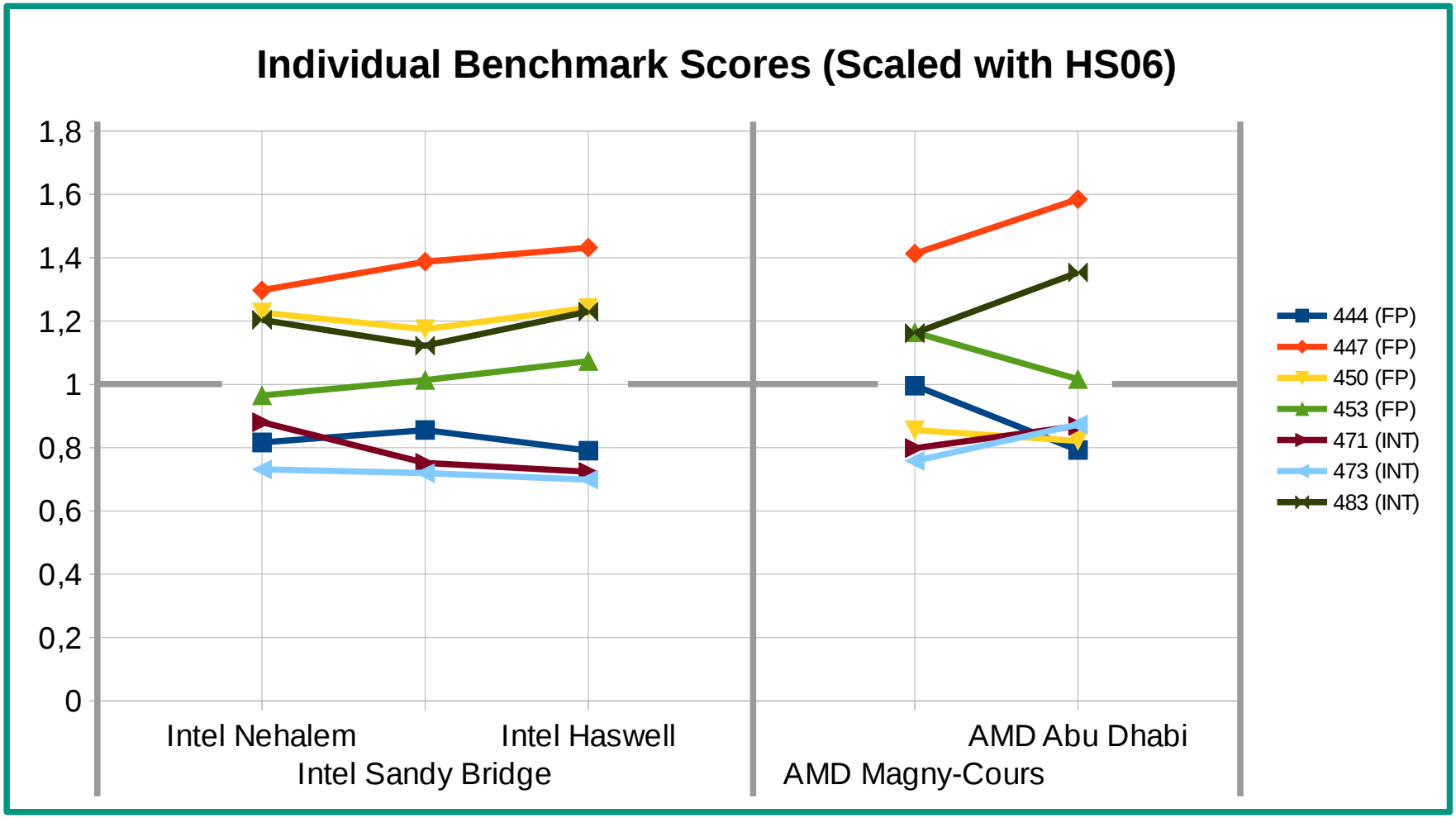
```
a := b + c
```

  - Are we talking about a single instruction, or about a loop?

```
a[0:n] := b[0:n] + c[0:n]
```

    - Some modern processors are coming with vector engines, e.g. AVX (2 operands) or AVX2 (3 operands)
      - Execution time per vector element depends on vector length!
      - Requirements:
        - Compiler must support all relevant hardware features
        - Features are enabled by compiler flags
      - Be aware that an application A1 will possibly run faster than A2 on hardware platform H1 while A2 runs faster on H2: ...

General Remarks



General Remarks

- What's the execution time of a simple arithmetic expression like this:

```
a := b + c
```

  - Which other tasks are running at the same time?
    - Fabric monitoring
    - System updates
    - ...
    - Concurrent VMs sharing the same physical host

General Remarks

- A benchmark can never scale exactly with each individual application (nor class of applications) on all relevant hardware types
- Benchmarks should scale with a typical application, or with a representative mix of applications
  - HS06 shall scale with the typical WLCG job mix
    - "Good enough" (10% initial objective)
  - Proved good correlation between HS06 scores and representative WLCG applications on various hardware platforms
- Remember that HS06 themselves is a mix of benchmarks (3 INT, 4 FP)

Issues with the Default Benchmark (HS06)

Issues with the Default Benchmark (HS06)

- Philippe Charpentier (LHCb) – see GDB 2015-09-09 + 2015-12-09:
  - Performance of LHCb jobs (comparing with HS06 score of the provided batch slot) about 45% faster on WNs with Haswell processors, and around 30% on WNs with AMD chips, than on hosts with Sandy Bridge processors
- Dirac (LHCb) fast benchmark results looking very similar
- Performance of Atlas jobs ('nevent/cpuconsumptiontime' from bigpanda store, compared with HS06):
  - Similar to the LHCb results for Atlas simulation jobs :-(
  - Linear scaling for all other types of Atlas jobs :-)
- Alice, CMS?

Issues with the Default Benchmark (HS06)

- Probable reasons:
  - Some jobs are possibly enabling platform dependent hardware features, e.g. vector engines (AVX2)
    - HS06 is based on the default compiler (gcc-4.4.x on SL6), and on a mandatory set of flags ('–O2 –pthread –fPIC –m32')
      - On Haswell box: 444.namd runs 20% faster when compiled with latest compiler gcc-5.3.x and with more aggressive flags '–O3 –m64 –march=haswell'
    - Are WLCG experiments using their own, more recent compiler implementations?
    - Hardware features are possibly enabled by default in SL6 Python packages (boosting Dirac fast benchmark)
  - Bonus performance for user jobs, no degradation

Fast Benchmarks

Fast Benchmarks

- HS06 is a tool suitable to describe installed capacities, procurements, accounting, fair-share batch scheduling, aso, and must scale well with the typical job mix running at WLCG sites
- However, HS06 is inapplicable in several scenarios, for instance users cannot run it to estimate the performance of the provided batch slot or VM:
  - Runtime
  - License issues
- Demand for a fast benchmark to estimate the performance
  - Runtime ≈ 1 minute
  - Probably less precise than full benchmark, but "good enough"
  - Free license

Fast Benchmarks

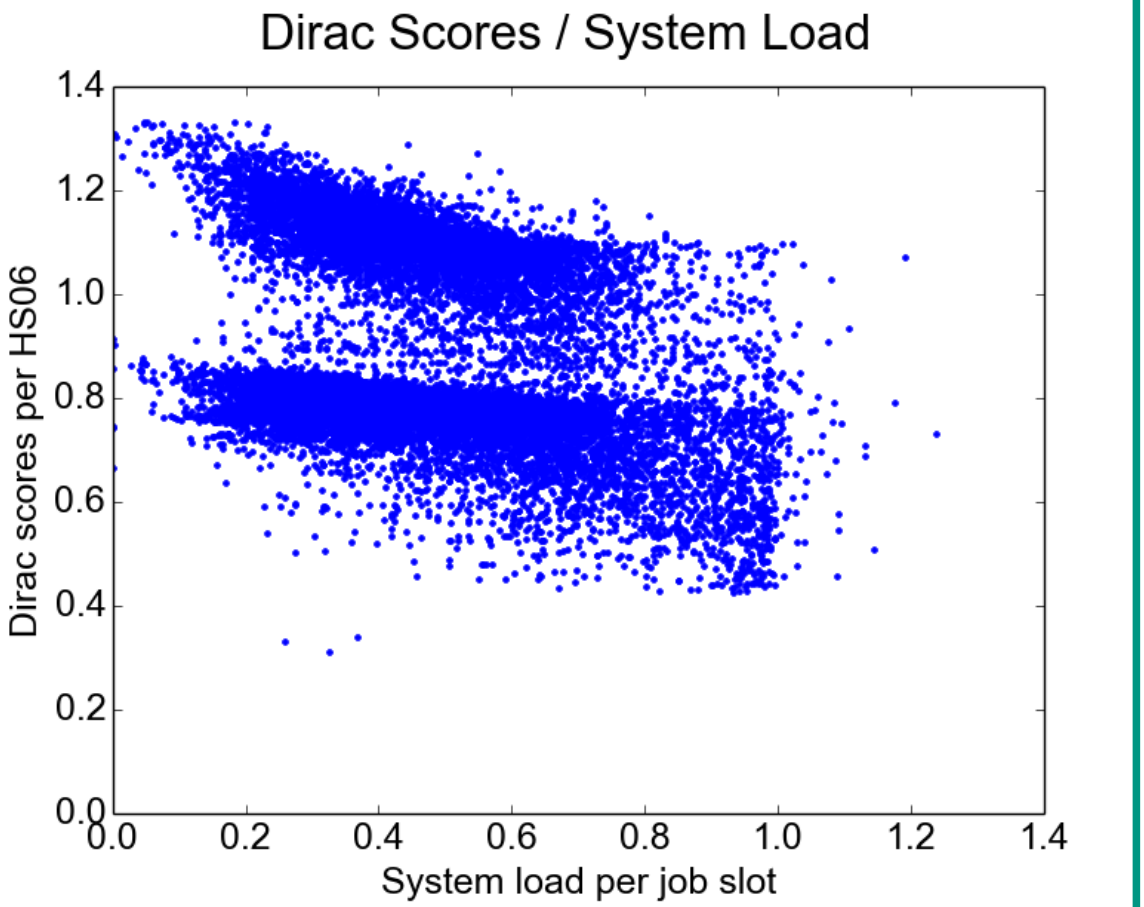
- Suggesting 5 candidates:
  - Dirac (LHCb) fast benchmark
    - LHCb, Belle II, ...
  - Atlas KV / Geant4 single muon events
    - Atlas
  - ROOT stress test
    - Alice
  - Whetstone, Dhrystone
    - HTCondor, Boinc, ...

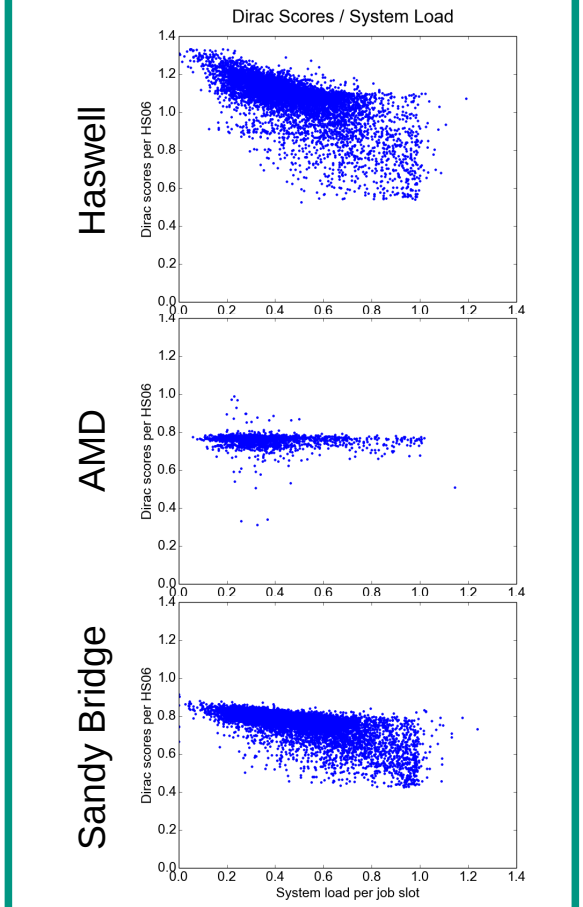
Fast Benchmarks

- First investigations:
  - Measurements in GridKa batch farm
    - ≈ 20,000 single-core jobs
      - System load at job start time
      - HS06 of the provided slot (from MJF)
      - Running several fast benchmarks
    - Farm utilization level varying between 50% and 100%
    - Results ...

Fast Benchmarks

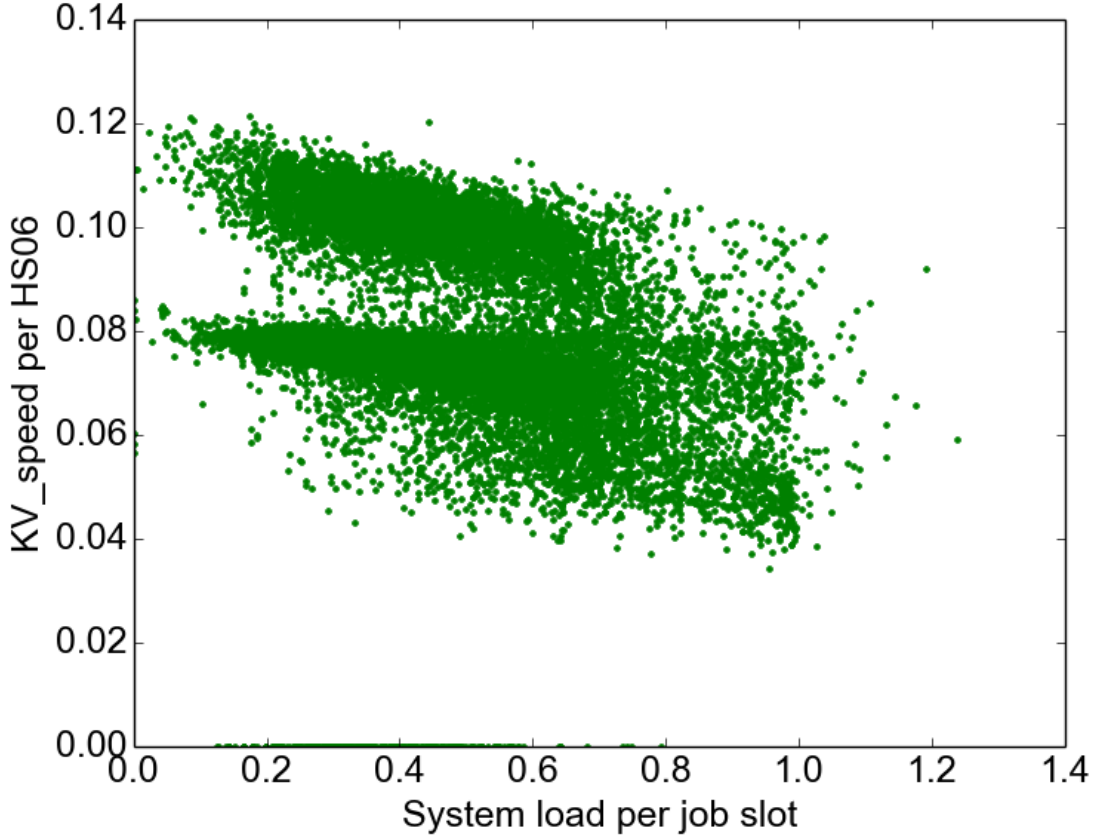
- **Dirac fast benchmark** 0:30 ... 2:30 minutes

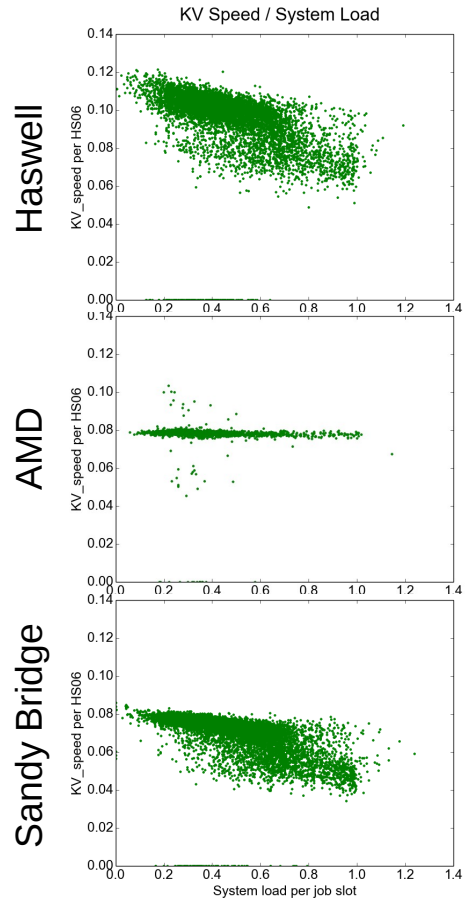




Fast Benchmarks

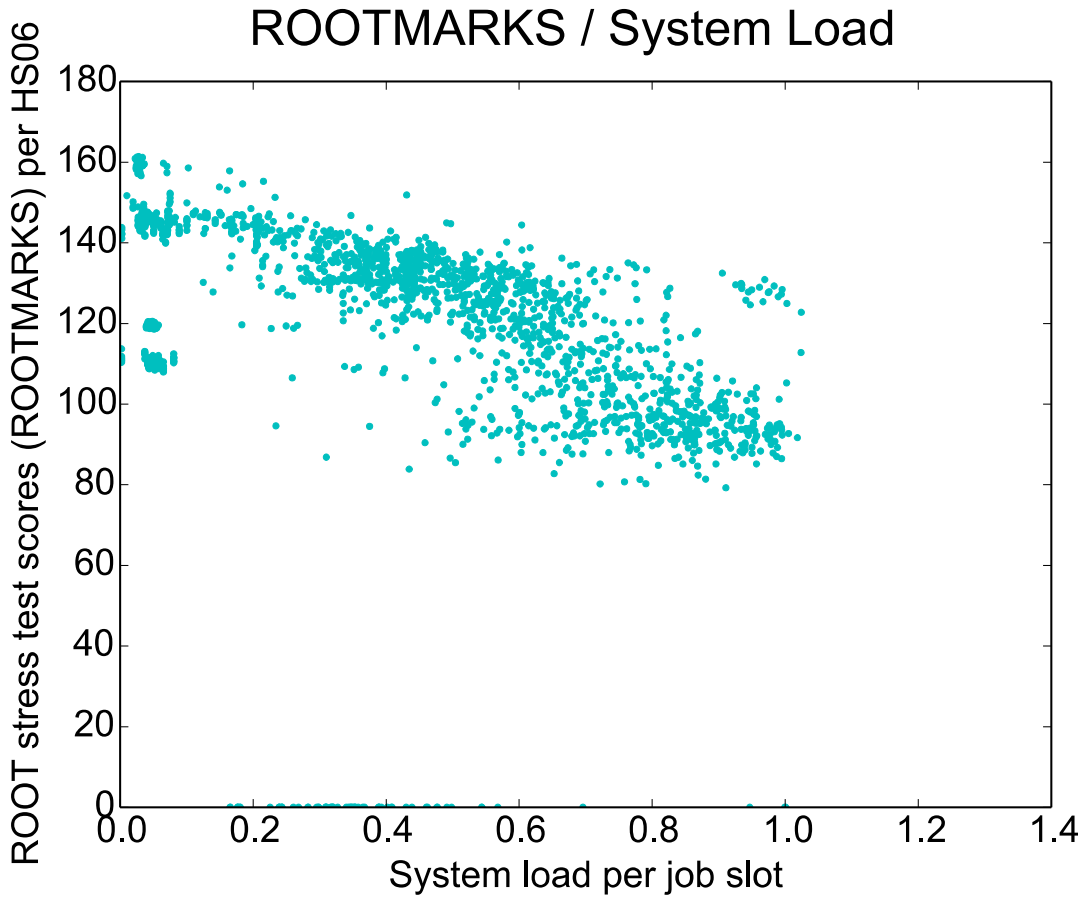
- **Atlas KV (KitValidation) benchmark** 6 ... 7 (–3) minutes

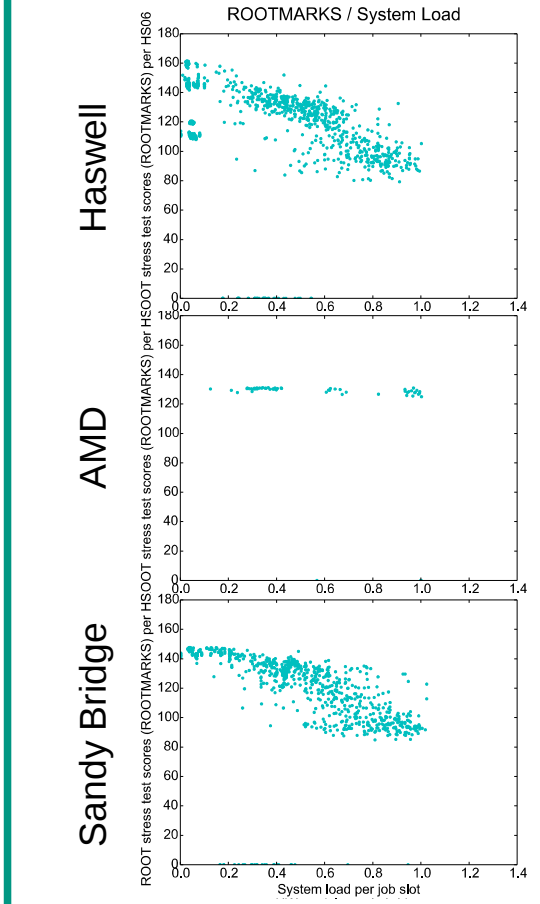




Fast Benchmarks

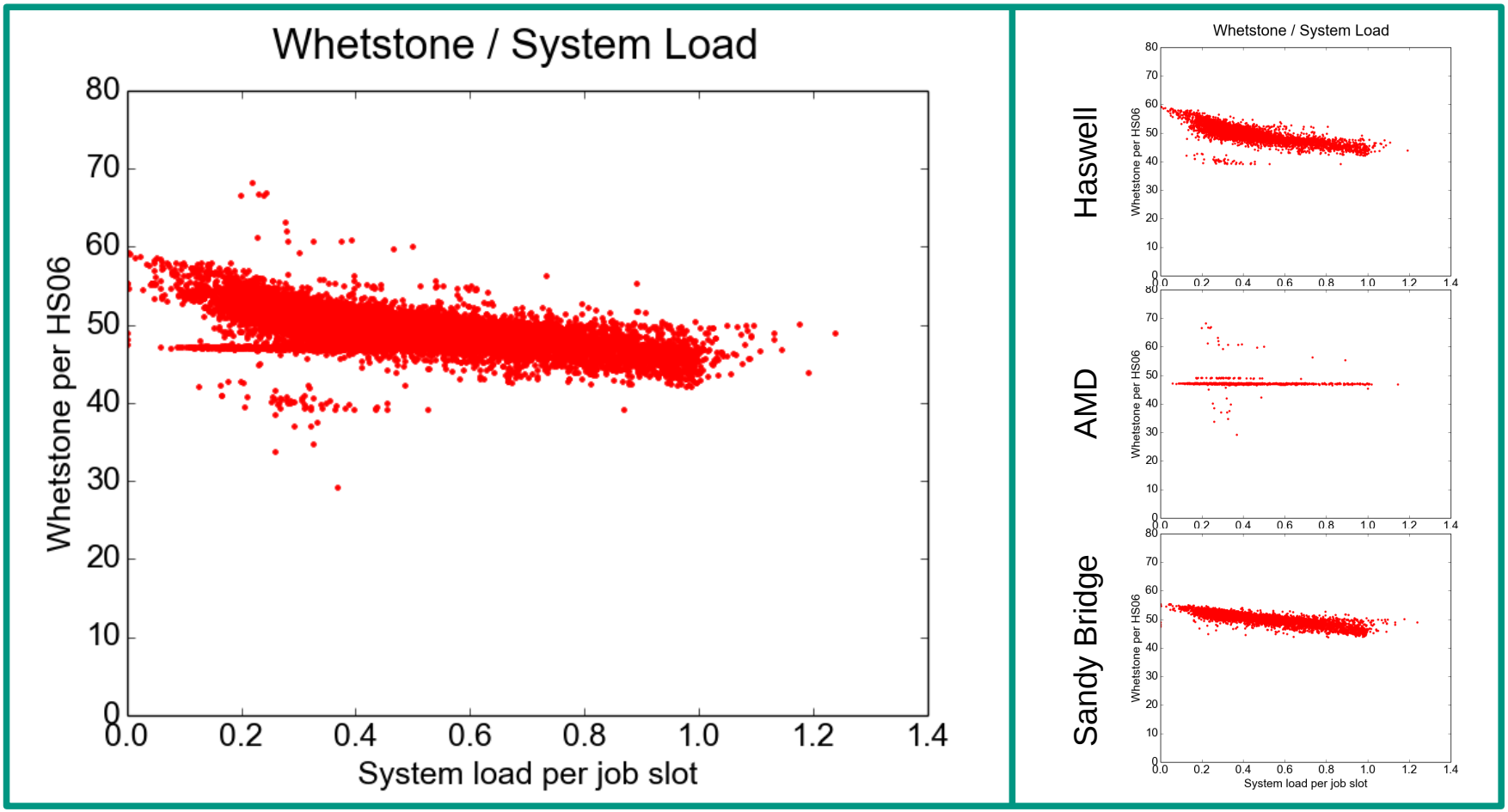
- **ROOT stress test** 0:30 minutes





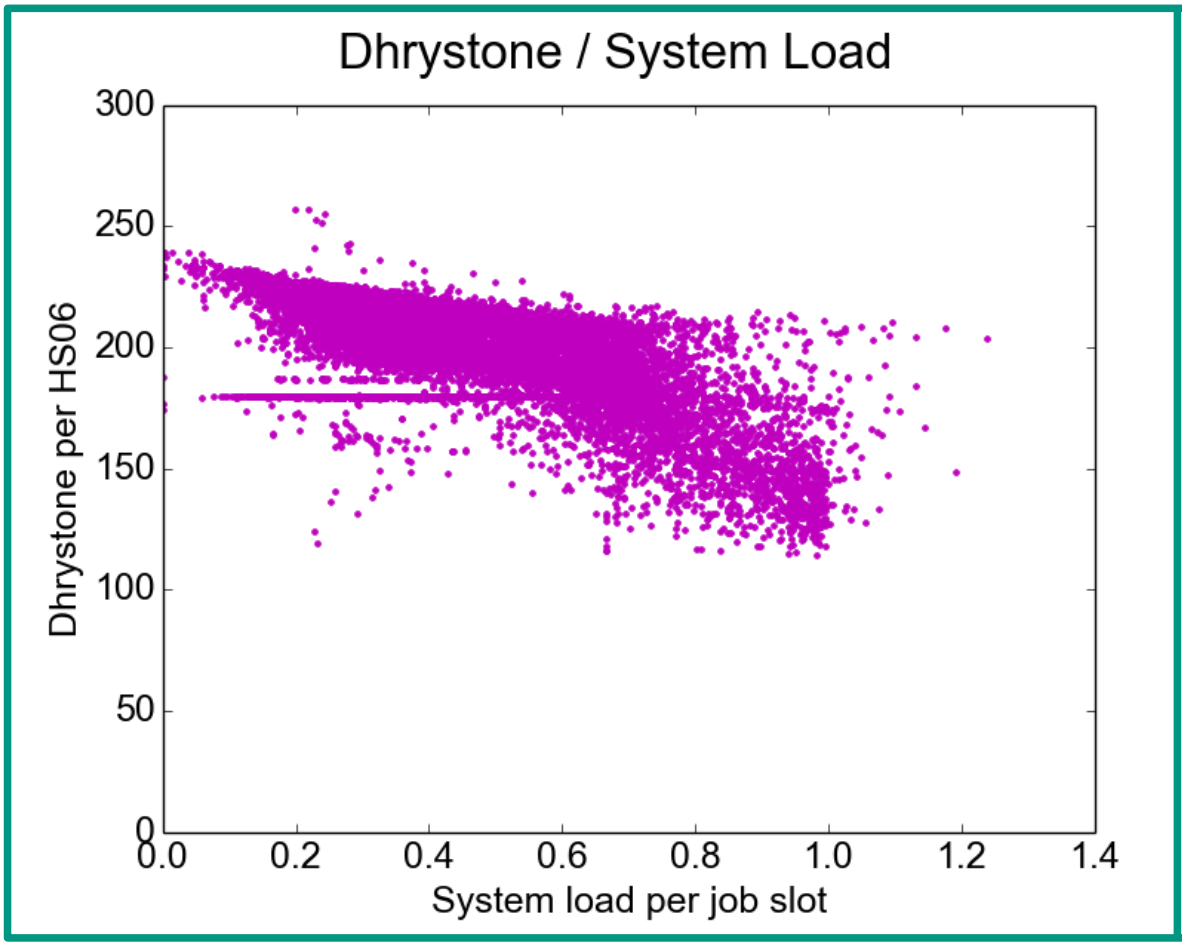
Fast Benchmarks

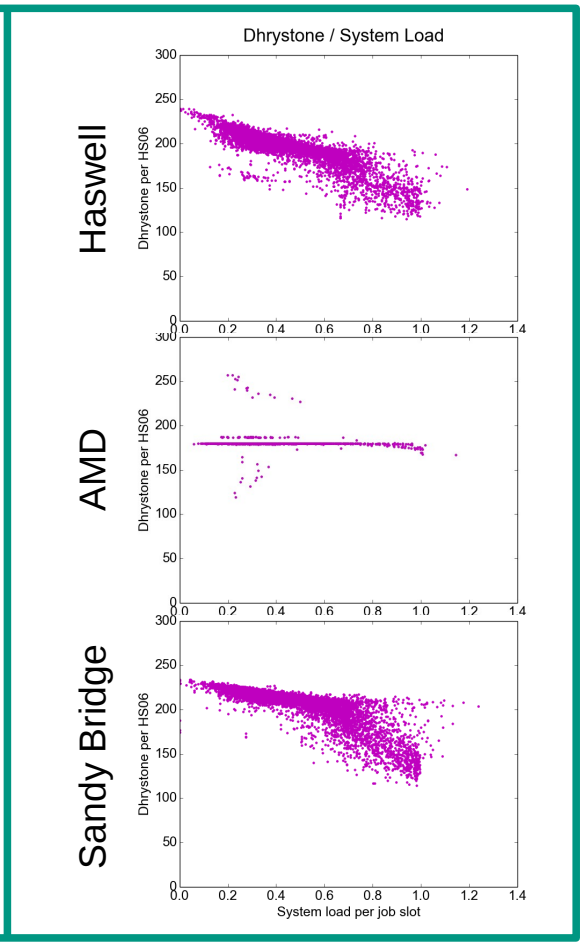
- **Whetstone** (UnixBench) 2 ... 3 minutes (10 iterations)



Fast Benchmarks

- **Dhrystone** (UnixBench) 2 ... 3 minutes (10 iterations)





HEPiX Benchmarking Working Group

HEPiX Benchmarking Working Group

- First informal meeting tomorrow
- Will start with deeper look into scaling and accuracies of fast benchmarks
  - Performance estimation of anonymous hardware, e.g. commercial clouds
  - Scaling with typical WLCG applications running on various hardware (VM) models
  - Spread of results, e.g. depending on system load
    - What are the neighbours doing?
  - Multi-core benchmarking

Conclusions

Conclusions

- The official WLCG benchmark HS06 doesn't scale with some of the most important applications on some hardware models
  - Mismatch of up to 45% (underestimation)
- The intended use cases of HS06 are installed capacities (accounting, procurements, fair-share batch scheduling, aso.)
  - HS06 must scale well (≈ 10%) with typical job mix at WLCG sites
- Performance prediction of individual jobs is not an intended use case!
  - Complementary (fast) benchmark(s) useful
  - (Repeating) accuracy less important
- Relaunch of the HEPiX Benchmarking Working Group
  - Starting with fast benchmarks

Questions, Comments