

ALICE CPU benchmarks

Costin.Grigoras@cern.ch



Context

- **~70% of the Grid time is taken by simulation jobs**
- **A benchmark reflecting the MC performance would help with the purchasing of new hardware**
- **HS06 is not representative for our workload, especially on new CPUs**
- **So we've been looking for alternatives**

Benchmark considerations

- **Simple to find and to run**
- **Short execution time relative to the job duration**
 - For automatic benchmarking of nodes
- **Reflecting the experiment's software performance on the hardware**
- **Simplified method to collect and summarize the results**
- **No licensing concerns**
- **Easier sharing of configuration and results**
- **Reproducible results**

MC simulation vs benchmarks

- **Reference production:**

- “pp 13 TeV, new PYTHIA6(Perugia-2011) min.bias, LHC15f anchors”
- 200 ev/job, avg(8h) running time, CPU-intensive
- Blanket production, 76 sites

- **Benchmarks:**

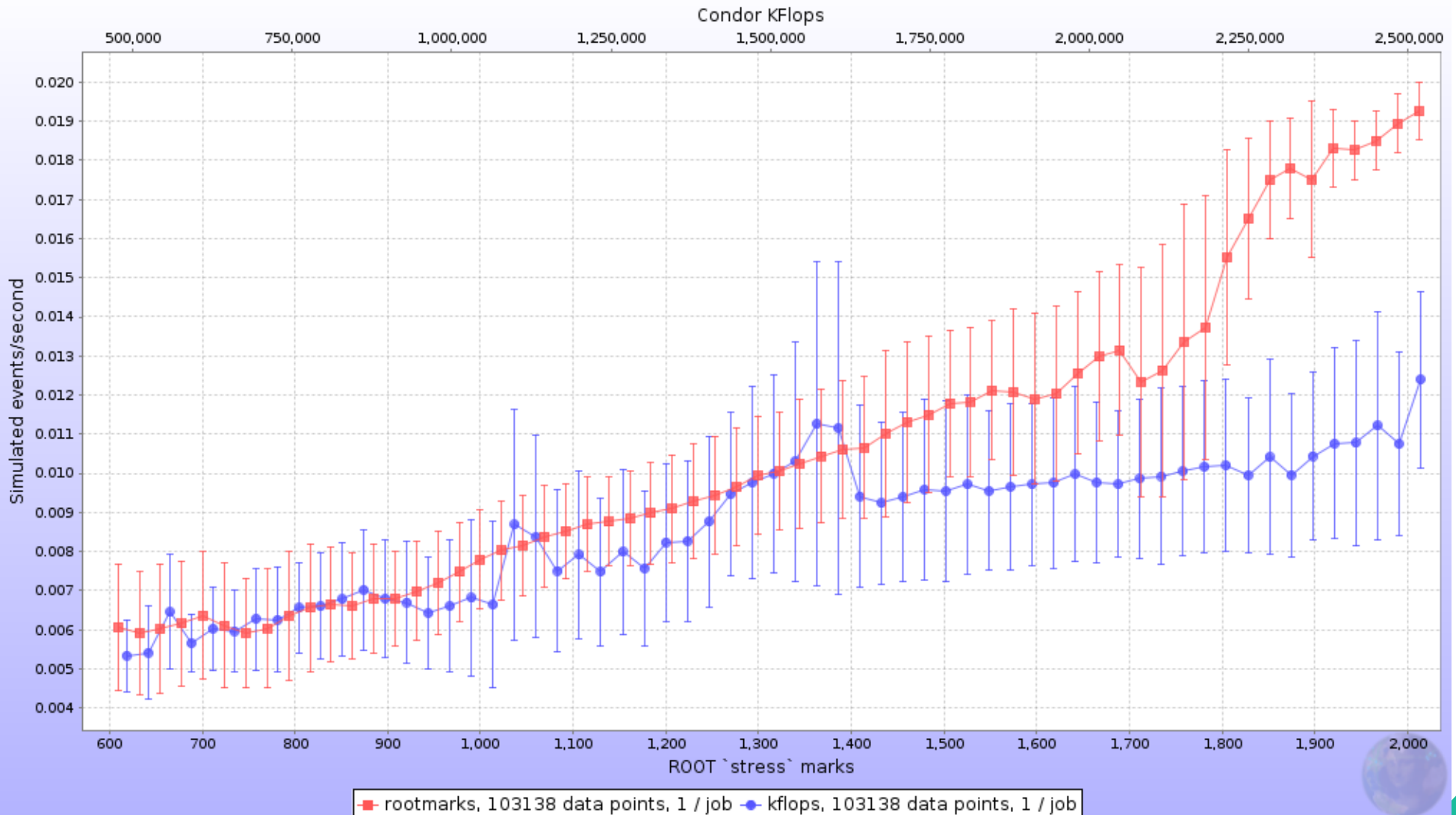
- ROOT's */test/stress* (O(30s))
- *condor_kflops* from ATLAS' repository (if found) (O(15s))

- **Each benchmark ran twice after the simulation**

- To fill in the CVMFS cache and load the libraries in mem
- Recording the second iteration only

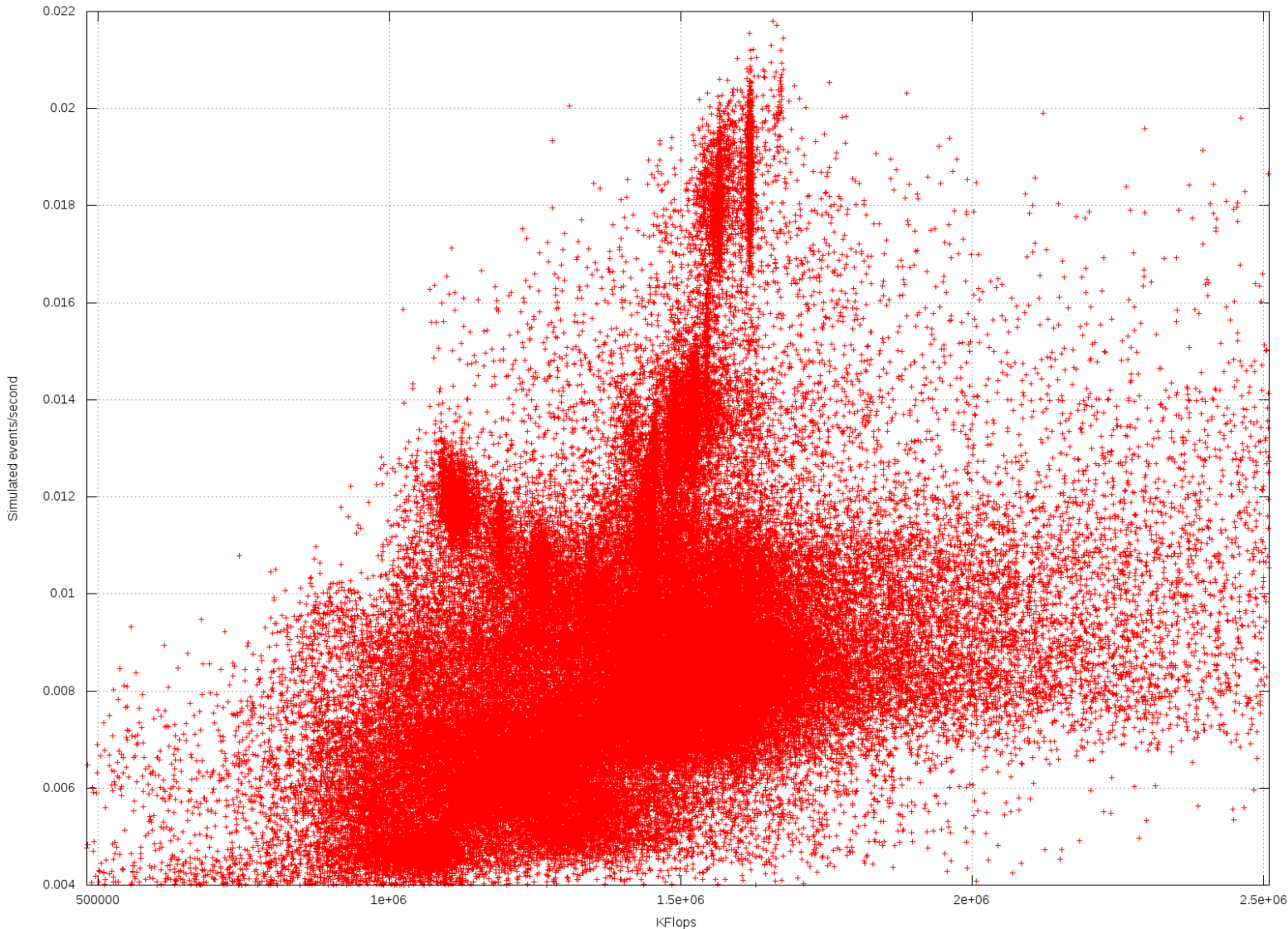
Results at a glance

ALICE pp event simulation time vs ROOT `stress` marks and Condor KFlops



Events/s vs KFlops

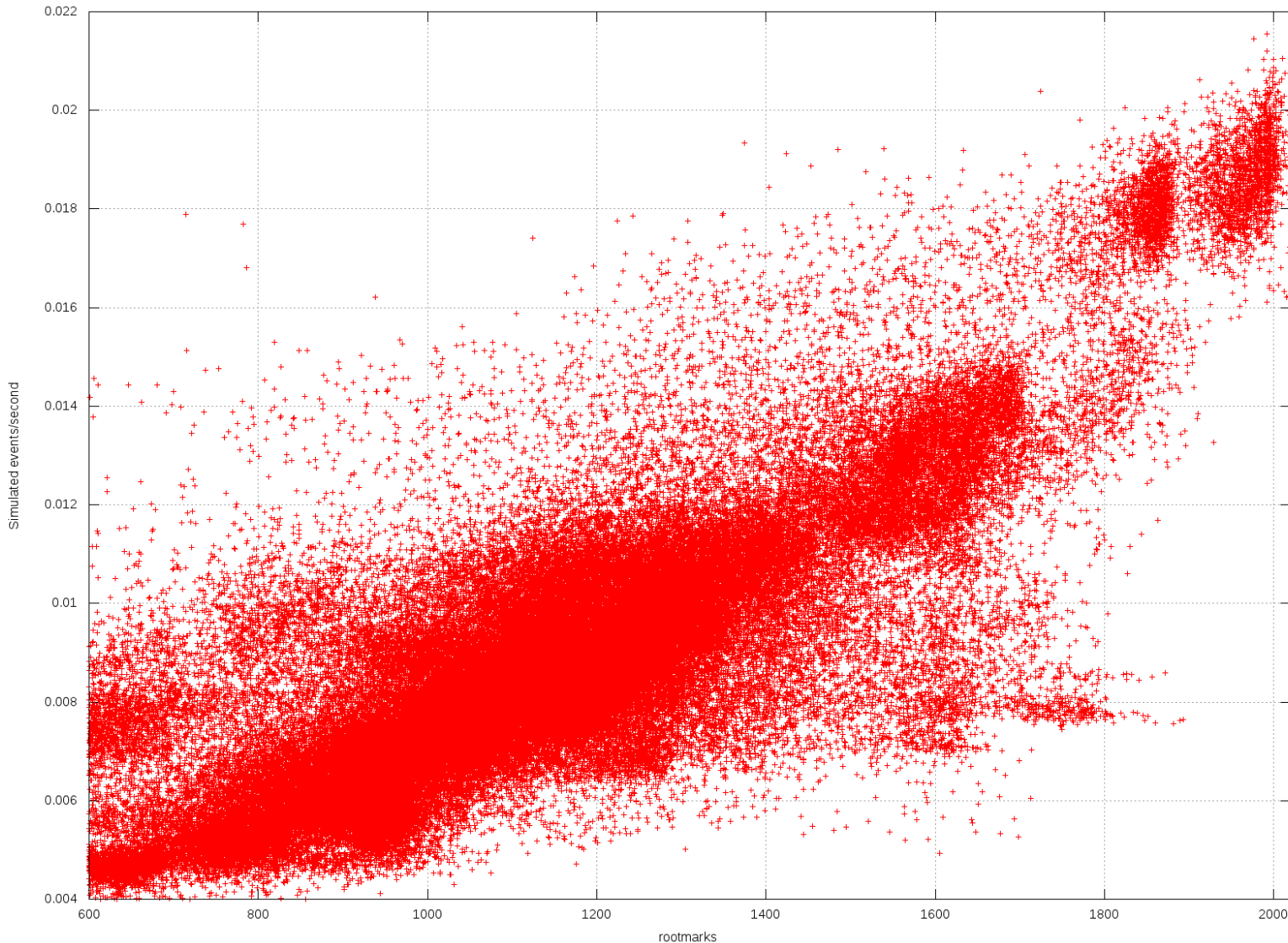
ALICE pp event simulation time vs Condor KFlops



No correlation between Kflops and simulation performance, probably because of small ratio of floating point operations in it.

Events/s vs rootmarks

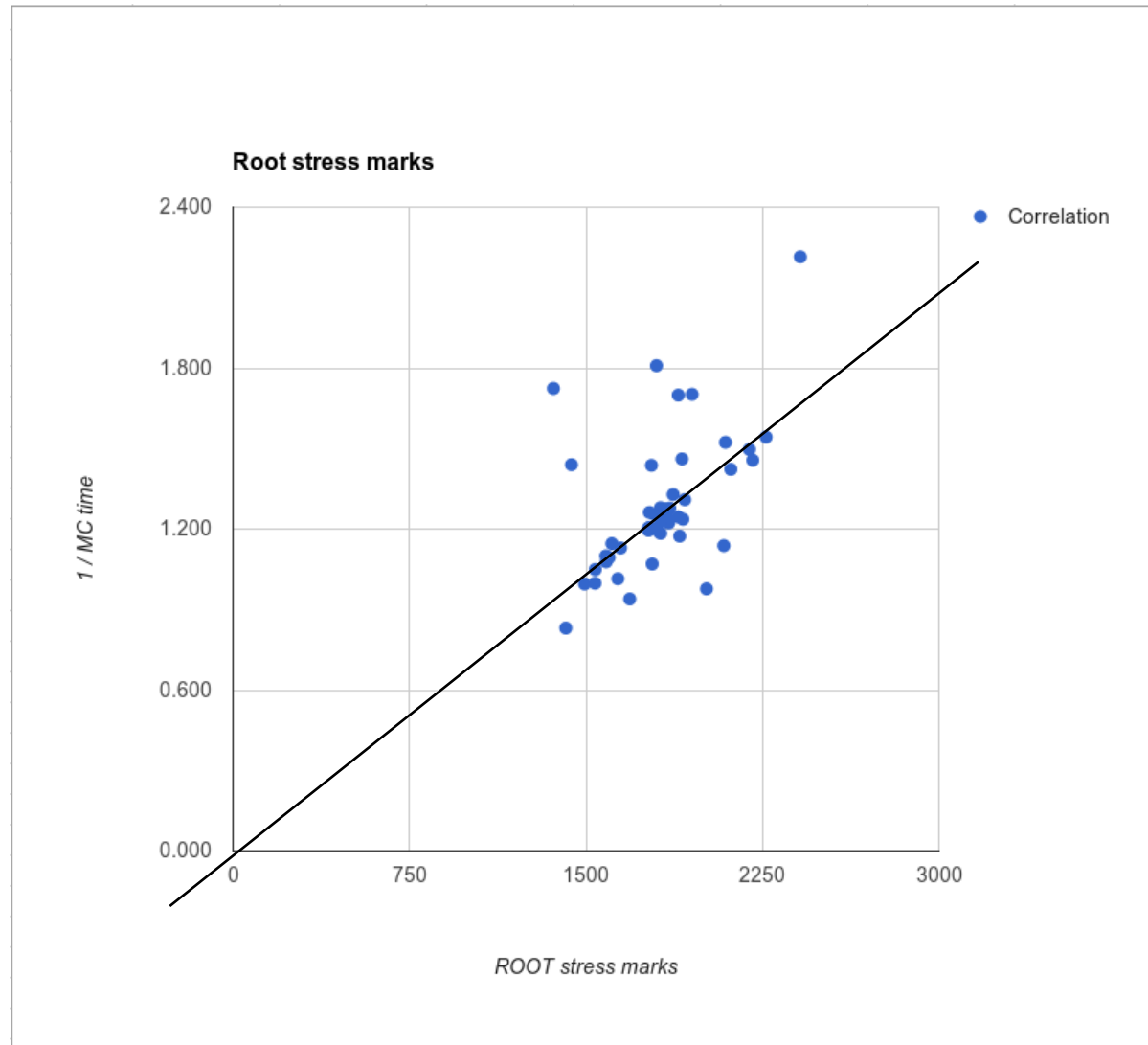
ALICE pp event simulation time vs ROOT marks



**Rootmarks
scale ~better
with the
simulation time
in the Grid
environment**

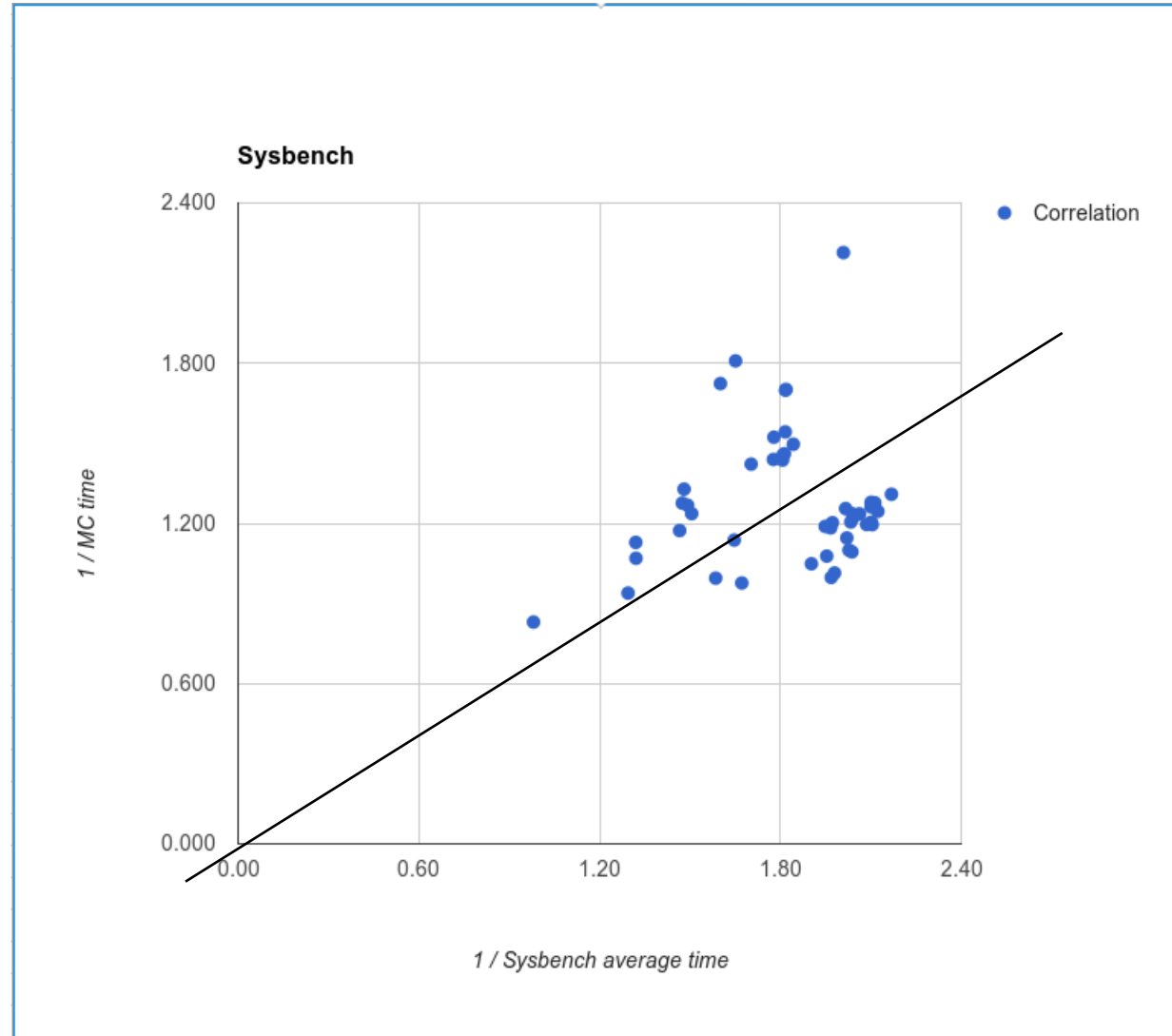
Is there anything better still?

- Running in a controlled environment
 - Central services (~50 hosts)
- ROOT stress test results don't look that good



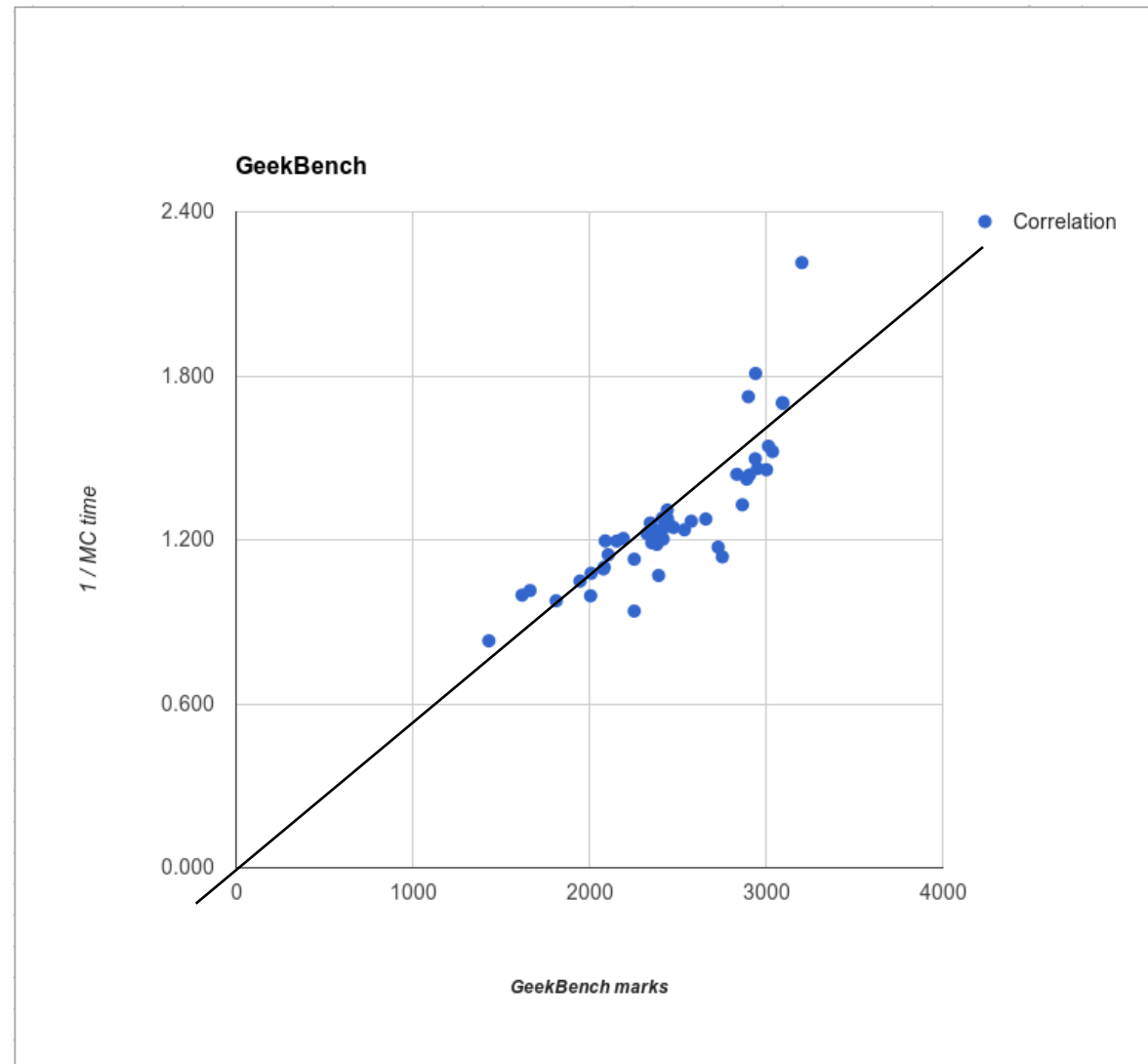
Sysbench ?

- Available by default on many Linux variants
- ~30s to run
- But it doesn't scale well...



GeekBench

- Commercial product
- Testing the 32b evaluation version
- ~2min to run

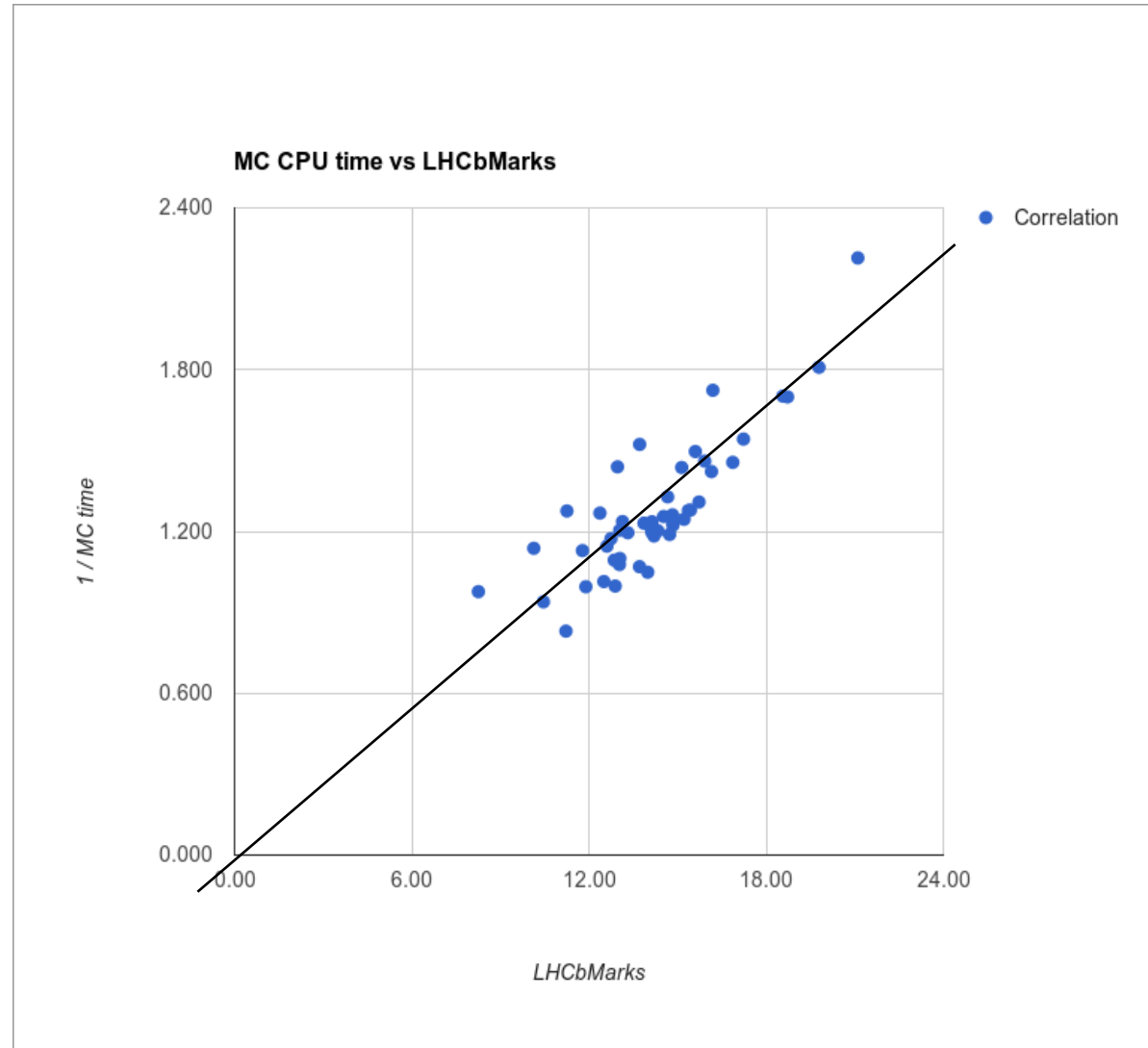


GeekBench, cont.

- **Promising results so far**
- **Single binary, easy to run**
- **Clarify licensing for our environment**
- **Run both the 32b and 64b Grid-wide**
- **Saving the results in a local file**
 - No direct way to fetch the results (web interface only) in the trial version

LHCb's test

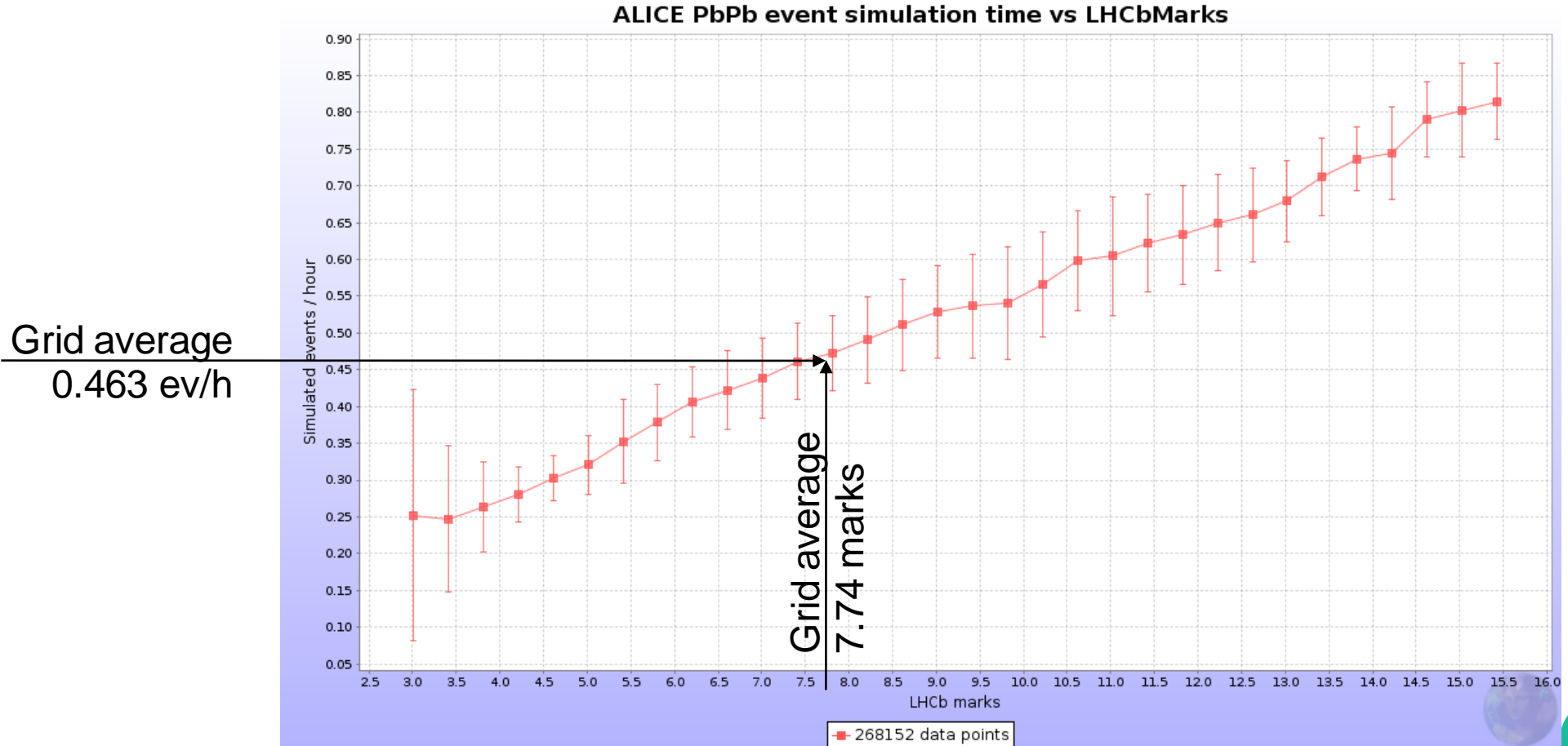
- Simple *python* script
- ~1 min to run
- Used to estimate how many events the job will be able to generate in a fixed amount of time
- Very good results on the CS machines



Grid-wide results

- **Best correlation so far!**

- 270K results, 15K nodes, 109 CPU models



CPU model performance

Trieste

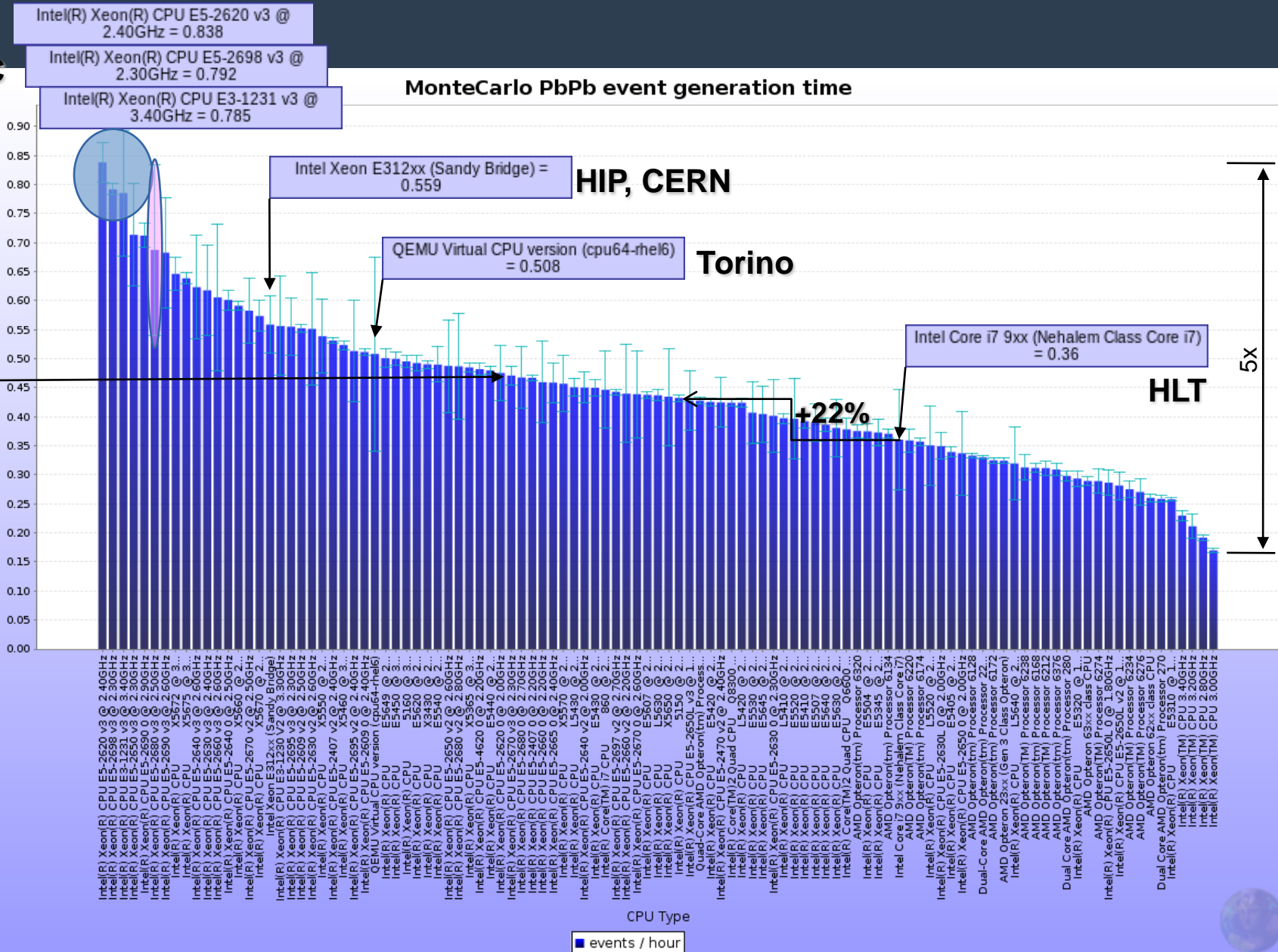
NERSC

KFKI

MonteCarlo PbPb event generation time

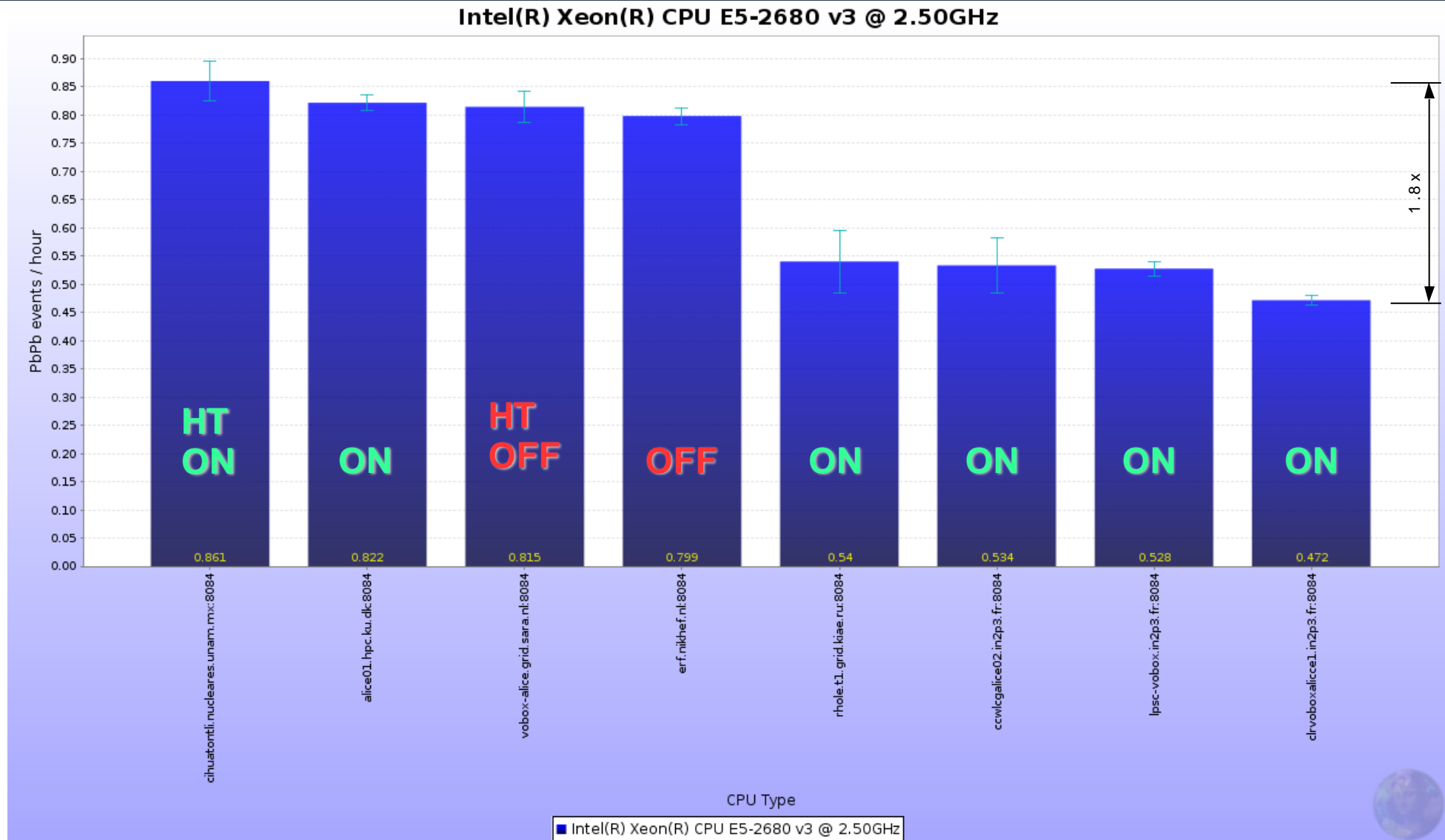
Grid
avg

PbPb events / hour



Site-specific configurations

HT on/off, mem type, #of slots / machine, ...



To do next

- **Settle on one of the fast benchmarks**
 - Preferably a common one with other VOs
- **Planning to build a per node database of benchmark results**
 - Current one based on CPU model
- **Use one of the benchmarks as precursor to (random) jobs to build it**
 - And/or use the [Machine/Job Features](#) project to get the value, if available
- **Final goal is to account in this unit**

Your thoughts here :)

-
-
-
-
-
-
-
-
-
-