



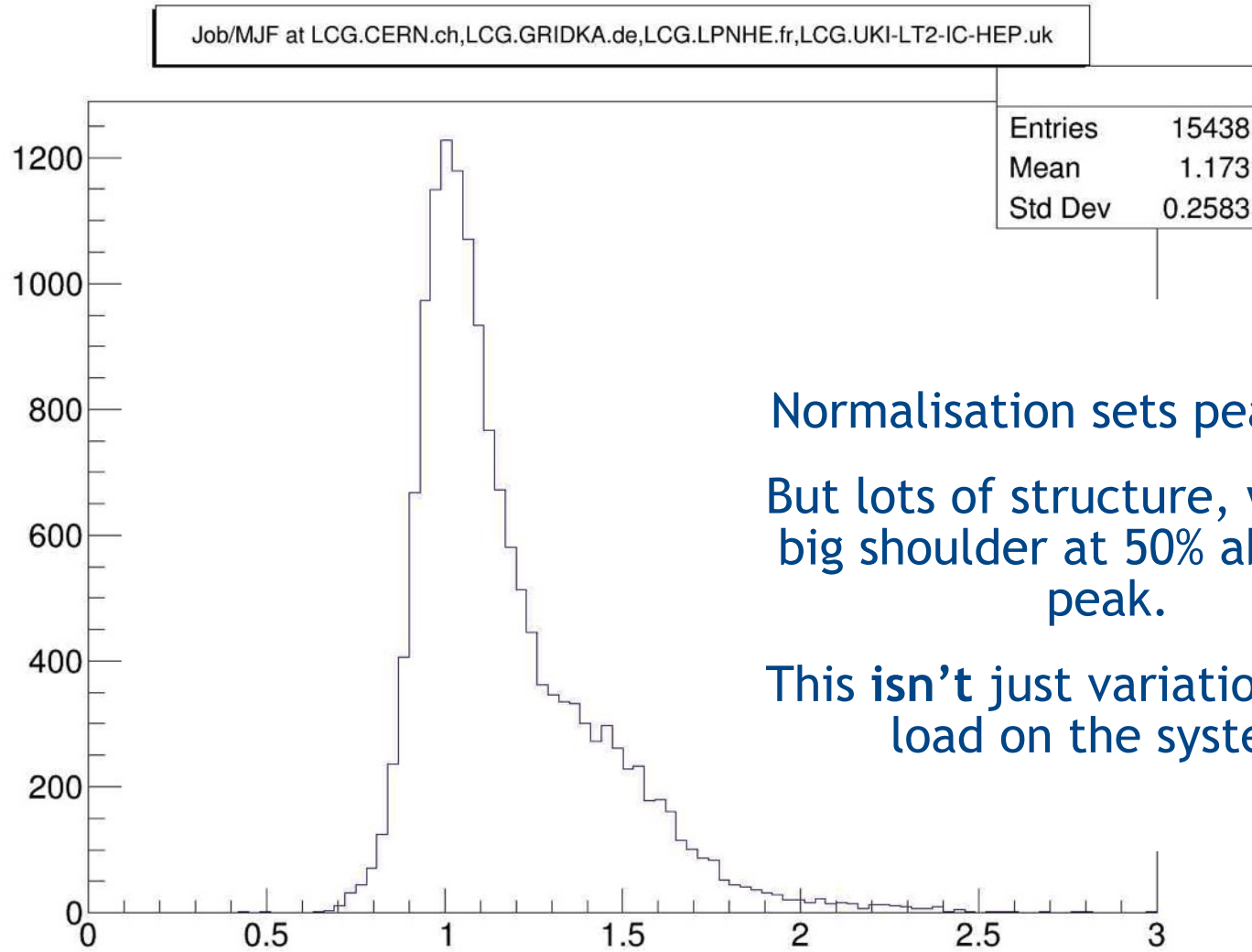
LHCb and fast benchmarking

Andrew McNab
University of Manchester
LHCb and GridPP

HS06, Job Power and DIRAC Power

- Graphs from Philippe Charpentier's comparisons of jobs and benchmarks
 - See Dec 2015 GDB talk for more details of procedure
- Compare three estimates of HS06 allocated to a job
 - **HS06** taken from Machine/Job Features
 - **Job Power** proportional to LHCb Monte Carlo events per second, but normalised to HS06 units using cross-site results
 - **DIRAC Power** uses the built-in DIRAC CPU benchmark, which is normalised to HS06
 - Run at start of each DIRAC pilot: LHCb, Belle II, ILC, LSST, ...
 - Results available to pilot+payload and recorded centrally by DIRAC
 - Very simple: Python `random.normalvariate()` run lots of times
 - Change in required normalisation to HS06 found in 2016

JobPower vs HS06 (from MJF)

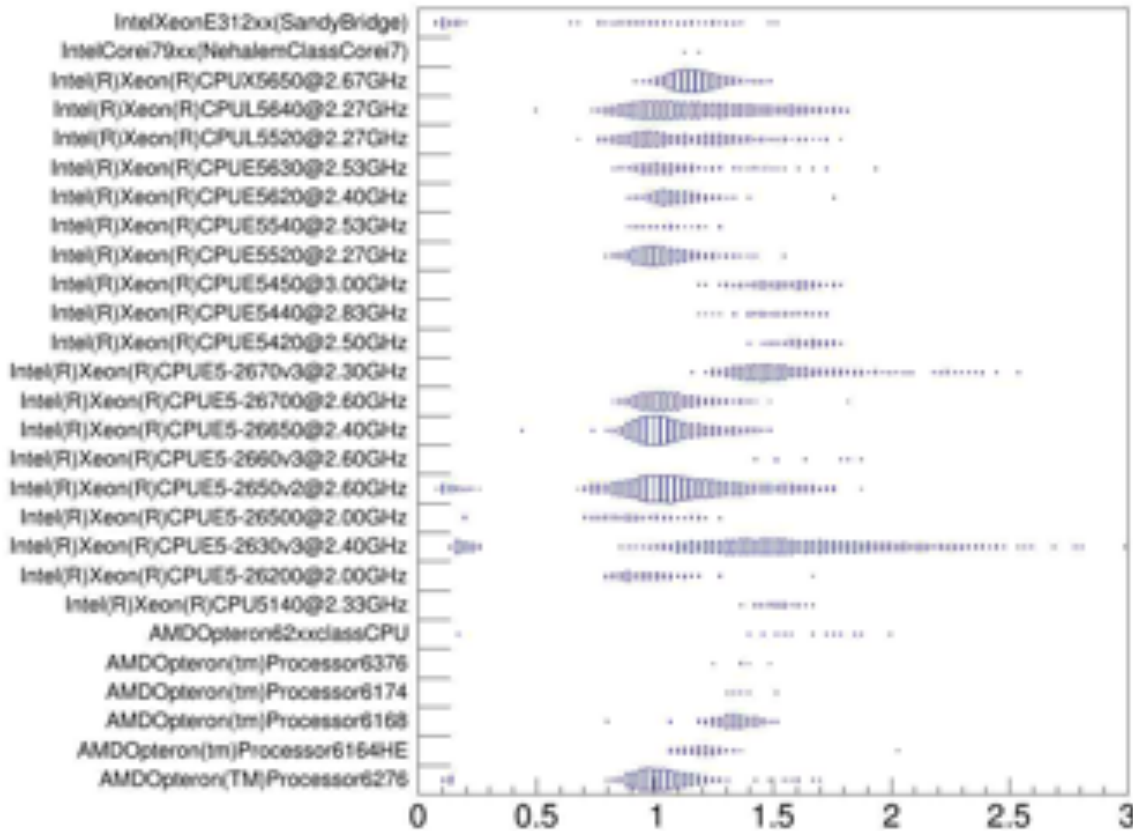


Normalisation sets peak to 1.0
But lots of structure, with that
big shoulder at 50% above the
peak.

This isn't just variations in the
load on the system.

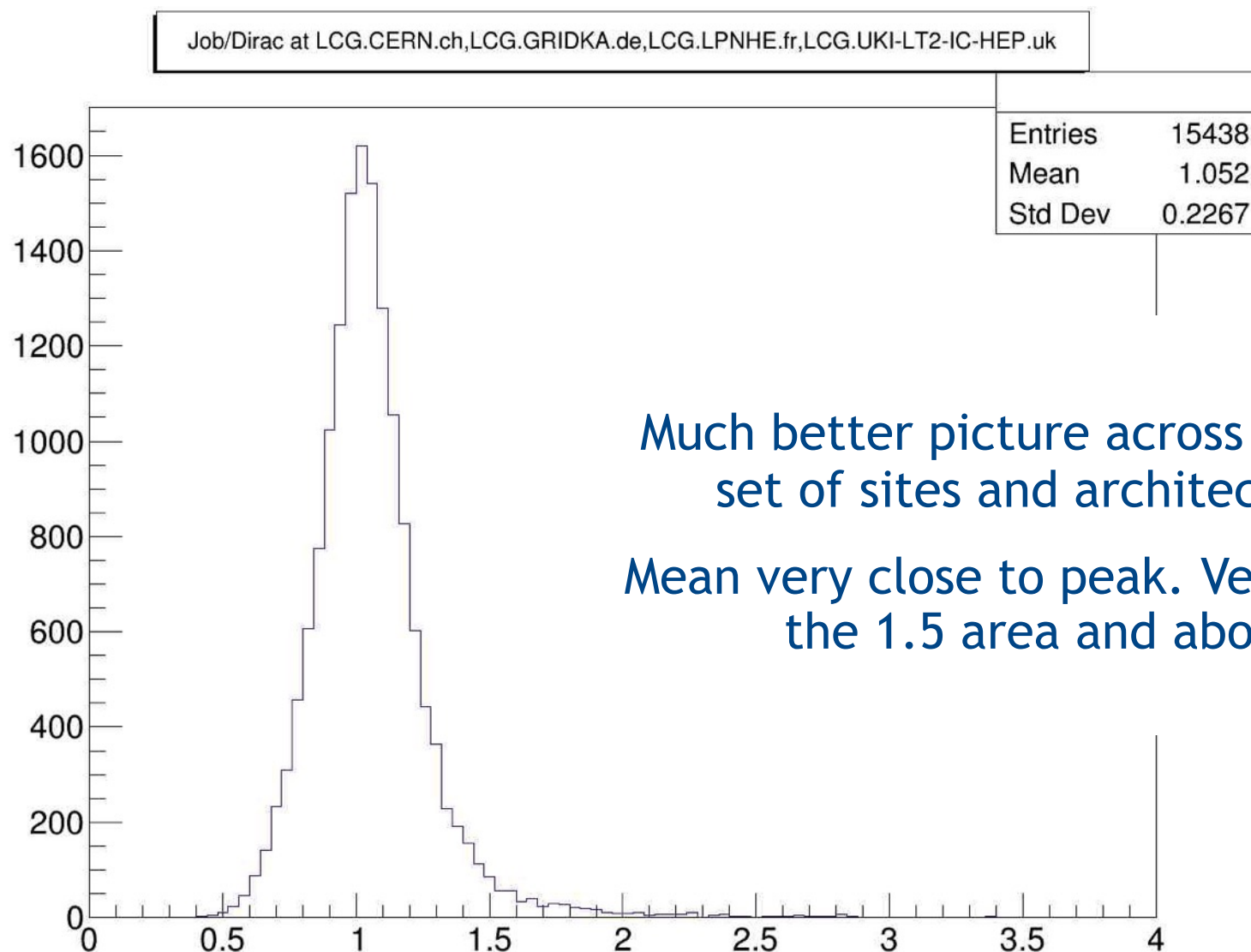
JobPower vs HS06 (from MJF)

WNModel vs Job/MJF at LCG.CERN.ch, LCG.GRIDKA.de, LCG.LPNHE.fr, LCG.UKI-LT2-IC-HEP.uk



Events per HS06
varying by ~50%
depending on
architecture.

JobPower vs DIRAC benchmark



Much better picture across the same set of sites and architectures.

Mean very close to peak. Very little in the 1.5 area and above.

MJF/hs06 vs fast benchmark

- There are arguments for having both
- Fast benchmark at start of pilot/payload gives a true picture of what is really available right now
 - However, other load on the machine could increase and you'll then get less power
- `$MACHINEFEATURES/hs06` and `$JOBFEATURES/hs06_job` give you the worst case that the site is promising to supply
 - These are also the figures recorded in the accounting that you are “paying for” (modulo any cluster-wide averaging)
 - Potentially important if site and experiment accounting disagree

Conclusions

- LHCb is happy with the DIRAC benchmark
- Simple and portable
- OpenSource so can be run within jobs for site validation, monitoring, “job planning”
- Better predictor for Monte Carlo events/second than HS06
 - Still investigating Reco vs HS06 vs DIRAC (but most of our CPU goes on MC anyway)
- Convenient for other projects using DIRAC at WLCG sites
- **We would like to see the DIRAC benchmark included in any WLCG fast benchmarking recommendations to experiments**
- Also included in SL to be run at boot time?
- And communicated via Machine Features mechanism?