# *Three Questions You Should Ask*

*William Gropp*
*Mathematics and Computer Science*

# *Panelists*

- Geoffrey Fox, Indiana U
- Bill Gropp, ANL/U of Chicago
- Jeff Hollingsworth, U Maryland
- John Morrison, University College Cork
- Phil Papadopoulos, SDSC

# *What's this panel about?*

- What 3 things drive *you* or your friends nuts about using clusters and/or grids?
- What would you measure to track and encourage progress toward a better experience?
- Not another discussion of what is right or wrong with the state of computing

# What doesn't qualify

- Any thing that isn't quantified or quantifiable
  - "Elegance"
- Don't "measure what you can measure"
  - CPU clock rate
  - Novice programmers writing throwaway parallel programs
- Best-case measurements
  - Don't forget Clarke's third law
    - *Any sufficiently advanced technology is indistinguishable from magic.*
  - And its all-important corollary:
    - *Any sufficiently rigged demo is indistinguishable from magic*
- Best-case tells us what might be possible, it doesn't guarantee that we even understand all of the issues yet

# An Example

- What I really want to know:
  - Will this node run my applications well?
- Best answer comes from measuring applications of interest running on the node
  - Not always practical or possible
- Easier question:
  - What is the STREAM performance?
- Why that question?
  - A good fit for my applications (memory bandwidth bound)
  - Easily described and measured
- What isn’t helpful:
  - What is the hardware memory bandwidth?
  - Why it isn’t helpful
    - *Doesn’t correlate well with measured performance.  Systems with > 10 GB/sec memory bandwidth often deliver 10%*

# *Question 1*

- For Node Performance
  - Measure STREAM, STREAM with a stride of the cache size, and random scatter gather.  Form the vector of performance in seconds/byte.
- Why
  - Many algorithms are memory bound (e.g., sparse matrix-vector multiply, mesh updates)
  - Other two measurements address some of the components of lower-than expected performance
- If you want a single number,
  - Measure the angle between the (1,1,1) vector and the measured performance. Zero is good (from a productivity standpoint)

# Question 2

- Parallel Code Performance
  - Sparse matrix assembly followed by matrix-vector multiply, on q of p processing elements, matrix elements are r x r blocks
- Why
  - Assembly: often a disproportionate amount of coding, stresses expressivity
  - q<p: supports hierarchical algorithms
  - Sparse matrix: many aspects of PDE simulation (explicit variable coefficient problems, Krylov methods and some preconditioners, multigrid); r x r typical for real multi-component problems.
  - Freedoms: data structure for sparse matrix representation (but bounded spatial overhead)
- If you want a single number
  - Min time / max time. 1 is good.
  - Max time can be infinity if not available (e.g., can't handle subsets of processes)
  - Addresses predictibility of performance, sensitivity to changes in algorithm, problem.

# *Question 3*

- Remote Data Access
  - Time to access n bytes in m sessions, for all sessions.  Include the time to install any necessary software
- Why
  - Addresses the perception/reality that grid operations are not reliable
  - Measures what practicing scientists see (the mean case, not the best case)
  - “They concluded that the technology was at best unready for prime time, and at worst total hype. So they decided to hold off. Grid advocates are now in the position of having to get people to change their minds, not just adopt a new idea.” (Raymond Turney, http://news.taborcommunications.com/msgget.jsp?mid=679313&xsl=story.xsl)
- If you want a single number
  - Like the parallel sparse example, min/max time/byte, over sessions.
  - Assign failures (could not get data moved) some penalty value (instead of zero performance), such as time that an alternate method (even sneaker-net) would take
  - If you’ll accept a complex number, then in mod-arg form, the modulus is the maximum performance and the arg is the angle $\phi$ such that $\cos(\phi)$ is the fraction maximum performance that was the worst-case (use secant($\phi$) to make a starker number)

# Summary

- Measured values need to include the variability seen by users
- We like to be optimistic
  - What is the best that we can do
- To get to "it just works," we must understand, and where possible, reduce variability in performance (and success)
  - Experimental science typically reports ranges (e.g., error bars)
  - We (the people in this room) can encourage this by
    - *Setting an example in the papers we write*
    - *Requiring these measurements when reviewing and editing*