Benchmarking in HPC
One person/site's experience

James H. Davenport
thanks to
Steven Chapman, Roshan Mathew (Bath),
Jessica Jones (Southampton)

University of Bath

20 April 2016

Why benchmark?

To ensure that you buy the best-performing machine for the workloads your users will run over the next three+ years!
In an ideal world

- You would know all the codes your users will run over the next three+ years
- (and in which proportion)
- You will get the vendors to benchmark them all on candidate machines
- You compute a weighted sum to give each machine a performance figure

But Your crystal ball may be down for long-term maintenance

How to benchmark?

Not having a time machine!

- we look at the workloads users are currently running

  But these are varied, not well documented, probably not portable

  And the manufacturers aren't going to put in the effort (nor are the majority of the users)

  JHD estimates he gets 1–2 weeks of manufacturer effort for a £1M tender

  Very different if you're a flagship national contract

  So what can/should one do?

Linpack benchmark

Standard	underpins the TOP 500 list etc.
Useful	for bragging rights: probably produces the highest performance number you'll ever see (because manufacturers, compiler writers, library writers all tune for it)
Probably	the only benchmark you have that stresses the whole machine (useful for testing cooling etc.)
However	unlikely to be representative

Being more realistic

What we'd like to measure is not what we can measure.

- ? so you have to extrapolate
- ?? which is a much fancier term than guessing
- – You may not know with sufficient detail what your users' codes are
- – The vendor may be unwilling/unable to benchmark your codes
- + So maybe benchmark a subset
- Or use the LLNL proxy codes: `https://codesign.llnl.gov/proxy-apps.php`
- 2016+ I'd probably use a mixture

Other tips

- benchmark with Turbo off: CPU speed is just too variable with Turbo on
- You can't just "benchmark Ansys" (or Gaussian, or any other big package): the performance varies drastically depending on which parts your users are exercising
- (assuming it's software your users want) insist on being provided with the build scripts as well as the binaries

It's all too easy to end up with "the new version is significantly slower than the old"

Benchmarking your machine

Re-run the benchmarks on your machine as delivered: Bath had some surprises here

- Vendor supplied single-rank DIMMs, but had benchmarked on double-rank DIMMs
- This showed up on our memory-intensive benchmark (VASP): too slow by 10%; also with ANSYS
- Vendor then moved the single-rank memory onto half the nodes (taking them from 64G to 128G) and supplied 64G of double-rank for the emptied nodes (therefore 5TB of extra memory)
- Also discovered a Phi card, one out of four, was under-performing (power supply issue)
- And the build scripts they'd supplied weren't the ones they'd actually used

Slow Nodes

A regiment marches as fast as the most footsore soldier
— Duke of Wellington

Experience says that not all nodes actually perform as specified. This matters for the Wellington reason, so they need to be weeded out pre-acceptance.

Causes	Nodes themselves (BIOS, thermal protection, memory), IB cables, IB switches (but causes generally irrelevant to purchaser)
Method	Run lots of smallish (4-node) jobs and look at outlying run times
If possible	Use scheduler to allocate nodes
else	do statistics on reported node numbers
Jobs	should be memory intensive; CPU intensive; network intensive (Jessica has four different jobs)

Rebenchmark??

DoE	rerun their benchmarks every (10,000 jobs, X000 hours, . . .) to check for performance drift
SC15	had a debate on "how often" — no consensus
JHD	doesn't know of anyone else currently doing this
But	is looking to do it annually at Bath

Conclusions

- Benchmarking is not easy
- Bath spent an elapsed six weeks (1 week JHD time) writing the benchmarking section of the OJEU
- But well worthwhile in terms of the quality of what you end up with
- And actually gets you closer to your users
- And (possibly) to your vendors