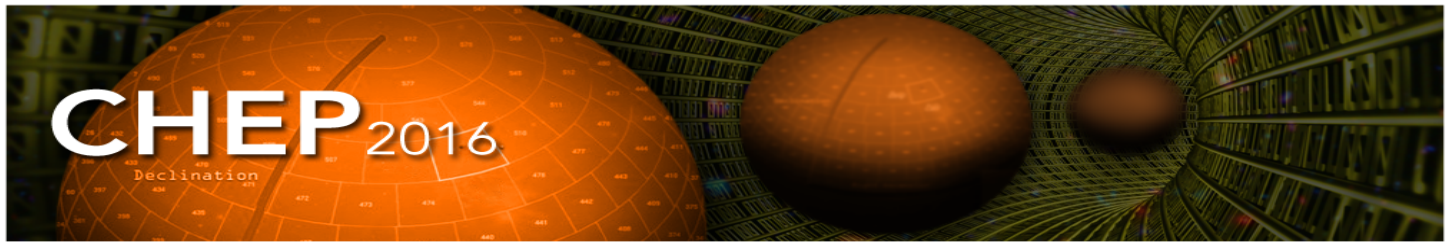


22nd International Conference on Computing in High Energy and Nuclear Physics, Hosted by SLAC and LBNL, Fall 2016

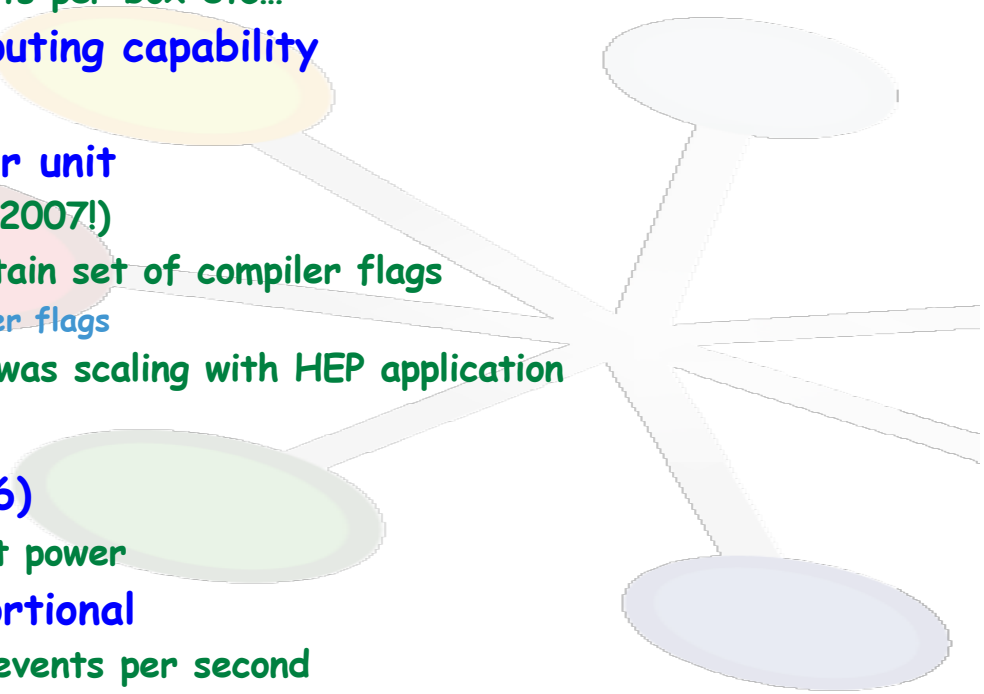
Benchmarking using LHCb production jobs

Ph.Charpentier

LHCb - CERN

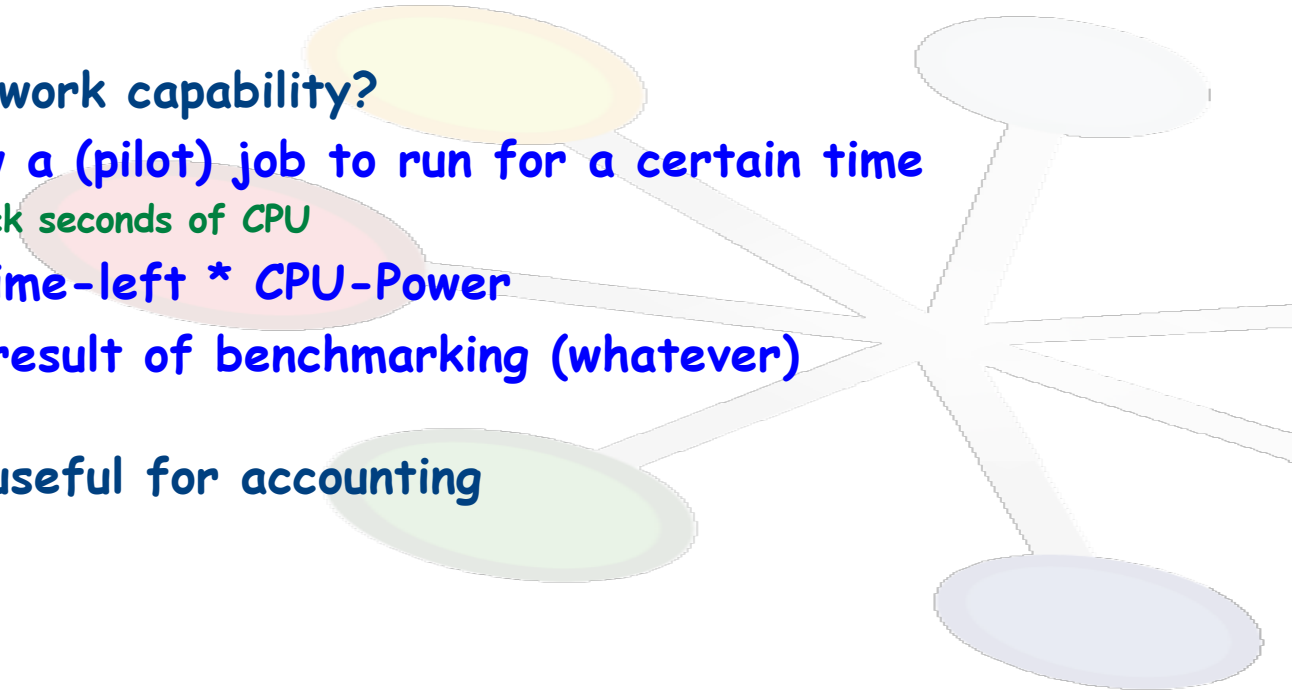
What is benchmarking?

- On the Grid, clouds and volunteer computing, a lot of different types of CPUs are used
  - A lot of configurations
    - Hyperthreading, memory, number of slots per box etc…
  - Each computing slot has its own computing capability
    - Let's call it "CPU-power"
  - WLCG relies on HEP-Spec06 as power unit
    - Well defined procedure (but defined in 2007!)
    - E.g. Compiled in 32-bit mode, with certain set of compiler flags
      - Applications use 64-bit, different compiler flags
    - At the time it was verified that HS06 was scaling with HEP application
- Scaling: what does it mean?
  - Run different applications (incl. HS06)
    - On very different setups, i.e. different power
  - Verify that all benchmarks are proportional
    - Benchmark for applications: number of events per second



Why is benchmarking important?

- When a pilot starts on a computing slot:
  - Before requesting a job, make sure it can run to the end
  - Allows to run multiple payloads and make job masonry
- How to compute CPU-work capability?
  - Most systems allow a (pilot) job to run for a certain time
    - Expressed in real clock seconds of CPU
  - CPU-work = slot-time-left * CPU-Power
  - CPU-Power is the result of benchmarking (whatever)
- Benchmarking is also useful for accounting

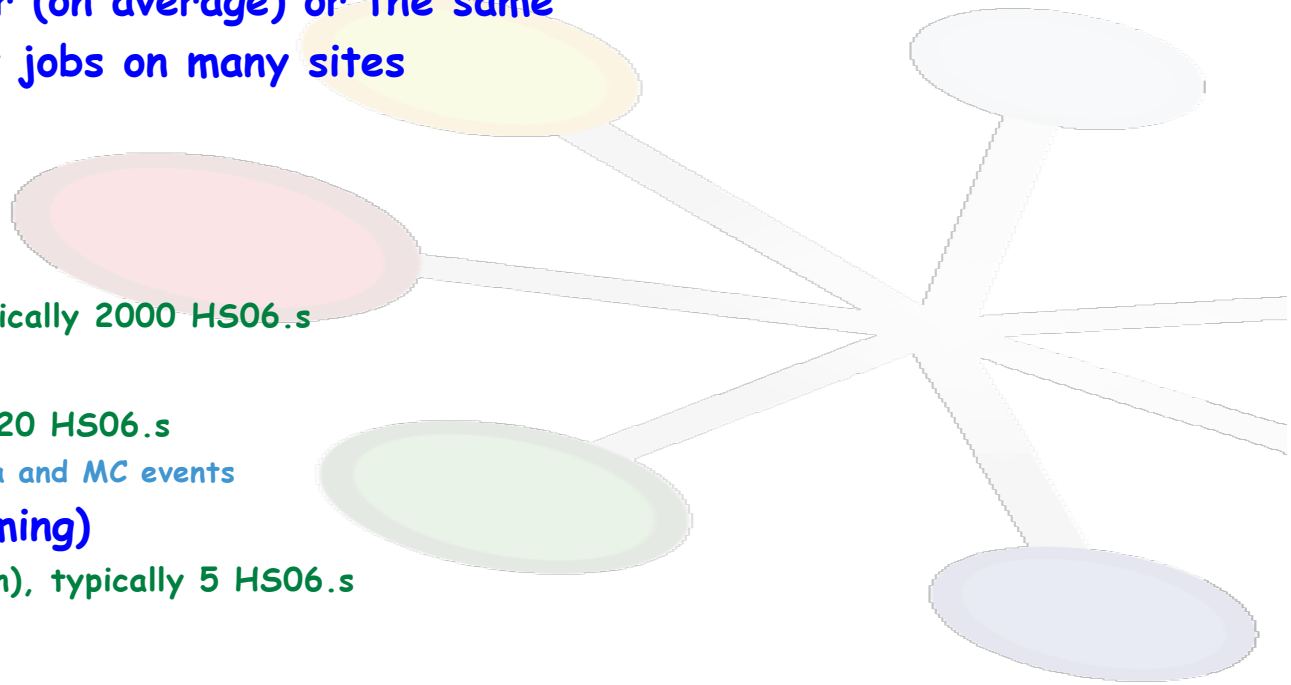


Available benchmarking

- HS06
  - Can only be executed by the site
    - Mandatory to run it in real life conditions
      - Running as many instance as job slots
      - Well defined compiler flags (even if outdated)
  - HS06 benchmarking of a compute slot
    - Depends on the CPU mode, number of processors
    - What is required is a "guaranteed" CPU power
      - HS06 is supposed to be a "worst case"
  - Available through the "Machine and Job Features" mechanism (MJF)
- Fast benchmarking (e.g. Dirac Benchmark DB12)
  - Python random number generation loop (~ 1 minute)
    - Execute several times, keep only last run
  - Absolute calibration as ~HS06-equivalent
    - Doesn't need to be necessarily very precise
    - Doesn't really matter for matching if both measurement and matching uses the same unit!
    - It was re-calibrated in January 2016 for LHCb by a factor 1/0.65 in order to adjust with MJF (DB16?)

Application benchmarking

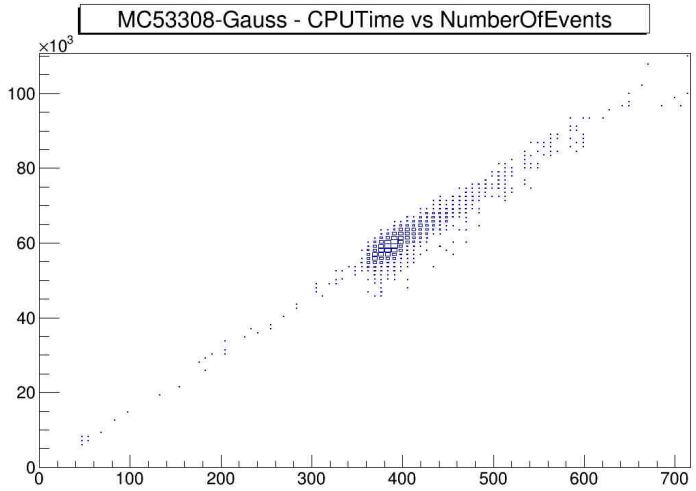
- Application benchmark (JobPower)
  - ~ CPUTime / NumberOfEvents
    - Initialisation + finalisation negligible if enough events
  - Events are very similar (on average) or the same
  - Use productions: many jobs on many sites
- LHCb applications
  - MC simulation
    - Gauss, using geant4, typically 2000 HS06.s
  - Event reconstruction
    - Brunel, between 10 and 20 HS06.s
      - Different for real data and MC events
  - Stripping (a.k.a. skimming)
    - DaVinci (physics selection), typically 5 HS06.s

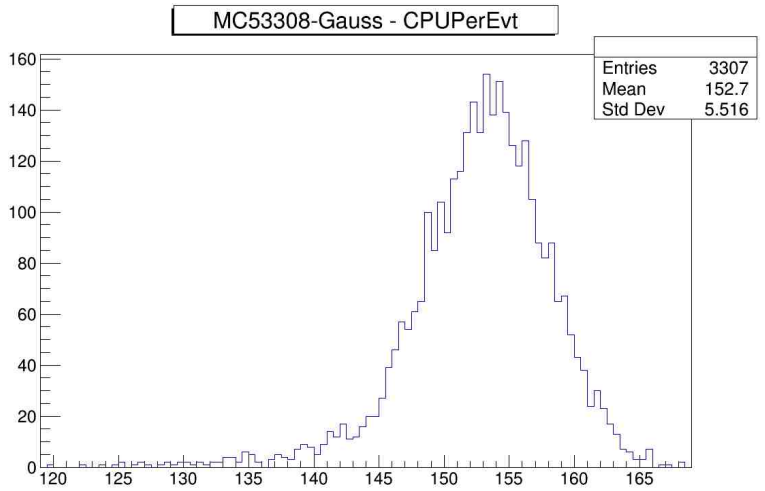


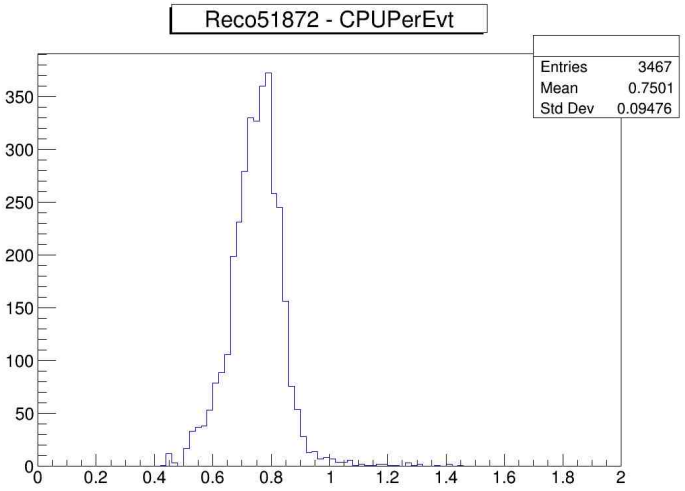
Check linearity of CPU-time with Nb of events (MC and Reco)

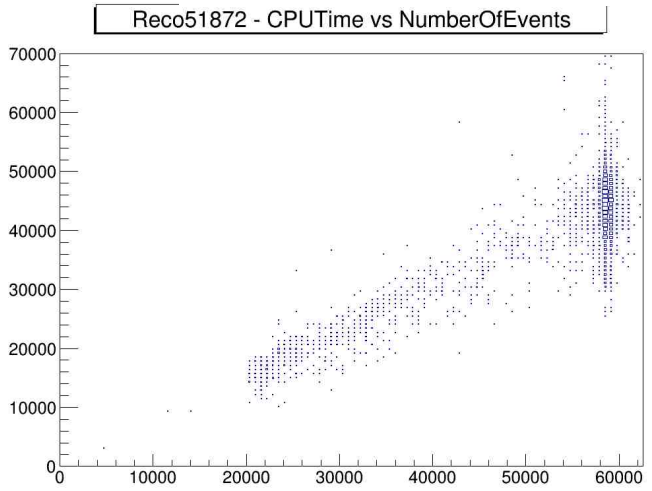
LHCb BENCHMARKING

Simulation

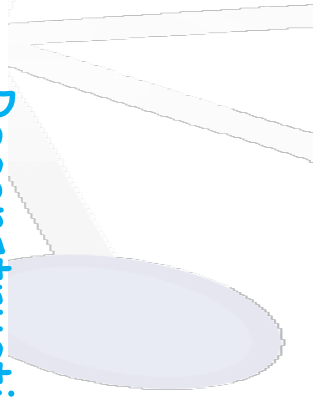




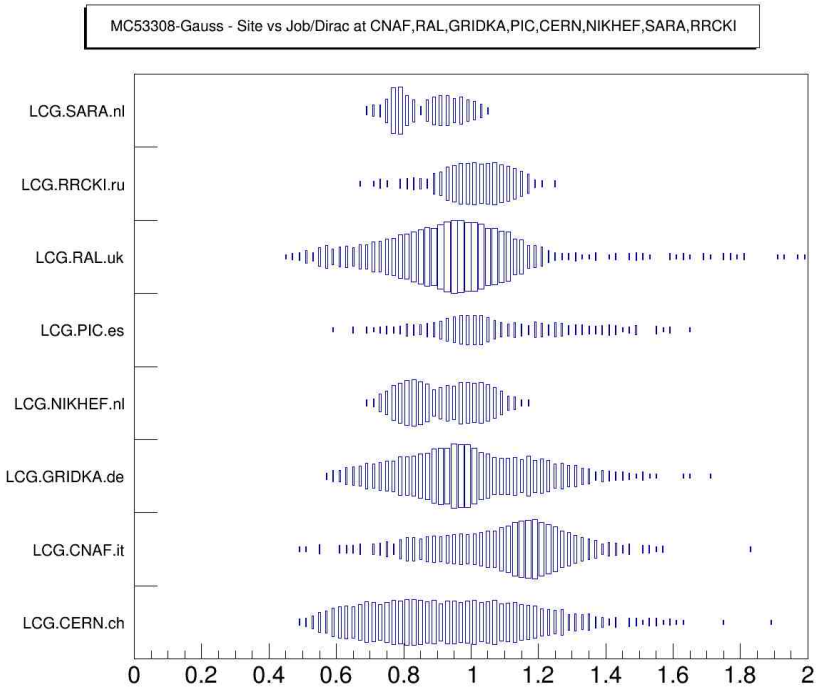




Reconstruction



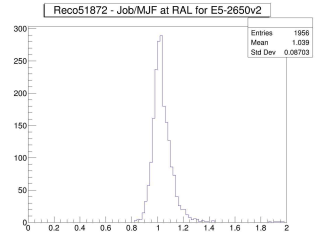
Comparison between JobPower and DB16

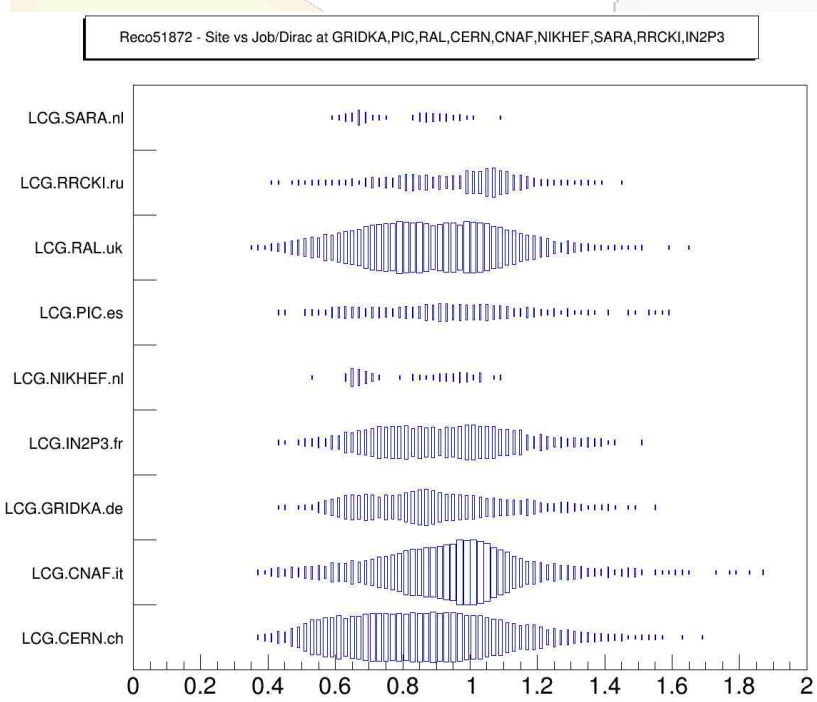


Simulation

- Moderate agreement at site level
  - ±20%

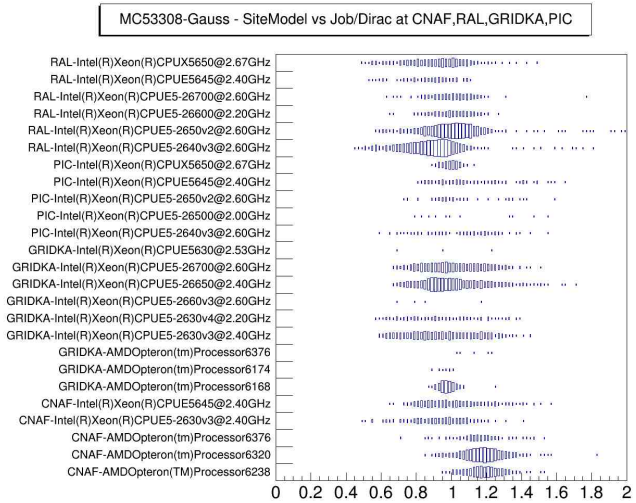
- JobPower absolute normalisation:
  - Set JobPower == MJFPower on CPUE5-2650v2@2.60GHz at RAL

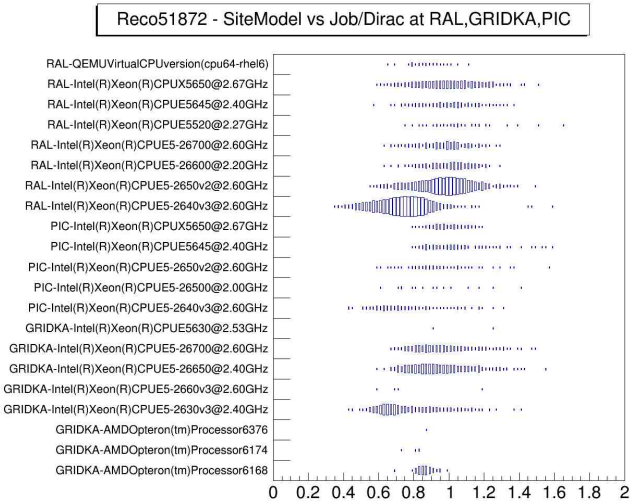


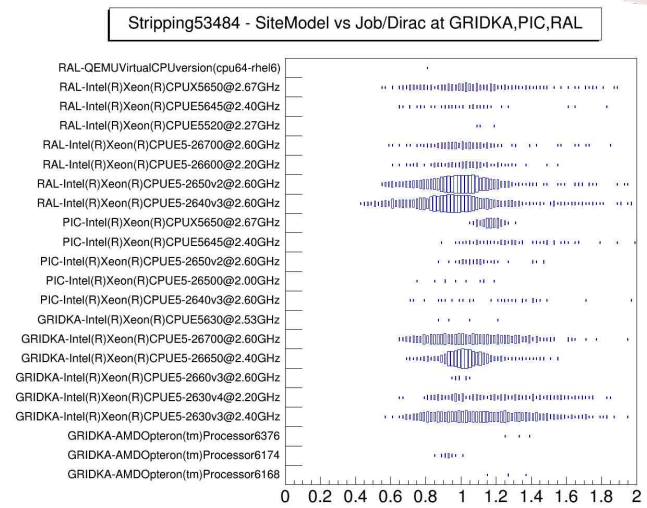


Reconstruction

JobPower and DB16: WN model dependency

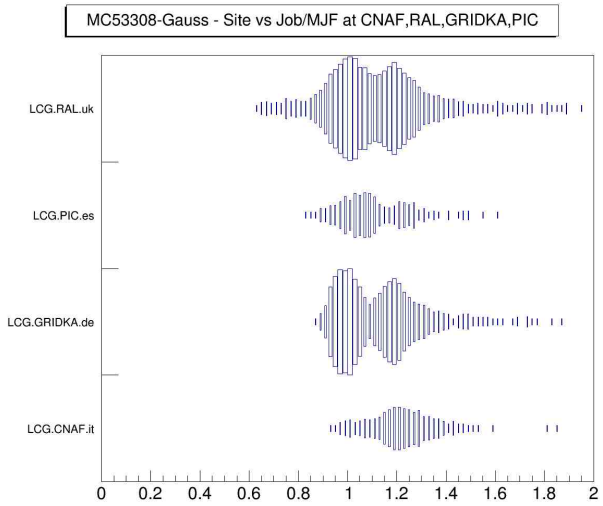




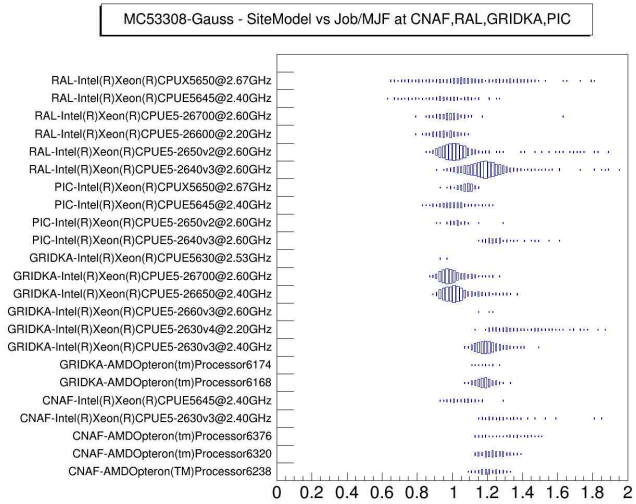


- Large WN model dependency
- Similar pattern for Simulation and Reconstruction
  - Although not quite identical…
  - Slightly better match for Simulation
    - Simulation scales better with DB16

Comparison between JobPower and HS06

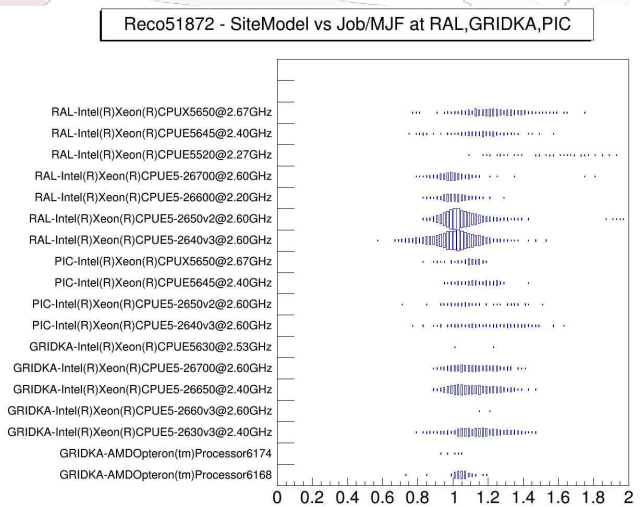


Simulation



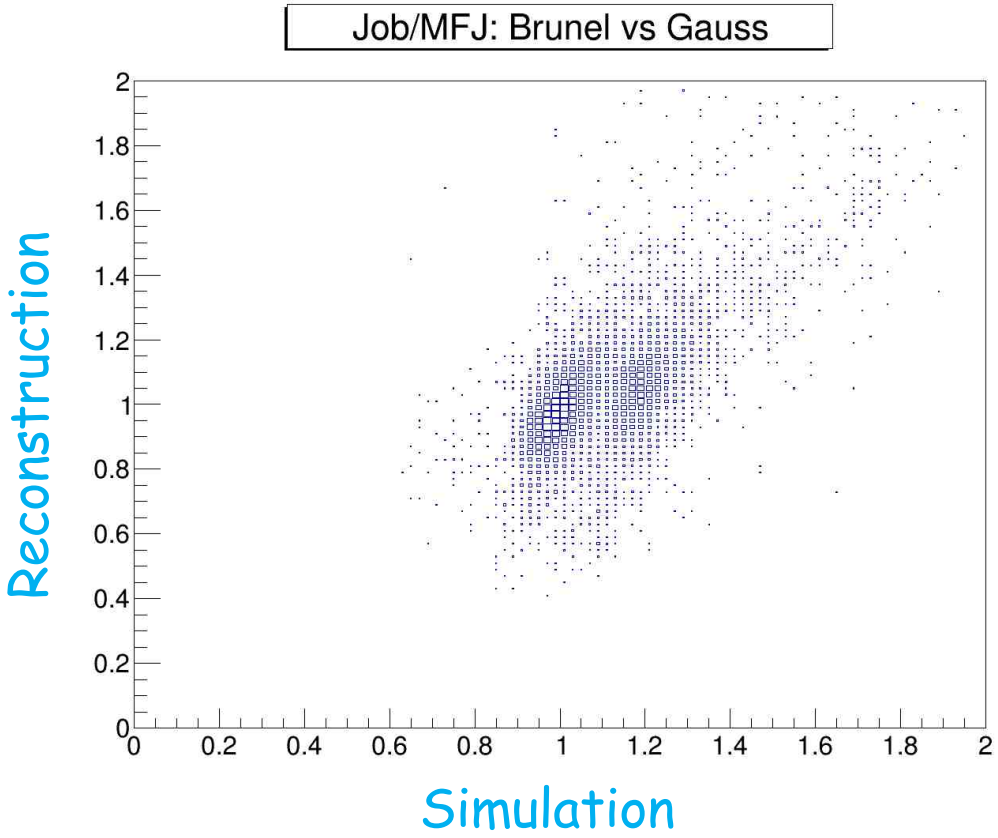
- Not many sites with MJF
- Simulation application
  - Pretty bad scaling
  - Model dependence (bottom left)
- Reconstruction application
  - Much better scaling (bottom right)

Reconstruction



Comparing Simulation and Reconstruction with HS06

LHCb BENCHMARKING



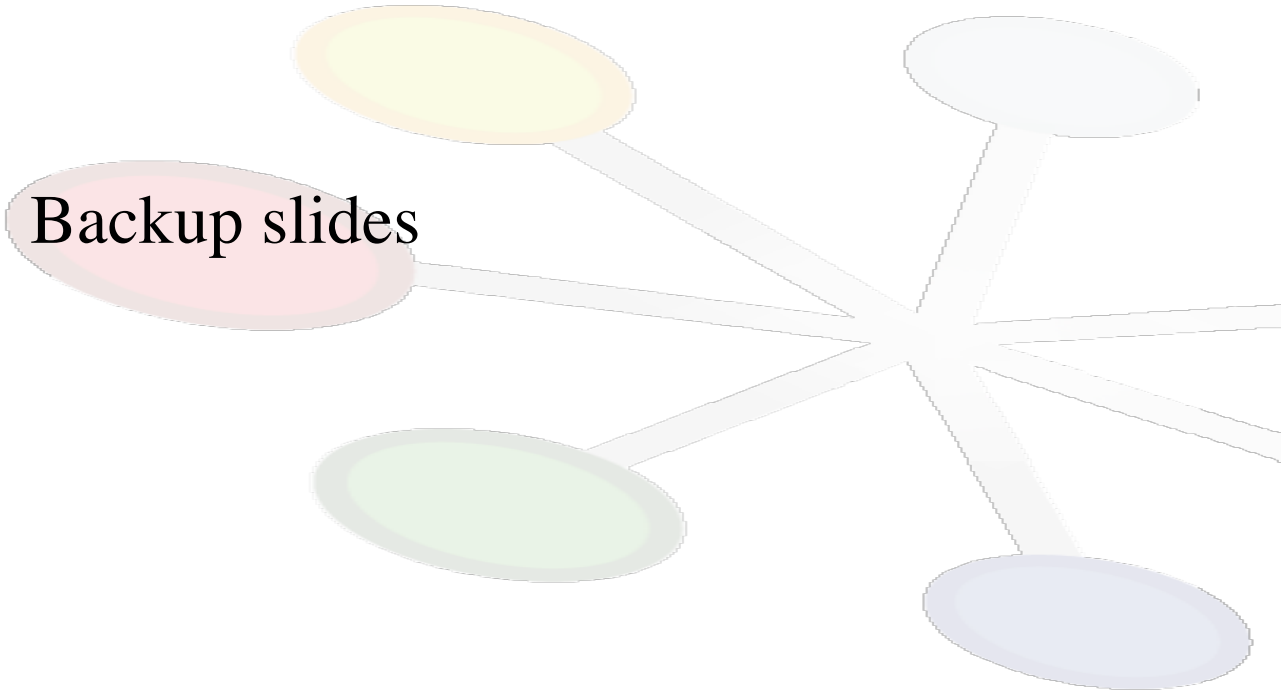
- Comparing within same job
  - No systematic effect
  - Shows a clear difference between Reconstruction and Simulation
  - Reconstruction scales much better with HS06 than Simulation

Conclusions

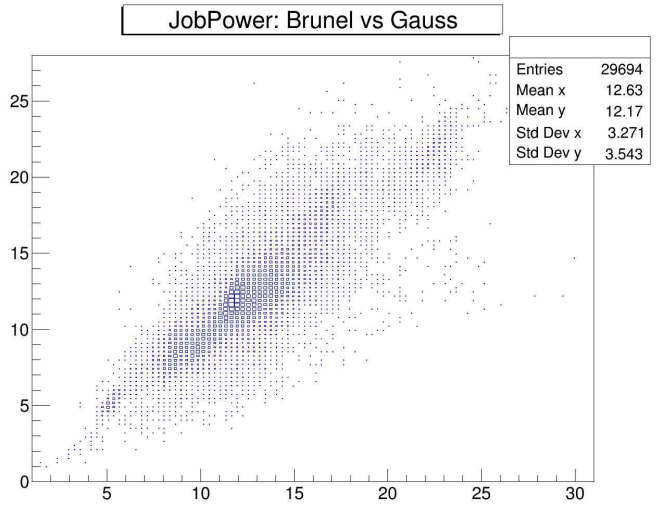
- DB12/DB16 is a fast benchmark, easy to install and run
  - Good enough to be used for matching (±20%)
  - Some WN model dependency w.r.t. applications
    - Take care when evaluating CPU requirements for an application
  - Requires to include a safety margin (20-30%)
- HS06 doesn't scale well with Simulation, but scales very well with Reconstruction
  - Unfortunately not available on many sites
    - Please, deploy!
  - Main difference is CPU / IO (factor 100)
- Site dependencies are large, knowing the WN model is not enough
  - Slots vs processors, overclocking, hyperthreading…
  - Benchmarking is necessary on every set of nodes (from the sites)
    - Fast benchmarking in each job helps a lot

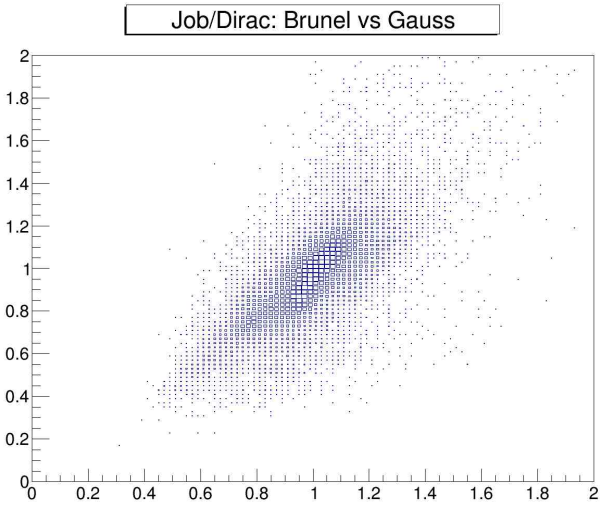
LHCb BENCHMARKING

Backup slides



Comparing Simulation and Reconstruction

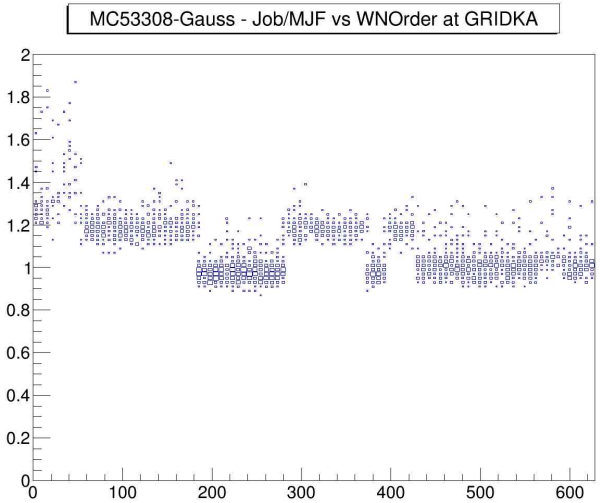


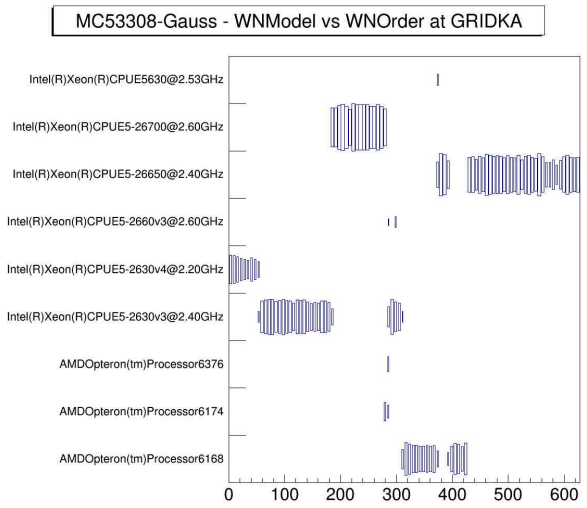


- Top plot
  - Power from event rate
  - Brunel (y) vs Gauss (x)
- Bottom plot
  - Job/Dirac powers

- Each point is from a single job
  - 2 different applications
  - No bias due to WNs
- Not exactly on the diagonal
  - More spread for Reco than Simul

More on WN model dependency





- Job/DB16 for each WN at a site
  - Compare with model per WN (bottom)
- Even for the same Model, large differences in JobPower
  - Probably related to number of slots (and hyperthreading)