HPC Performance Metrics: Should We Drop FLOPS?

International Workshop on Performance Analysis and Optimization of High-End Computing Systems

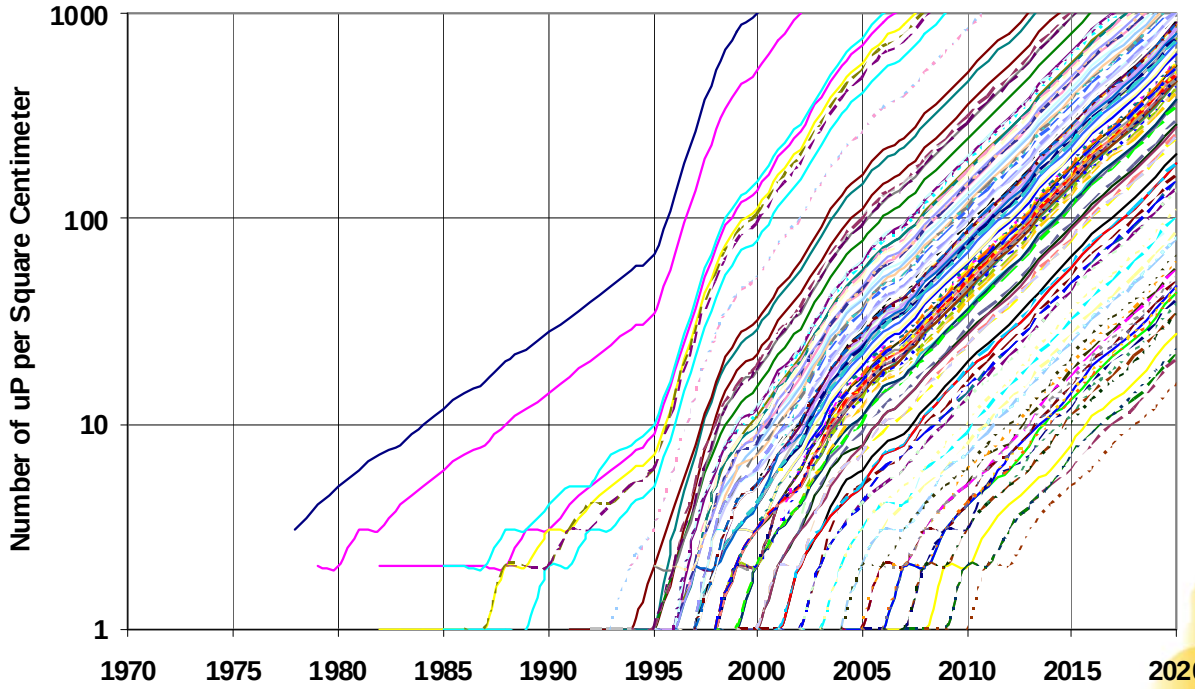
Jeffrey S. Vetter

ORNL and Georgia Tech

http://ft.ornl.gov
Vetter@computer.org

Agreed – PEAK FLOPS are cheap and getting cheaper…

With a *Staggering* Potential



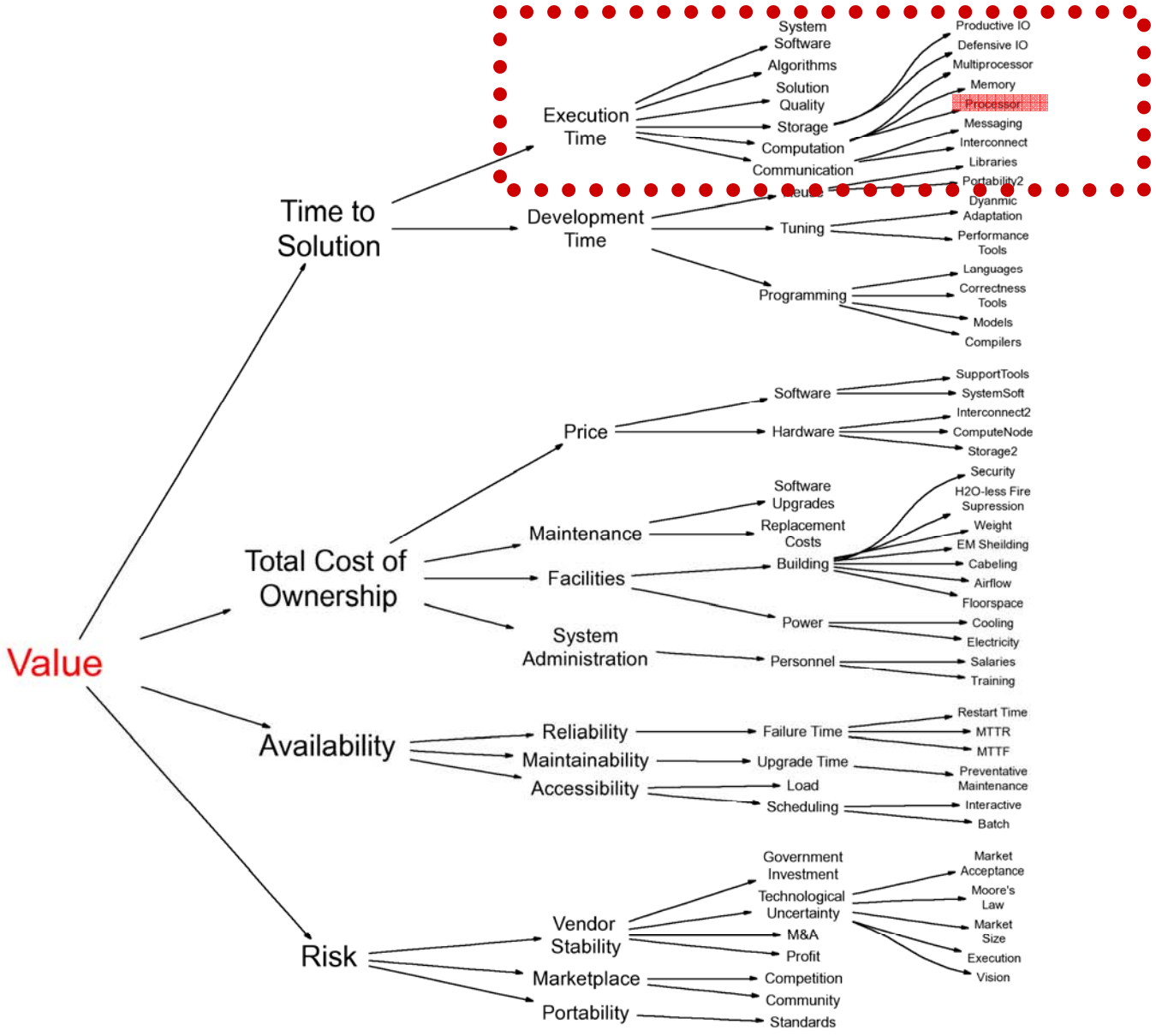
UNIVERSITY OF NOTRE DAME

ENABLING INNOVATION

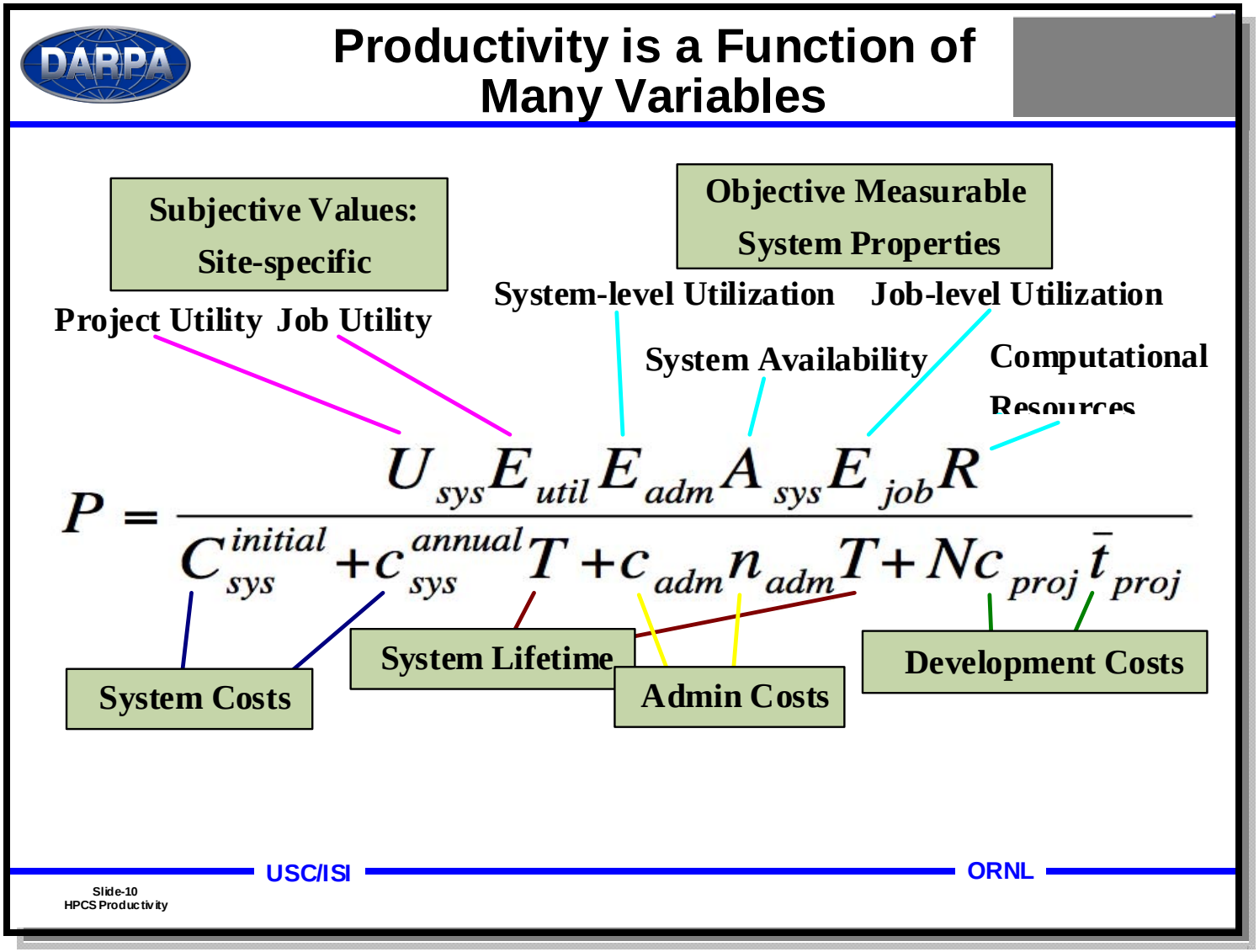
10

Peter Kogge, 2007

Many factors influence Value!



HPCS Productivity Studies

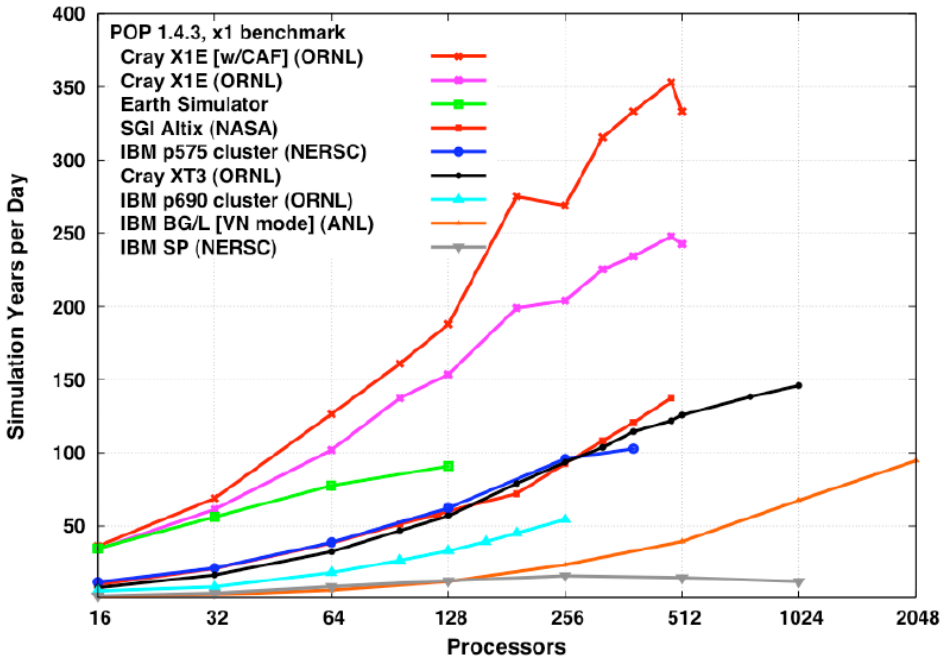


Observations

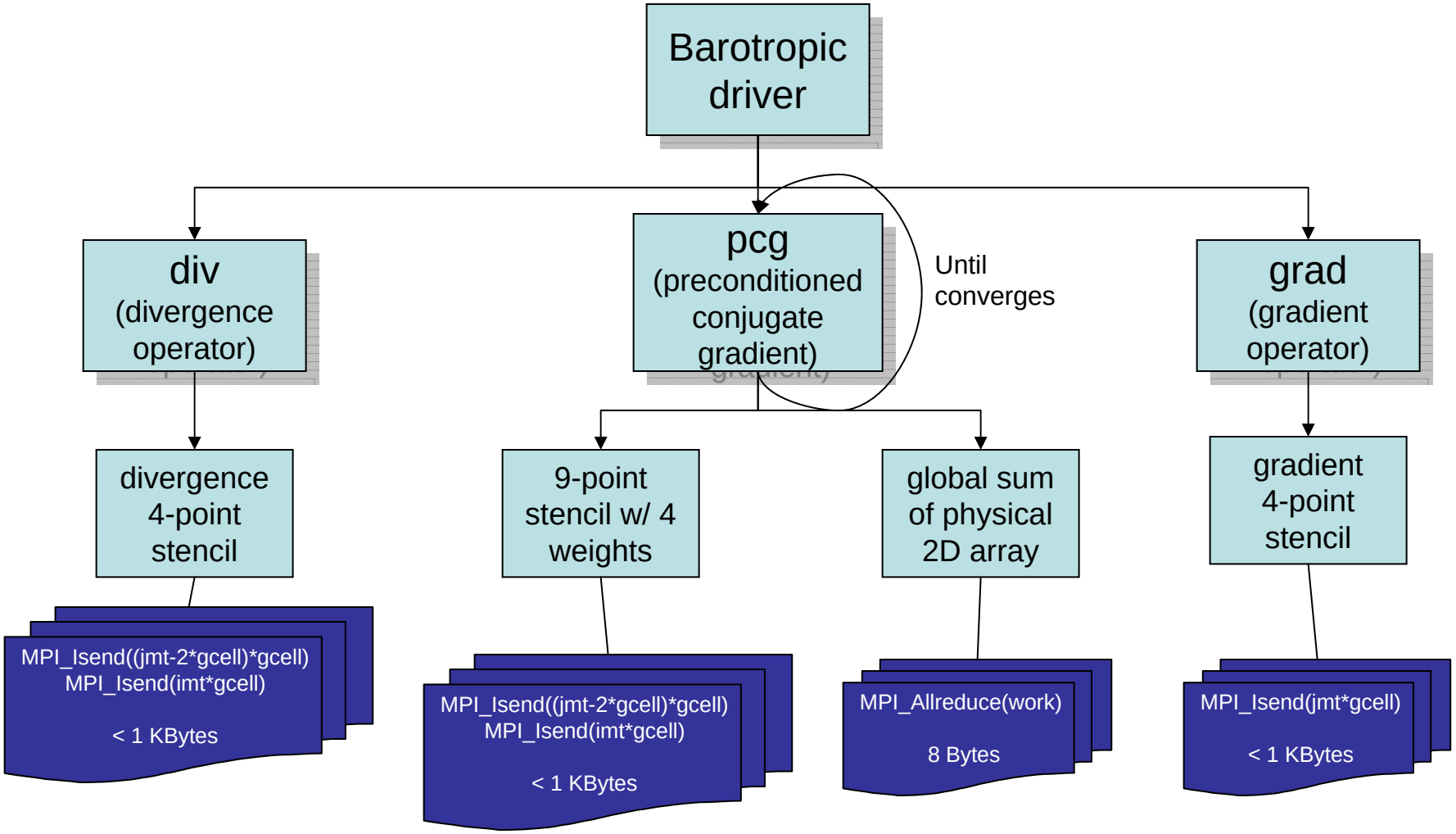
- **At one time, architectures and applications were relatively simple**
  - FLOP throughput was easy to calculate and predict for both architectures and applications
  - The PEAK FLOPs metric remains easy to calculate and to use to scope initial designs
- **Applications and architectures are becoming much more complex**
  - HPC Challenge recognizes more complex criteria
    - Focus on memory (I/O) operations and the storage hierarchy
      - Spatial and temporal locality, yet this can be difficult to measure
      - Package designs and signaling rates are 'relatively' flat
  - Many of today's procurements include performance criteria other than PEAK
    - DOE, DOD, NSF
- **Many other metrics are very difficult (impossible) to quantify**
  - Subjective, non-linear development time efforts
  - Predicting sustained performance, reliability on radically new architectures and software systems
    - Simulators don't scale well
    - Analytical models don't capture full complexity of features

We need science-based metrics!!

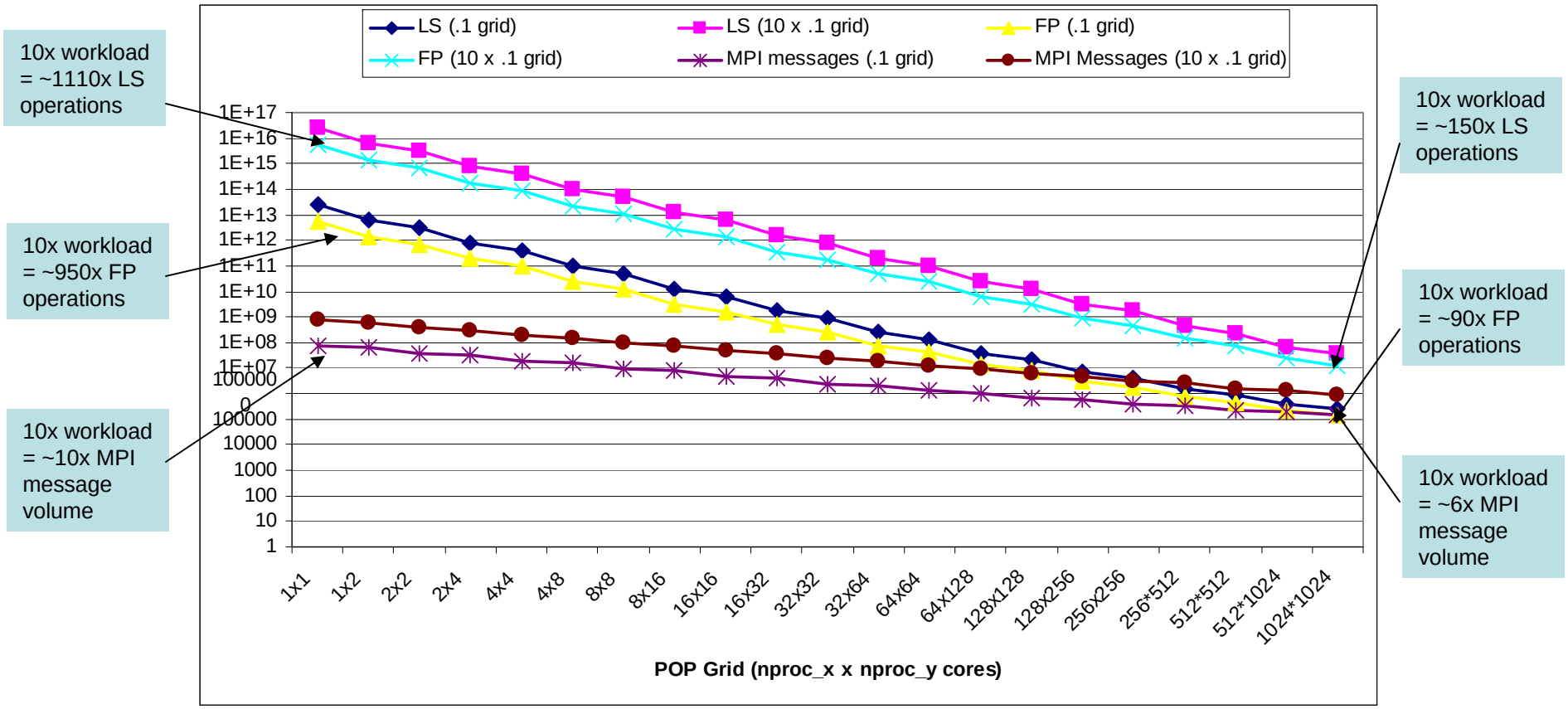
- **Application-specific measures that relate to better/more science**
  - POP uses simulation years per day
- **Methods to predict their performance accurately**
  - Using scalable simulators, symbolic modeling tools, code analysis tools
  - Targeting complex architectures like multicore, accelerators (e.g., CELL, GPUs)
- **Solve the inverse problem for 'perfect' HPC architecture**
- **Joule metrics in DOE have characterized science-based metrics for ~12 applications**



Modeling Assertions Provide Scalable Prediction Methods

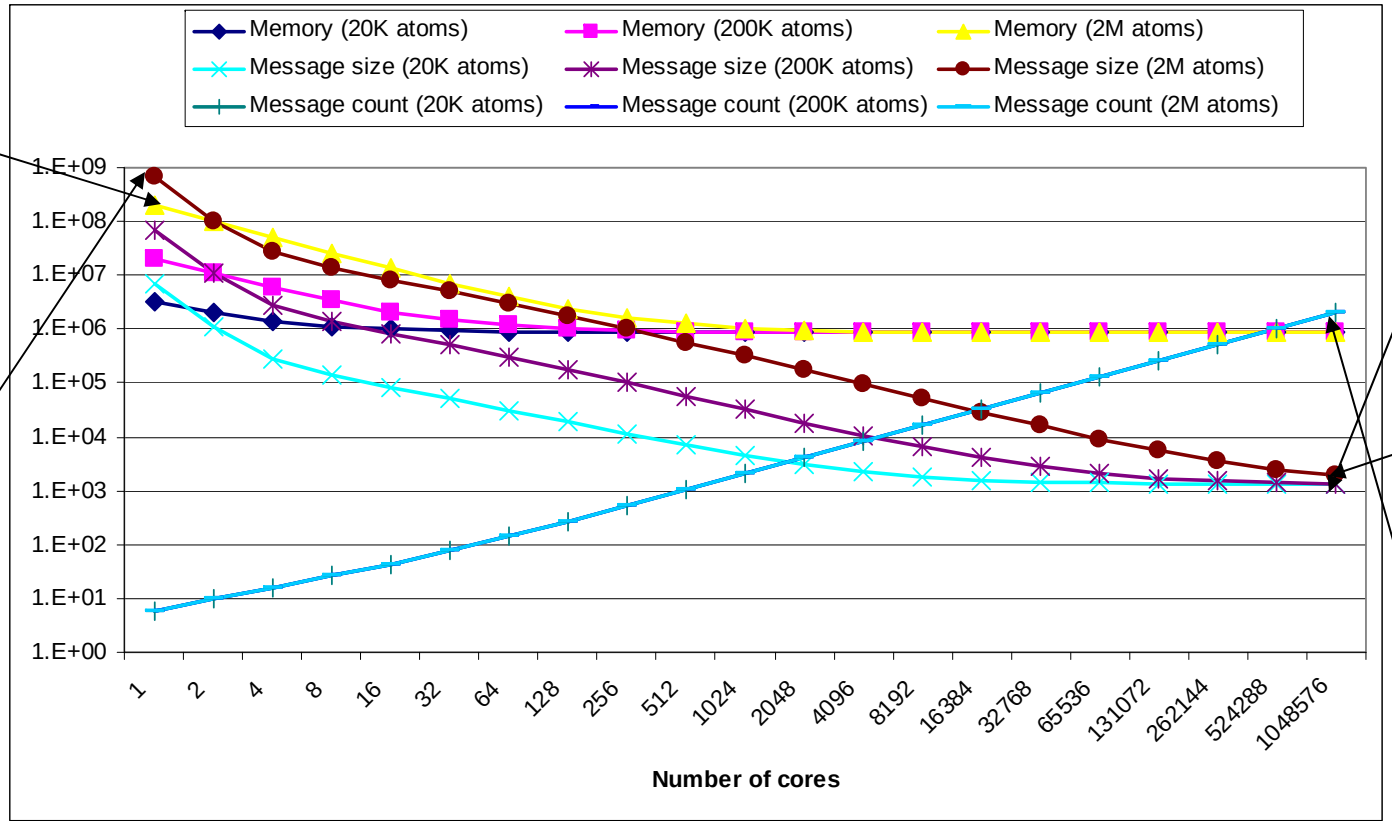


POP Scaling



- **Symbolic performance models for limited scalability Barotropic phase:**
  - Number of floating-point operations
  - Number of load-store operations
  - MPI message sizes and patterns
- **Error bounds:**
  - Convergence rates of Conjugate Gradient iterations

AMBER Scaling



100x workload = ~65x physical memory requirements

100x workload = ~97x MPI message volume (bytes)

100x workload = ~1x physical memory requirement

100x workload = ~2x MPI message volume

Message count on 1M cores = 400,000x on single core

- **Sander program in AMBER framework:**
  - Biomolecular MD simulations
  - Explicit solvant method
  - Particle Mesh Ewald (PME) approximation
- **Assumptions about input configurations:**
  - Cubic box with periodic boundaries
  - Constant Residual value

Science Based Metrics are the Ultimate Goal

- **Capture the full application and architecture interactions to advance science**
- **Difficult to predict accurately**
  - Use proxies instead
    - FLOPS
    - Memory, I/O operations
- **Build performance prediction environments that**
  - are accurate, efficient, scalable
  - Use combination of techniques: simulators, symbolic modeling

"HPC Performance Metrics: Should We Drop FLOPS?"

> As computer architectures have advanced, making
> floating-point operations relatively inexpensive,
> it is still very ingrained in HPC to compare systems
> by their FLOPS rates, and to optimize algorithms to
> minimize their floating-point operation count even
> at the cost of increased memory bandwidth demands
> or programming effort.
>
> Should we drop FLOPS as a metric? If the answer is yes,
> then is there a way to gracefully (or perhaps suddenly)
> move to a metric other than FLOPS that has the predictive
> value and ease-of-analysis that FLOPS had in the 1960s
> and 1970s?