

Why do benchmarking?

Procurement Acquisition Acceptance In service planning evaluation testing testing Choosing where to Stakeholder Code tuning R&D run your code buy-in Seeking a Competitor Marketing technology edge analysis



Detail: which hardware?

Processor

Base clock

Dynamic clock settings

Cores

Cache

Hardware threading

Node

Node type

RAM capacity

RAM speed

HBM

SSD vs off-node

Network

Software

Compiler

Math library

OS version

MPI library

Batch system

..

Statistics

Results (metrics) will vary from run to run

Run many times, collect statistics

Best case? Worst case? "Typical" case? 90th percentile?

What about non-normal distribution of results?

Effect of first run / initializations

Massive scope for misinterpretation (or misleading!)

Performance Metrics

Run duration

Rate (e.g., FLOPS)

Scalability

Energy use

Efficiency (e.g., %peak)

Perf/Power (e.g., FLOPS/W)

Perf/Cost (e.g., FLOPS/\$)

Model steps per TCO

Rate of pizza consumption

YMMV

System A clear winner for Code 1 but poor on Code 2
System B clear winner for Code 2 but poor on Code 1
System C "ok" for both codes

... expand to suite of N codes x M systems ...

Are you measuring performance of

- System
- Software layers & tools
- Application code
- Benchmarking/optimizing staff
- •

HPC systems (hw + sw + ...) are now too complex to be definitively measured:

We now diagnose rather than measure

Benchmarks are the tools used for diagnosis

Benchmarks as risk reduction

Reducing some of the risks of the acquisition decision

- Eliminate unrealistic candidates
- Broad relative positioning of candidates
- Understand fraction of application estate will perform best/good enough/poorly on each candidate
- Identify specific performance concerns and investments needed
- Maturity of software ecosystem (or need for investment)
- Confidence in system sizing / performance promises
- etc.

(NOT "we'll pick the system that wins the benchmarks")

Final answer?

Benchmarking is probably the one HPC activity that is simultaneously the most:

Fun

• I'd do benchmarking / technology research full time if anyone paid me to!

Frustrating

Yup.

Useful

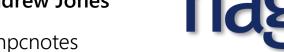
risk reduction, cost optimization, planning of code & people investments, ...

Experts in High Performance Computing, Algorithms and Numerical Software Engineering

www.nag.com blog.nag.com @NAGtalk



International centre-of-excellence in business and technical aspects of HPC **Andrew Jones**



@hpcnotes

www.linkedin.com/in/andrewjones