Benchmarking in WLCG

Helge Meinhard, CERN-IT

HEPiX Fall 2015 at BNL

16-Oct-2015

Background

- Need for CPU benchmarking well established
  - Resource requests, pledges, installed capacity, accounting
  - Reference for procurement
- HS06 pretty well established and recognised
- Some latent feeling of uneasiness
  - Mostly by experiments, less by sites

GDB Discussion 09-Sep-2015

- Series of discussions and reports in WLCG bodies
  - Grid Deployment Board (GDB) and Management Board (MB)
  - Mostly status updates by providers (Manfred, Michele, HM)
- 09-Sep-2015: GDB discussion taking it the other way
  - https://twiki.cern.ch/twiki/bin/view/LCG/GDBMeetingNotes20150909#CPU_Benchmarking
  - Machine-job features
  - Benchmarking: contributions by the LHC experiments
    - Long discussions highlighting uncomfortable feelings
      - A number of areas to be improved
      - … even though for some issues it may be the communication!
    - IMO a little confused
  - Attempt to structure… (Manfred Alef, Ian Bird, Michel Jouvin, HM, …)
- 15-Sep-2015: MB discussion on the follow-up
  - Four areas for follow-up identified coordinated by a small group

Area 1: CPU Power Seen by Job

- Predict power of compute slot (batch, cloud) for the running job
  - Often needed for job matching and masonry
  - Two approaches:
    - Use HS06 (via MJF) – possibly underestimate because of advanced CPU features
    - Determine on the running job – possibly unprecise if workload changes
      - Needs to be fast and to require access to job slot only
      - HS06 clearly inappropriate – takes hours, requires licence, expects access to full machine
      - Known candidates: LHCb Python script, ROOTmark, Drystone/Whetstone, KitValidation, HTCondor benchmarks, …
      - Some work done, but not conclusive yet
      - Ideally one single choice for all experiments and application types

Area 2: Whole-Server Benchmark

- Benchmark precisely a whole farm
  - Needed for procurements, pledges, installed capacity, CPU accounting, …
  - Requires (possibly long-running) benchmark programs controlling the full machine
  - HEP-SPEC 06 emerged from common WLCG/HEPiX activity back in 2007/2008
  - No known issue with HS06 per se
    - No evidence of scaling issues beyond 10% - the initial objective
    - Choice of boundary conditions for running HS06 has served community well
  - Applications, machines and industry-standard benchmarks have evolved since
    - Should move forward to a new benchmark soon, following proper verification against typical experiment applications
    - Candidates: (Subset of?) new SPEC CPU benchmark suite (expected to be released soon), Geant4 benchmark, …

Area 3: Accounting

- Acute or latent suspicion about accounting numbers being inaccurate
  - CPU time used times slot power in HS06
    - HS06 of machine divided by number of slots, or
    - Average HS06 per slot of a whole compute farm or CE
  - Increasingly inaccurate due to increased machine sophistication – factors including
    - Symmetric multi-threading / hyper-threading
    - Turbo-boost
    - Virtualisation
    - …
  - Exactly the same reasons why (1) and (2) are potentially very different!
- Check whether this is the only source of discrepancy (and unhappiness)

Area 4: Storage and Transport

- … of benchmarking information
- Current attempt: Machine-job features
  - Definitely the right direction
  - Deployment is easy (still risks to take long due to chicken-egg situation)
  - Needed at least for precise estimate of lower limit of job slot performance, and for proper per-job accounting

MB Decision

- Quite some expertise around
  - … and even way more diverging views ;-)
  - Co-ordination and planning needed
- Establish a small group mandated to plan concrete steps to tackle issues (1) to (4)
  - Include all LHC experiments, selected site reps and benchmarking and accounting experts
  - Report back to MB (and GDB)
  - Subject to MB approval, kick off activities around issues (1) to (4) with clearly defined objectives and target dates
    - In particular for (1) and (2), collaborate with HEPiX and their benchmarking experts
- Group now in the process of being formed

Benchmarking WG

Michele Michelotto – INFN Padova

Two separate problems

- A benchmark to understand the potential throughput in terms of events/sec that a Computing Worker Node can produce
- Fast Benchmark: A program that try to "guesstimate" how many event can be produced in a time slot of batch node or on a Virtual Machine

The HS06

- A benchmark to understand the potential throughput in terms of events/sec that a Computing Worker Node can produce
  - It is needed for procurements and tender
  - It is used also to describe the potential throughput of a cluster of WNs, of a Tier2, Tier1 etc…
  - So it is used for CPU power pledges

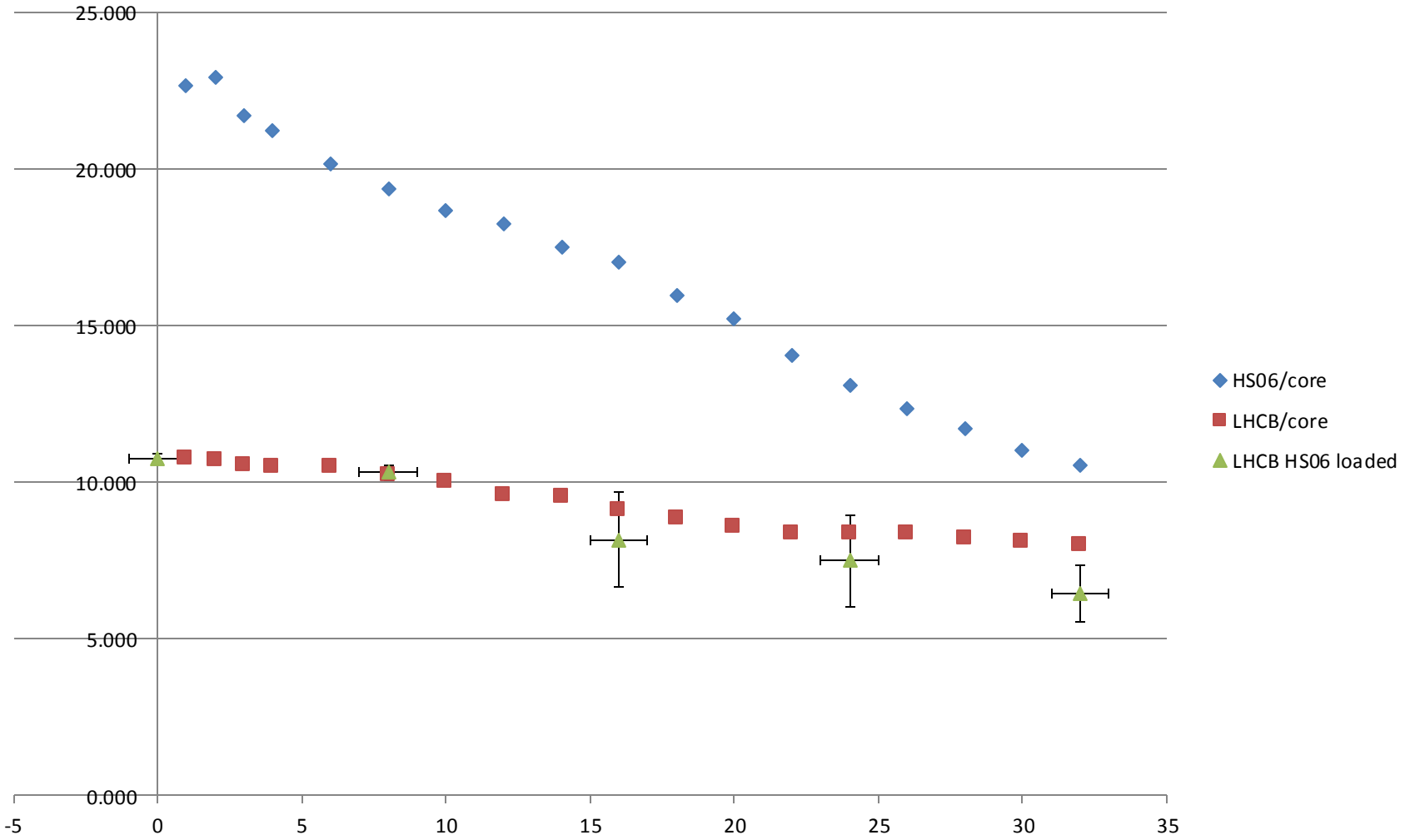
The HS06

- This benchmark was SPEC CPU 2000 and since 2006 is Hep-Spec06 (aka HS06)
  - Long time to run, because it runs only once in the life of worker node
  - few percent precision, aiming to correlate with Full Simulation but HS06 came up in good agreement also with Reconstruction and other applications
  - Easy to run for computing vendor
  - Must be maintained
  - Must be stable
  - Must be free or at least cheap
  - Need a clear recipe to define it (compiler, optimization, 32/64, etc) it because is a unit of measurement like a pint

Fast Benchmark

- Request mainly from WLCG community via GDB to have a fast benchmark
- Users want to know about the performance of the provided job slot
- They tried to use HS06 on that machine to estimate it but there are several factor…
  - Should they take HS06 for the WN and divides by logical core or by physical core? (20 job slot for a 20c/40t WN?)
  - Some sites do a little overbooking, e.g. for a worker node with 20 co/40t they define 30 job slots.
  - The HS06 doesn't describe the actual load on the Worker Node, because of dynamic frequency scaling of CPU clock (clock throttling) but also because of competition for other hardware and software resources on the worker node

HS06/core and LHCB/core

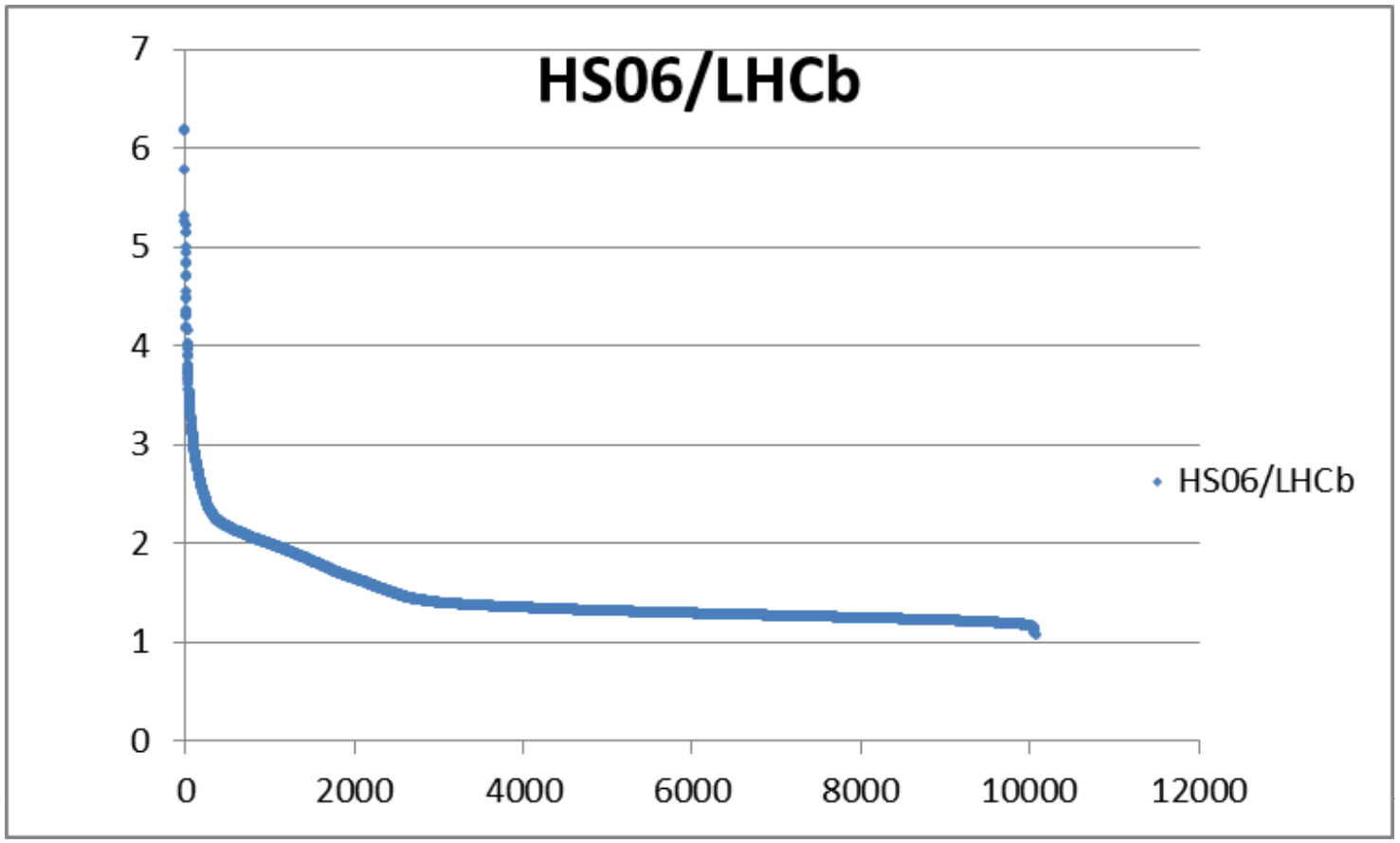


Fast Benchmark

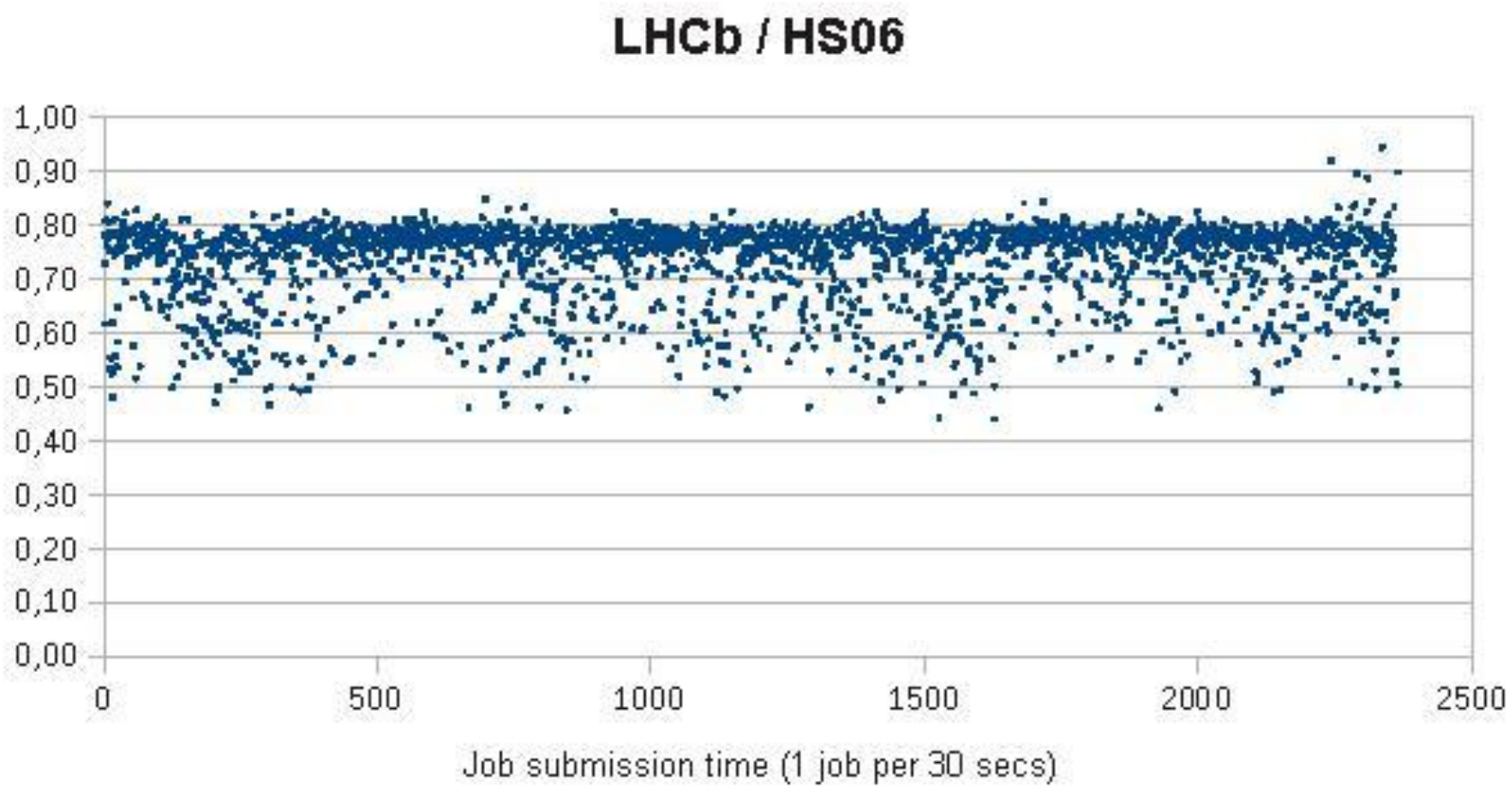
- The fast benchmark runs for one to few minutes instead of hours
- It measures the performances of the provided job slot (on batch farm as well as in cloud environment) and takes in account the actual load of the job slot (single threaded)
- It takes in account the load in those few minutes while it is running but the load may be better (waste of cpu time slot) or worse (job is aborted) when the actual job will be running
- Sometimes the WN can be in some bad condition so the fast benchmark will give a very poor result, uncorrelated with HS06/slot
- Running on bare metal or virtualized could make a small difference (big differences if you use special instructions available only on real machines)

Sometimes LHCb gets slow

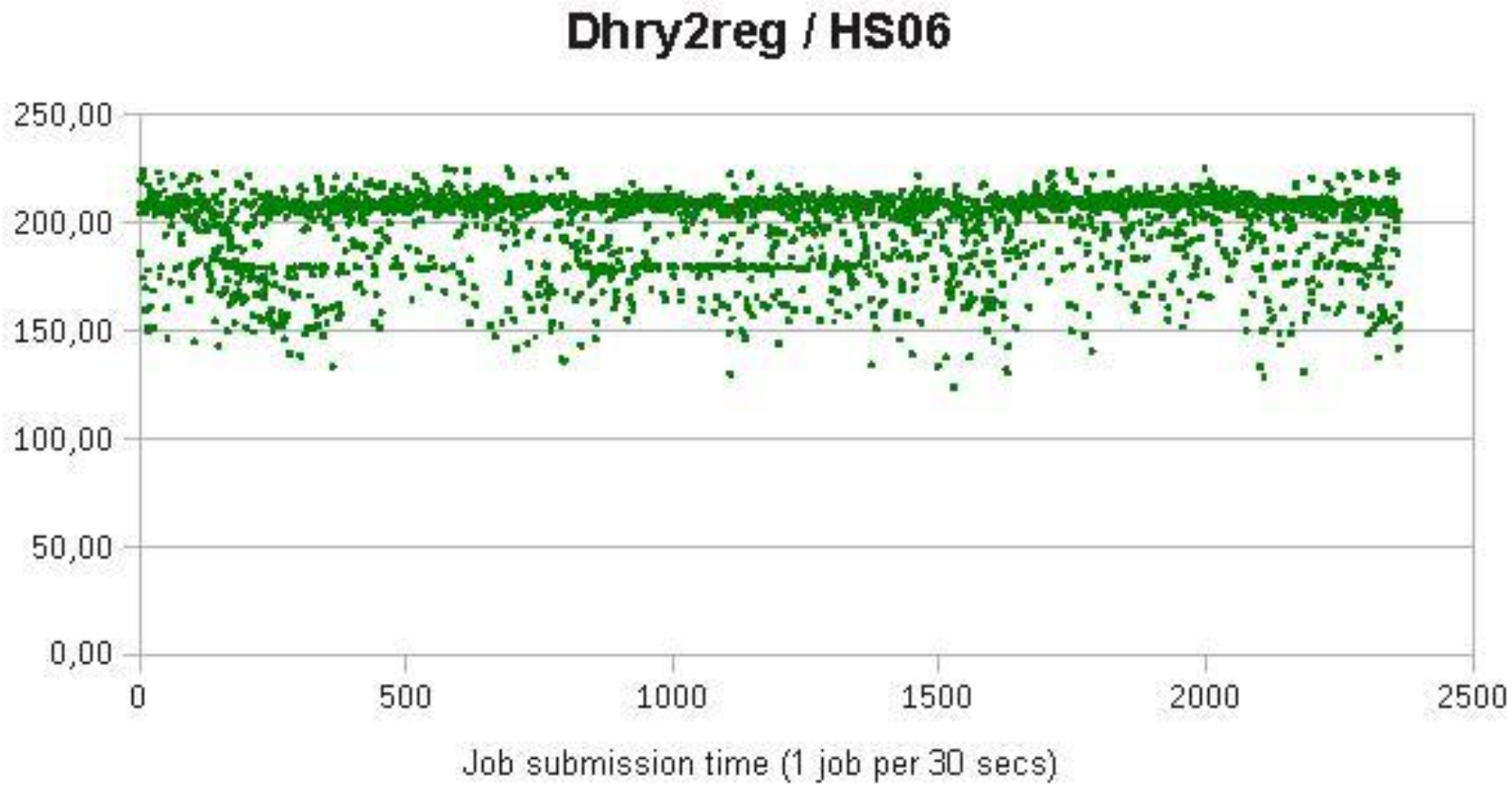
- HS06/LHCb score is around 1.2 – 1.6
- Occasionally is can go to more than 2.0



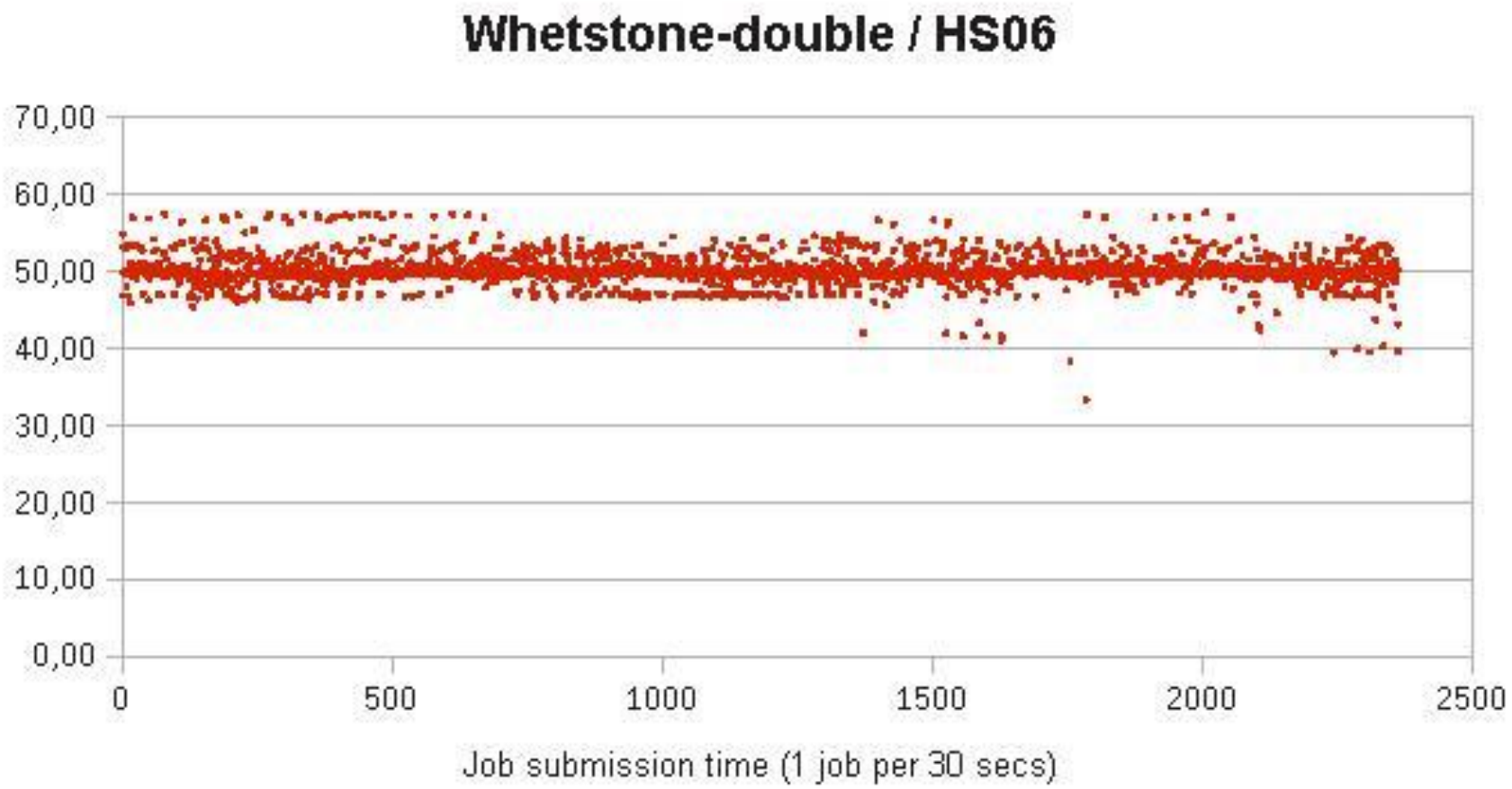
HS06 vs LHCB


LHCb / HS06
1,00
0,90
0,80
0,70
0,60
0,50
0,40
0,30
0,20
0,10
0,00
0
500
1000
1500
2000
2500
Job submission time (1 job per 30 secs)

Other fast Dhrystone

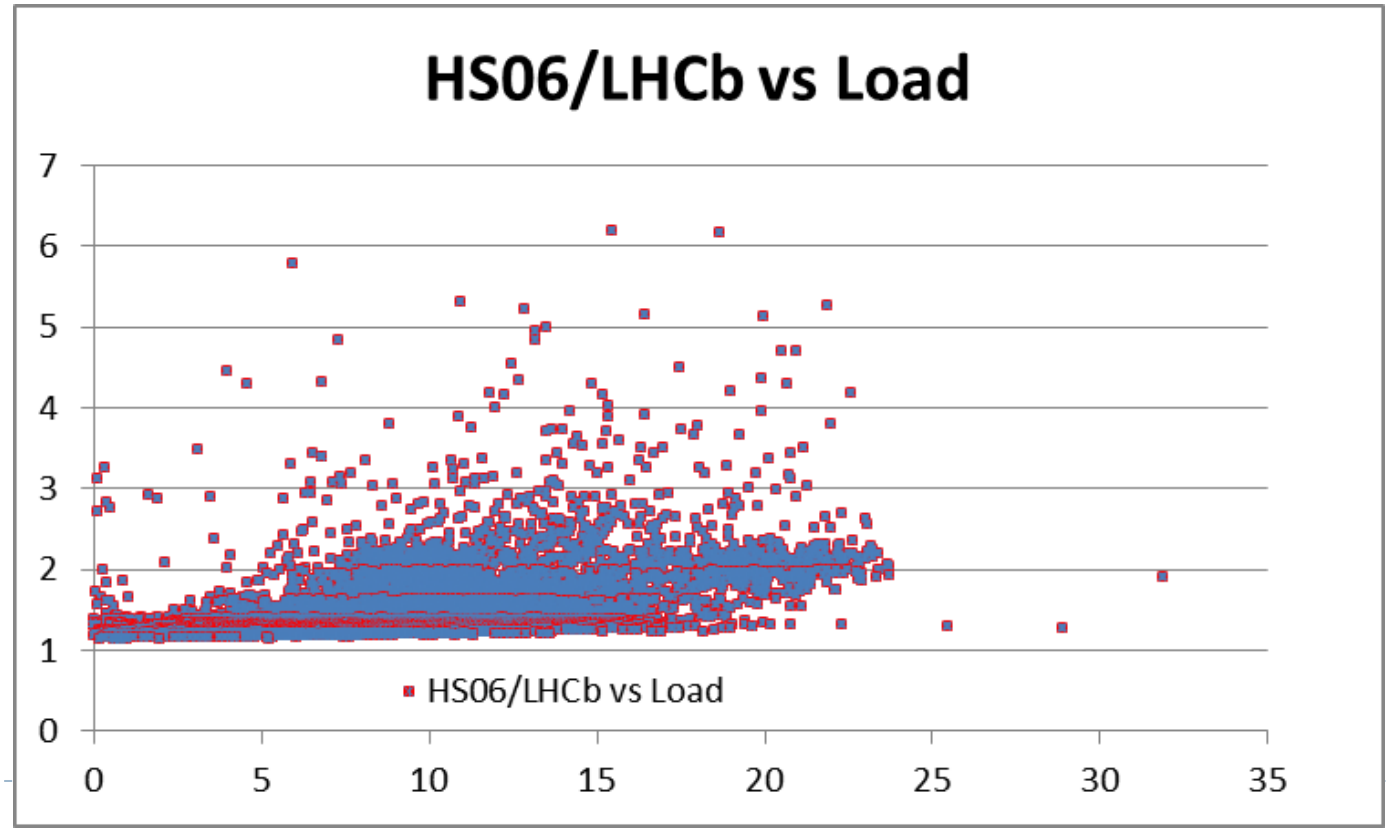


HS06 vs whetstone-double



Measuring on a production cluster

- Manfred measured on a GridKa production cluster the LHCb.py score compared with the HS06 per slot
- HS06/LHCB vs load



Future of HS06

- In the past SPEC was forced to change the SPEC CPU benchmark very often 1989, 1992, 1995, 2000, 2006.
- Now the core is rather stable. Increase in performances mainly from more core/processor. Attention to other issues like Power Consumption
- SPEC is working since several years on the next CPU version tentatively call v6.
- Rumors of a public release in 2014 were too optimistic
- Trying to revive the HEPiX Working Group but it was too early. When the next version of SPEC will be released we will need to check again with the experiments and the community of experts.
- Out of the records…