Adaptive, Large-Scale Computing Systems: Need vs. Want

Daniel A. Reed
Vice President for Research and Economic Development
University Computational Science and Bioinformatics Chair
Computer Science, Electrical Engineering & Computer Engineering, and Medicine

www.hpcdan.org

dan-reed@uiowa.edu

The University of Iowa

DOE SEAB HPC committee charge

The justification for an exascale computing capability initiative.

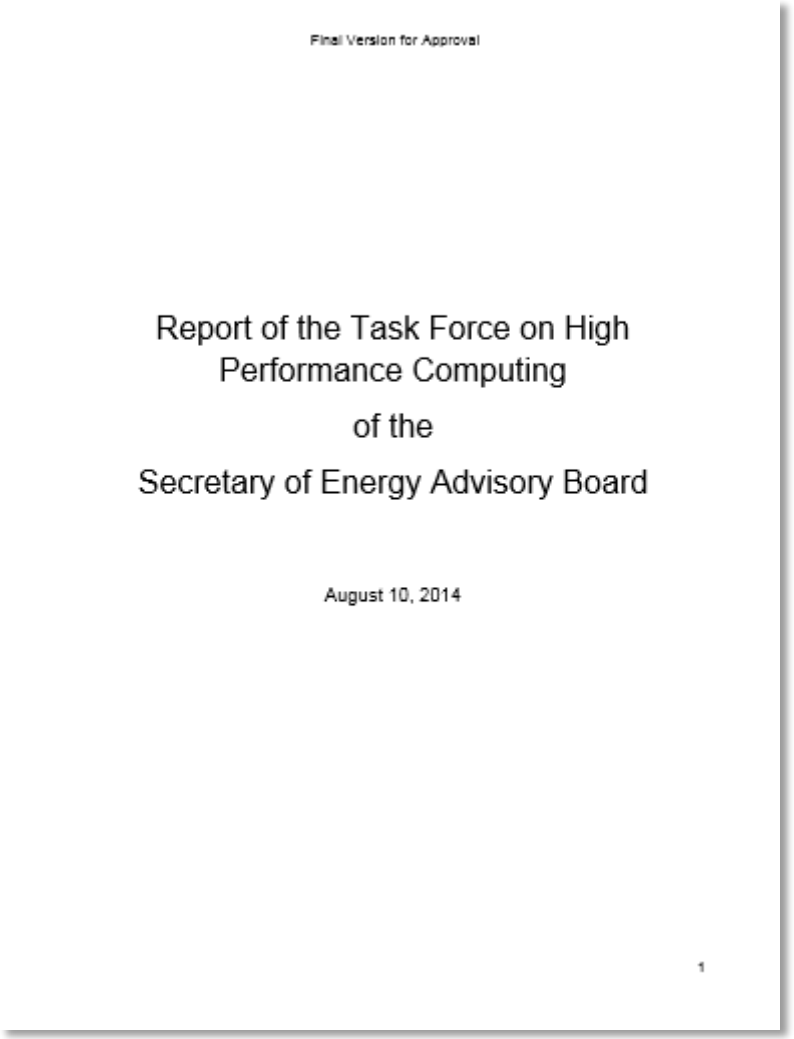
- DOE missions
- Fundamental research opportunities
- Broader societal benefits from an open, non-classified exascale program and potential market barriers inhibiting private development of exascale computing

Related basic research necessary to enable next generation high performance computing (e.g. mathematics, computer science, etc., including quantum and superconducting computing)

The current state of technology and plans for an exascale program in the Department of Energy and other federal agencies

Role of the Department of Energy in leading the development of exascale computing - including its involvement and collaboration with industry, universities and other government agencies on high performance computing

Implications of data centric computing for exascale computing


Final Version for Approval

Report of the Task Force on High Performance Computing

of the

Secretary of Energy Advisory Board

August 10, 2014

1

energy.gov/seab/secretary-energy-advisory-board-seab-task-force-next-generation-high-performance-computing

U.S. DEPARTMENT OF ENERGY

THE UNIVERSITY OF IOWA

U.S. DOE SEAB HPC committee

SEAB Members

- Shirley Ann Jackson, Co-Chair, Rensselaer Polytechnic Institute
- Michael McQuade, Co-Chair, United Technologies Corporation
- Ram Shenoy, ConocoPhillips
- Steve Koonin, NYU Center for Urban Science and Progress

External Members

- Roscoe Giles, Boston University
- Jim Hendler, Rensselaer Polytechnic Institute
- Peter Highnam, IARPA
- Anita Jones, University of Virginia
- John Kelly, IBM
- Craig Mundie, Microsoft
- Thomas Ohki, Raytheon BBN Technologies
- Dan Reed, University of Iowa
- Kord Smith, Massachusetts Institute of Technology
- John Tracy, Boeing (Ted Colbert)

DOE SEAB committee draft recommendations

1. Investable needs exist for an exaX class machine
2. *Significant, but projectable technology development can enable one last "current" generation machine (1-10 exascale level)*
3. *Classical high end simulation machines are already significantly impacted by many of the data volume and architecture issue*
4. Data centric at the exascale is already important for DOE missions
   - *Rapidly scaling to and beyond levels of performance that are comparable to hose needed for classic high performance floating point computation*
5. Common challenges and under-girding technologies span compute needs
6. Factors driving DOE's historical role in leadership computing still exist and will continue to do so
7. A broad and healthy ecosystem is critical to the development of exascale and beyond systems
8. *It is timely to invest in science, technology and human investments for "Beyond Next"*
   - *Superconducting, quantum, biological/cognitive (neurosynaptic)*

Note: emphasis mine

Culture and economics: getting what you need

Industry

- Capital is really, really cheap (look at interest rates)
- Labor is increasingly expensive
- ROI drives behavior

Academia and government

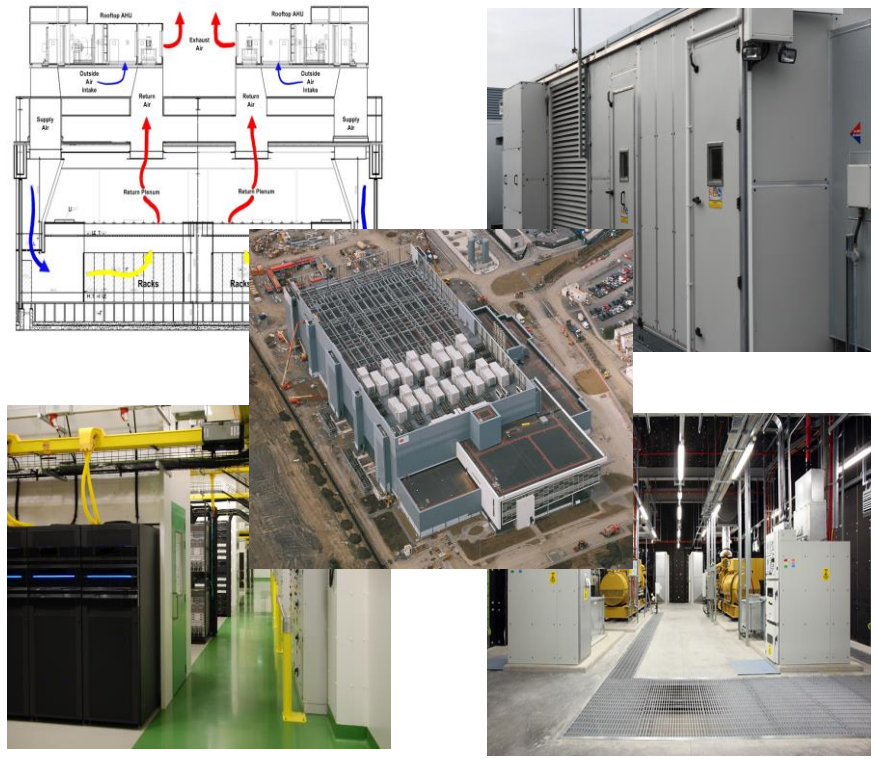
- Capital is very expensive
- Labor is still cheap
- Other metrics drive success

To change the game, change the metrics …

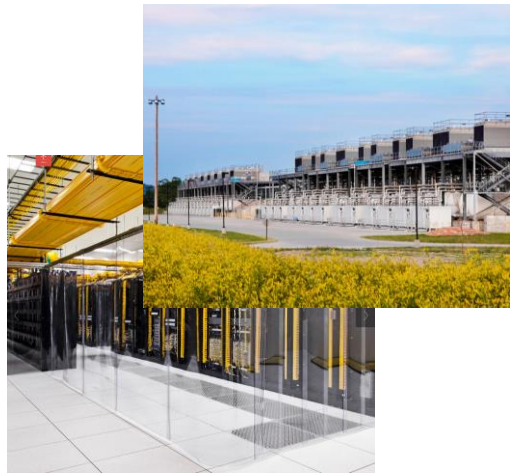
- Infrastructure, personnel, social and political

Put another way, match what you want with what you need

Cloud computing lessons



Microsoft Dublin



Google, Council Bluffs

Generic server design

- *OpenCompute*
- *Workload-specific optimization*
  - *Functional accelerators*
- *ODM, not OEM partnerships*

Energy optimization

- *Substations and generation*
- *Switchgear control*

Programming efficiency

- *Rich toolkits and expression*

Systemic resilience

- *Failure management, not avoidance*

Network optimization

- Flatter networks
- Software virtualization and flow

Supply chain optimization

- The advantage of scale

Netflix Simian Army

Chaos Monkey

- *Random service termination to ensure other services continue operation*

Latency Monkey

- *Simulates service degradation and ensures services react*

Janitor Monkey

- Searches for and turns off unused resources

Conformity Monkey

- Ensures virtual machines meet specified standards

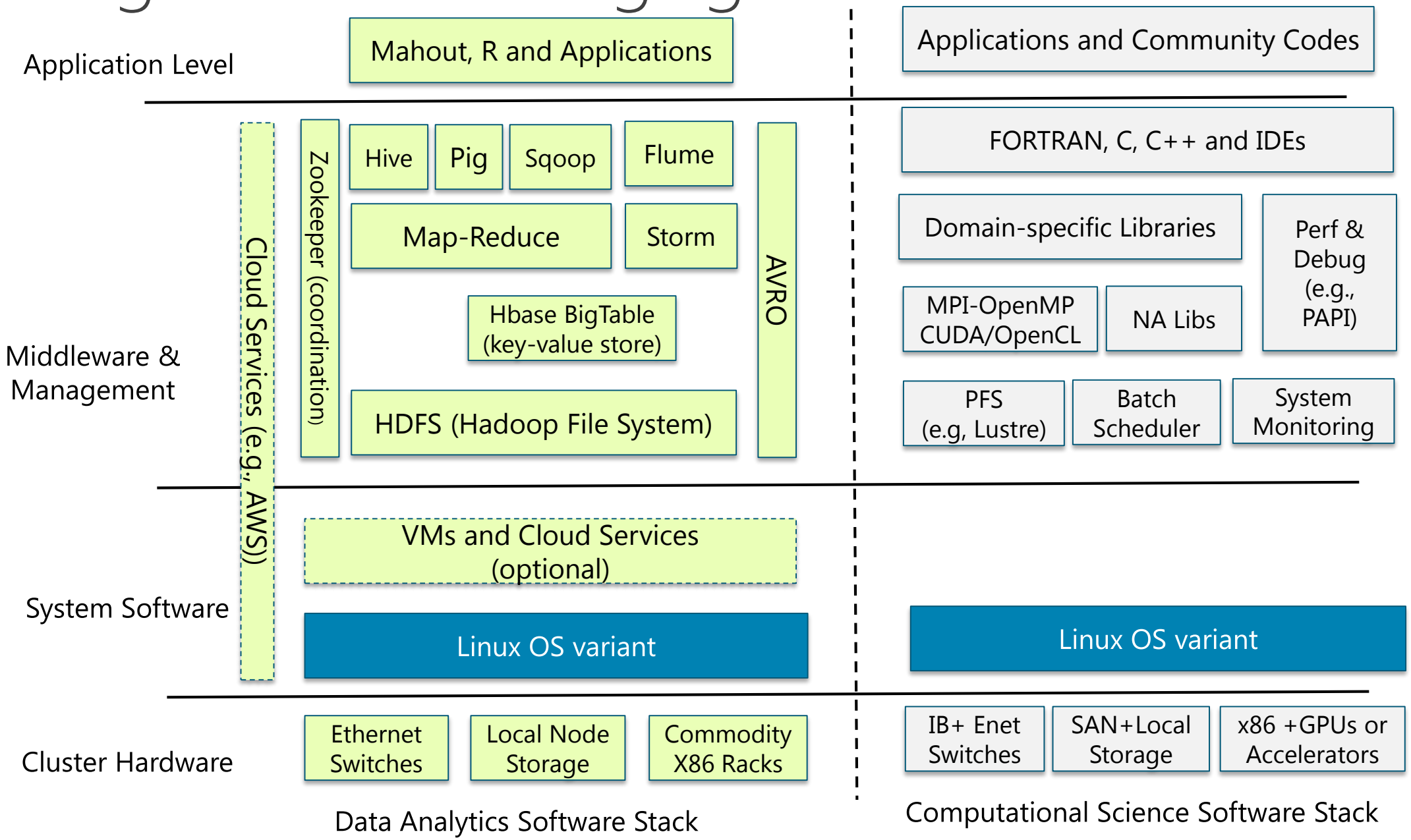
Doctor Monkey

- monitors the “health” of various virtual machines

Security Monkey

- Monitors and analyzes system security

Integration is challenging in both directions



Diverging cultures and loss

Scientific application complexity is rising

- Multidisciplinary fusion
- Temporal and spatial adaptation
- Data assimilation and processing

… along with multiple optimization axes

- Massive parallelism with heterogeneous cores
- Resilience/reliability at large scale
- Energy optimization for utility

C, Fortran, C++, MPI, OpenMP

CUDA/OpenCL

Python, Ruby, PIG, FLUME, R

Cloud/Web Services

Cognitive complexity must decline …

- *… else the number of parallel software developers will asymptotically approach zero*

Technical and mainstream software development have diverged

Shifting ratios: shifting optimizations

Rising costs

- Personnel (no Moore's law here)
- Energy costs (capital and operating)

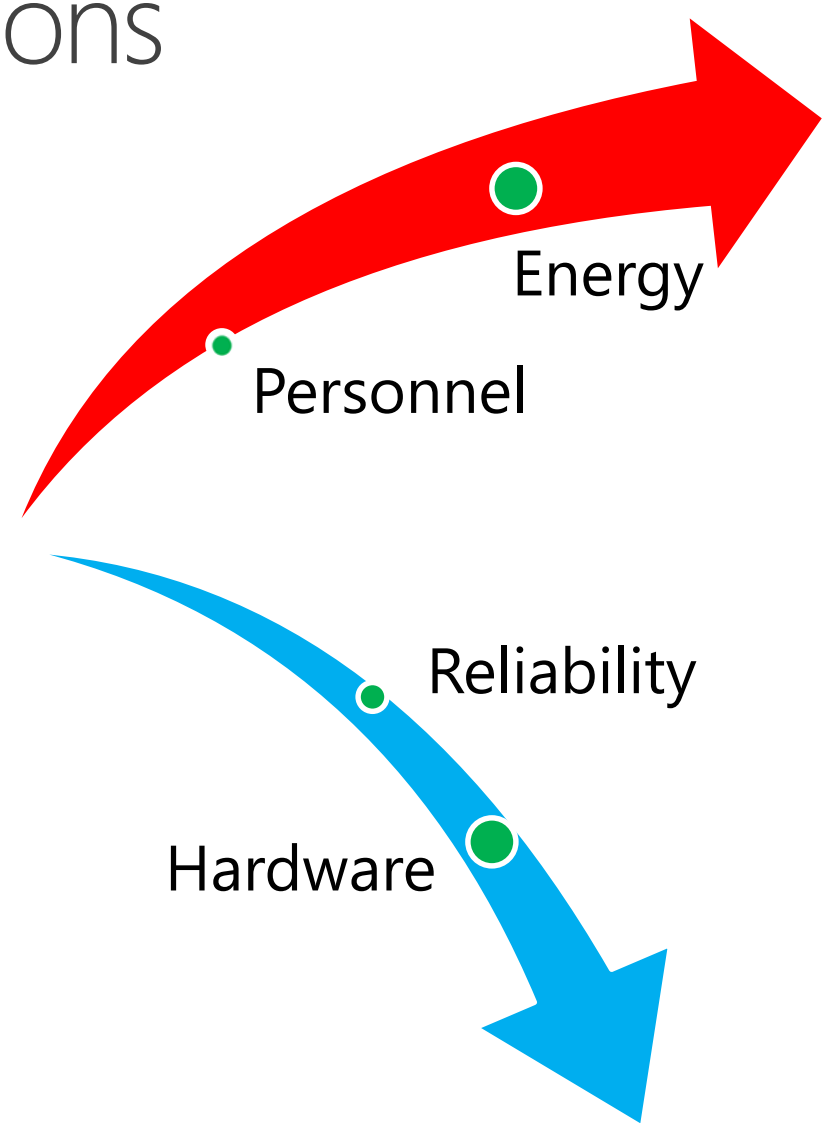
Declining costs

- Hardware for given performance level
- $/FLOP continues to decline rapidly

Hardware reliability

- Partially determined by market
- Mass market rewards low cost, not high reliability
  - Think mobile devices

Follow the money …



Exascale resilience, provenance and scale

System sizes are rising rapidly
- The law of large numbers applies

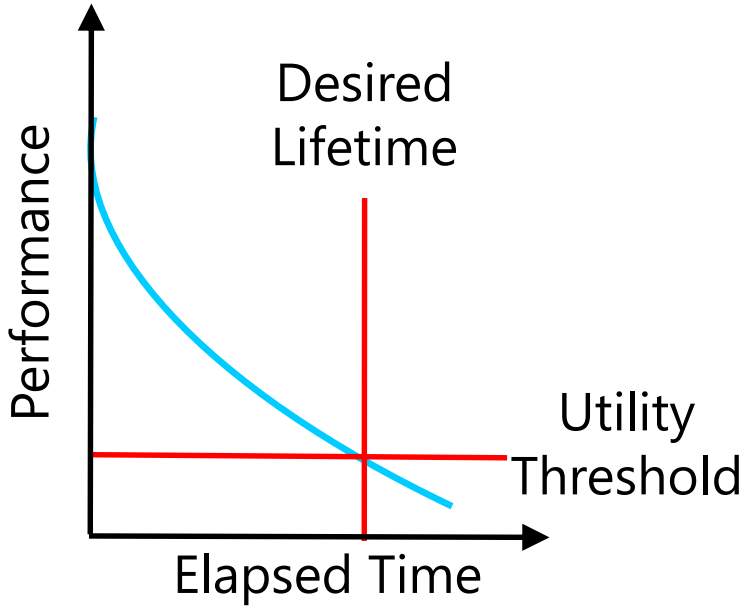
Failures are frequent
- Component MTBF is not that high
  - Disks, power supplies, fans, DRAM

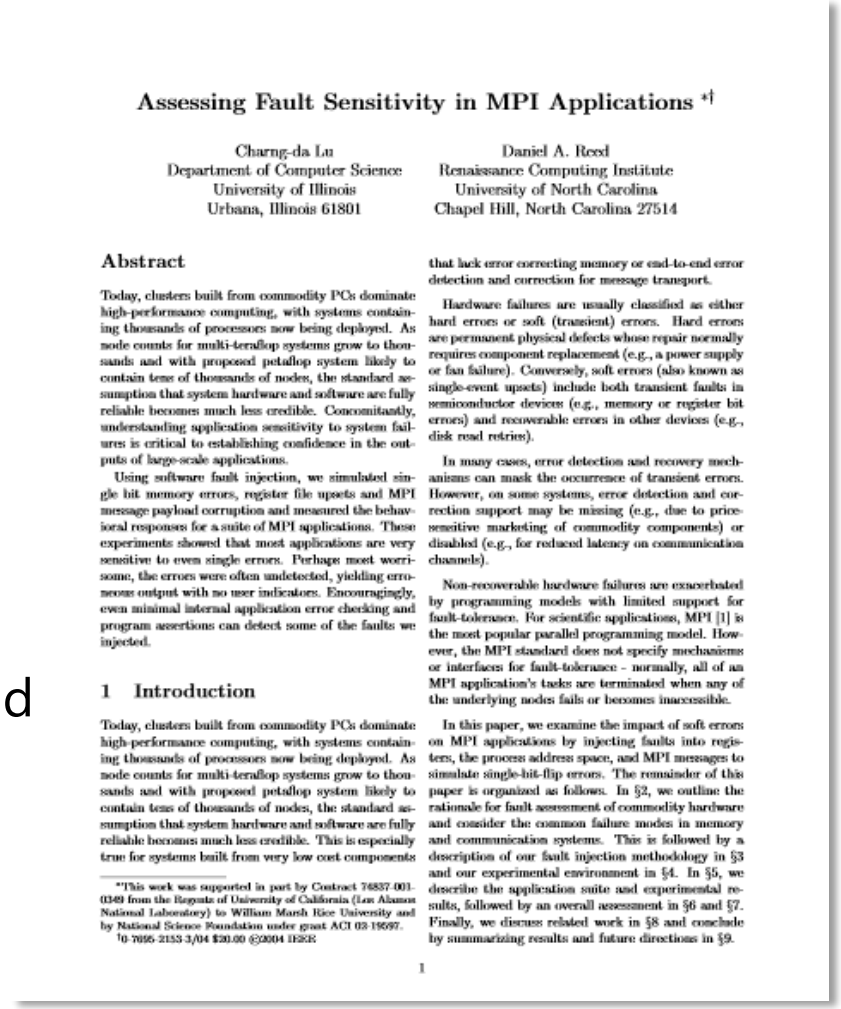
System resilience dominates
- Components are less important

Fail in place performability
- Over-provision and accept failure

Weak consistency
- Eventual reconciliation

Biologically inspired resilience models
- Systemic resilience with unreliable components





Assessing Fault Sensitivity in MPI Applications *†

Charng-da Lu
Department of Computer Science
University of Illinois
Urbana, Illinois 61801

Daniel A. Reed
Renaissance Computing Institute
University of North Carolina
Chapel Hill, North Carolina 27514

Abstract

Today, clusters built from commodity PCs dominate high-performance computing, with systems containing thousands of processors now being deployed. As node counts for multi-teraflop systems grow to thousands and with proposed petaflop system likely to contain tens of thousands of nodes, the standard assumption that system hardware and software are fully reliable becomes much less credible. Concomitantly, understanding application sensitivity to system failures is critical to establishing confidence in the outputs of large-scale applications.

Using software fault injection, we simulated single bit memory errors, register file upsets and MPI message payload corruption and measured the behavioral responses for a suite of MPI applications. These experiments showed that most applications are very sensitive to even single errors. Perhaps most worrisome, the errors were often undetected, yielding erroneous output with no user indicators. Encouragingly, even minimal internal application error checking and program assertions can detect some of the faults we injected.

1 Introduction

Today, clusters built from commodity PCs dominate high-performance computing, with systems containing thousands of processors now being deployed. As node counts for multi-teraflop systems grow to thousands and with proposed petaflop system likely to contain tens of thousands of nodes, the standard assumption that system hardware and software are fully reliable becomes much less credible. This is especially true for systems built from very low cost components that lack error correcting memory or end-to-end error detection and correction for message transport.

Hardware failures are usually classified as either hard errors or soft (transient) errors. Hard errors are permanent physical defects whose repair normally requires component replacement (e.g., a power supply or fan failure). Conversely, soft errors (also known as single-event upsets) include both transient faults in semiconductor devices (e.g., memory or register bit errors) and recoverable errors in other devices (e.g., disk read retries).

In many cases, error detection and recovery mechanisms can mask the occurrence of transient errors. However, on some systems, error detection and correction support may be missing (e.g., due to price-sensitive marketing of commodity components) or disabled (e.g., for reduced latency on communication channels).

Non-recoverable hardware failures are exacerbated by programming models with limited support for fault-tolerance. For scientific applications, MPI [1] is the most popular parallel programming model. However, the MPI standard does not specify mechanisms or interfaces for fault-tolerance – normally, all of an MPI application's tasks are terminated when any of the underlying nodes fails or becomes inaccessible.

In this paper, we examine the impact of soft errors on MPI applications by injecting faults into registers, the process address space, and MPI messages to simulate single-bit-flip errors. The remainder of this paper is organized as follows. In §2, we outline the rationale for fault assessment of commodity hardware and consider the common failure modes in memory and communication systems. This is followed by a description of our fault injection methodology in §3 and our experimental environment in §4. In §5, we describe the application suite and experimental results, followed by an overall assessment in §6 and §7. Finally, we discuss related work in §8 and conclude by summarizing results and future directions in §9.

*This work was supported in part by Contract 74837-001-0349 from the Regents of University of California (Los Alamos National Laboratory) to William Marsh Rice University and by National Science Foundation under grant ACI 03-19697.

†0-7695-2153-3/04 $20.00 ©2004 IEEE

SC'04 Paper

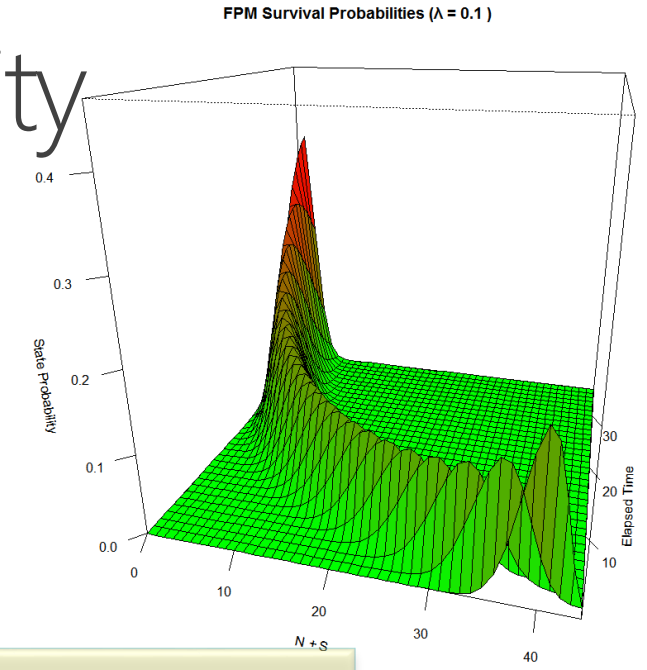
Field replaceable module (FPM) optimization

IBM

Microsoft

Google

Over-provisioned hardware performability

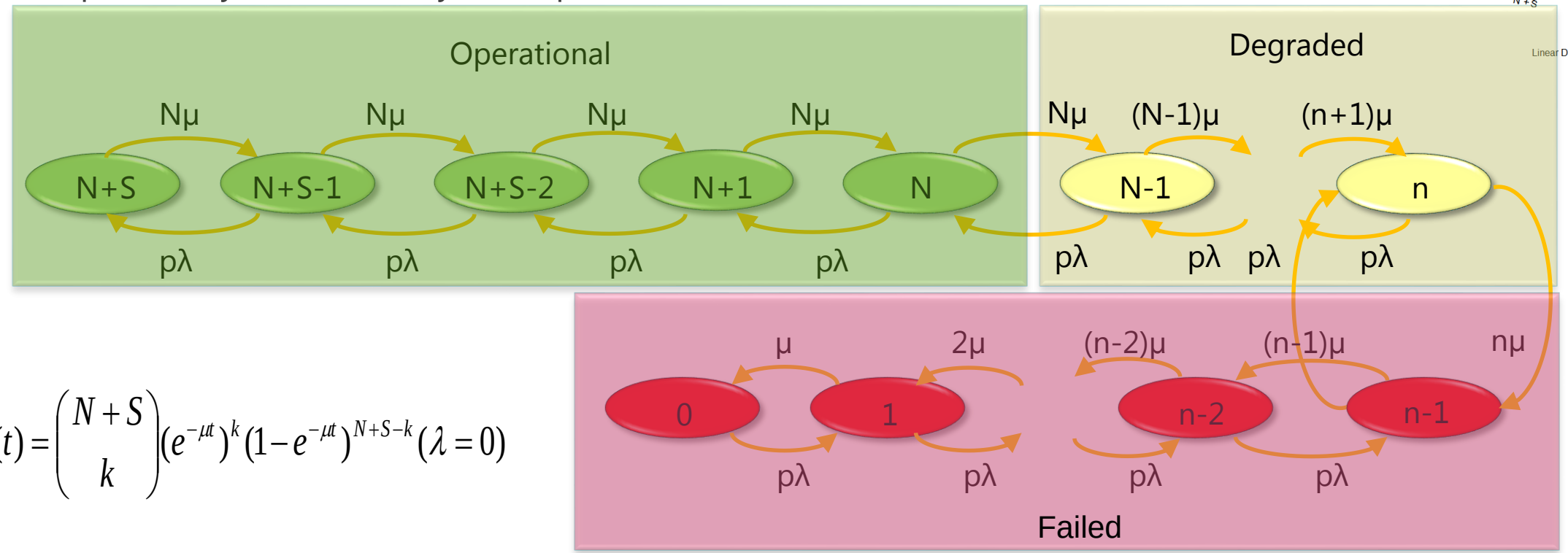


Balance node RAS against system RAS

- *NRE versus acquisition/repair*

A simple, optimistic provisioning model

- N servers and S spares
- Servers fail independently at rate μ
  - Spares may (λ >0) or may not repaired (λ = 0)



$$p_k(t) = \binom{N+S}{k} (e^{-\mu t})^k (1-e^{-\mu t})^{N+S-k} (\lambda = 0)$$

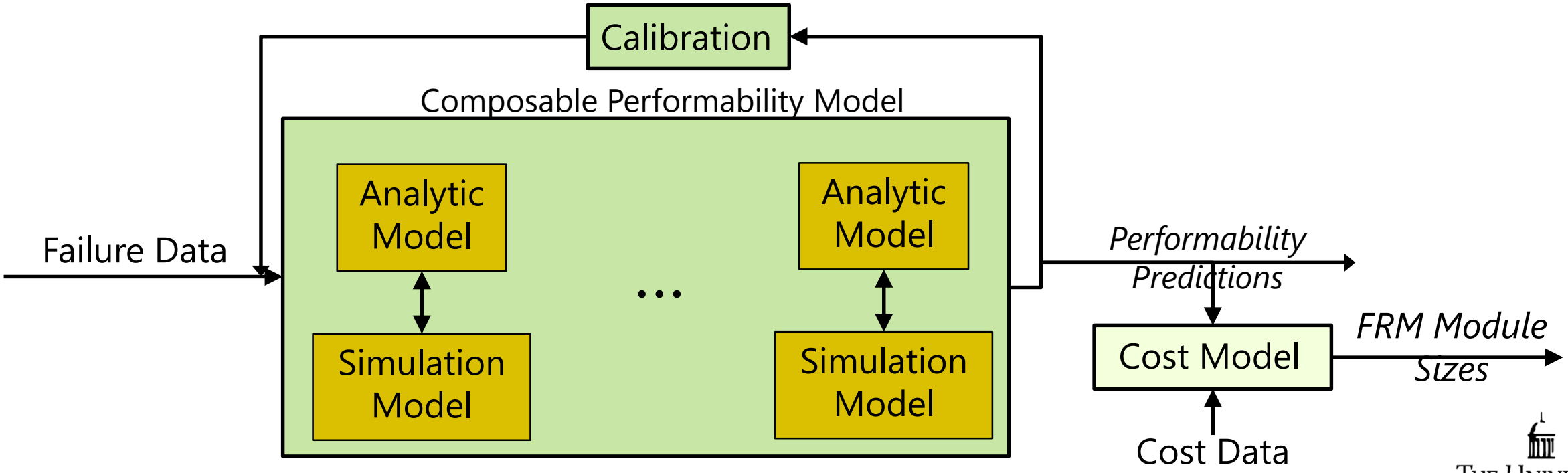
Modeling approaches

Analytic models

- Simple, aggregate node models
- Only analytically tractable distributions
- State space explosion challenges
- Easy to explore large parameter spaces

Discrete event simulation

- Detailed, hierarchical models
- Generalized failure distributions
- More difficult to explore large parameter spaces



Scaling and near complete decomposability

Weakly interacting partitions (racks or FPMs)

- Failures (usually) have local hardware impact
- Rack failures rarely affect other racks

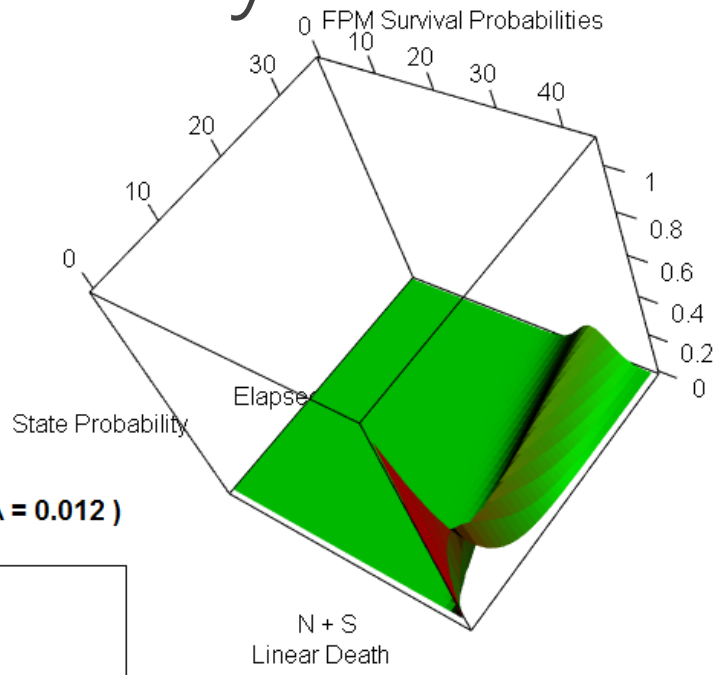
Implications

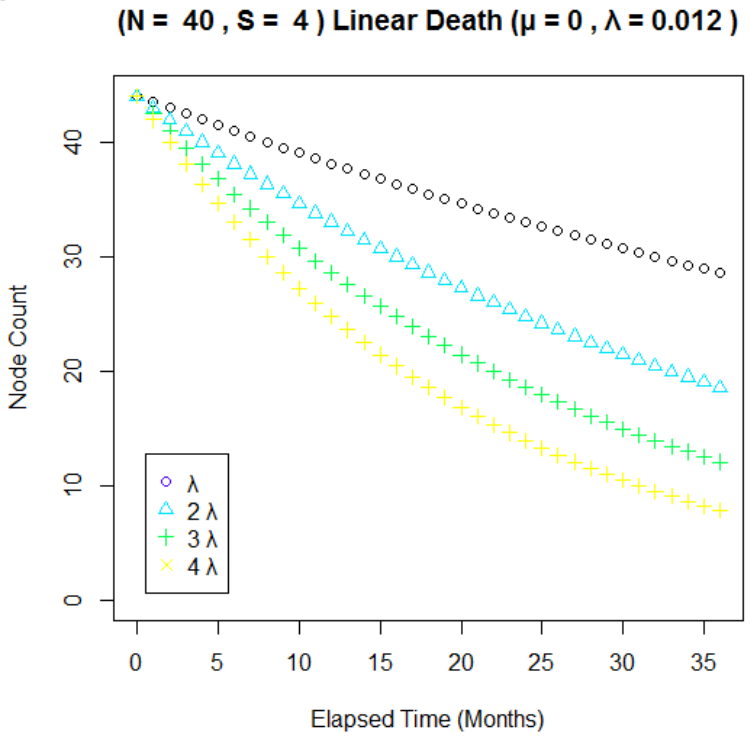
- State transition matrix P is strongly diagonal
- Partition models can be solved separately

Solution to state space explosion

- Numerically tractable even at large scale

$$P = \begin{bmatrix} q_{11} & & & & \\ & q_{22} & & & \\ & & \cdots & & \\ & & & q_{kk} & \\ & & & & \cdots \\ & & & & & q_{MM} \end{bmatrix} + \varepsilon S$$





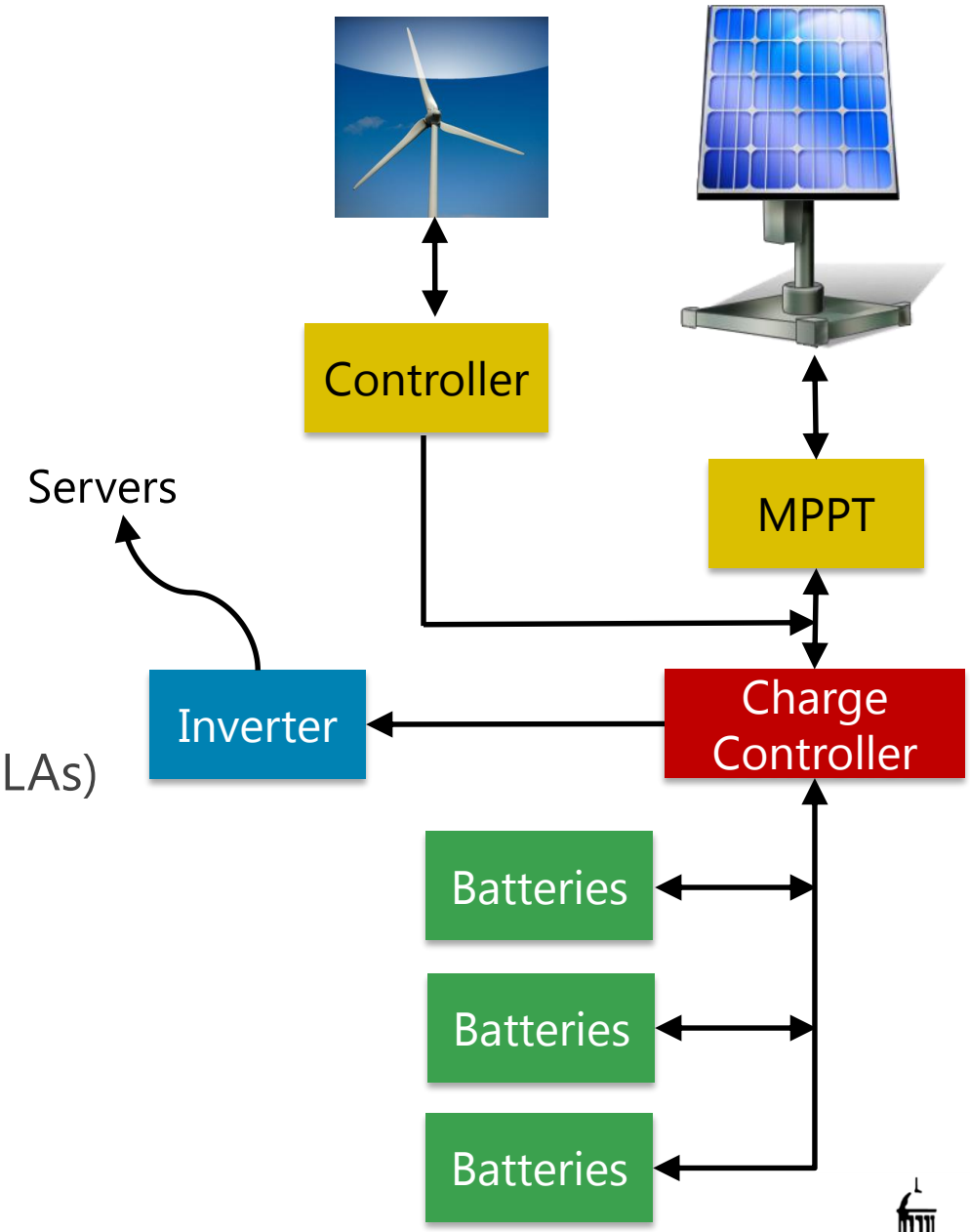
Rethinking computing energy

Multiple energy sources
- Electrical grid, solar, wind, fuel cell, …

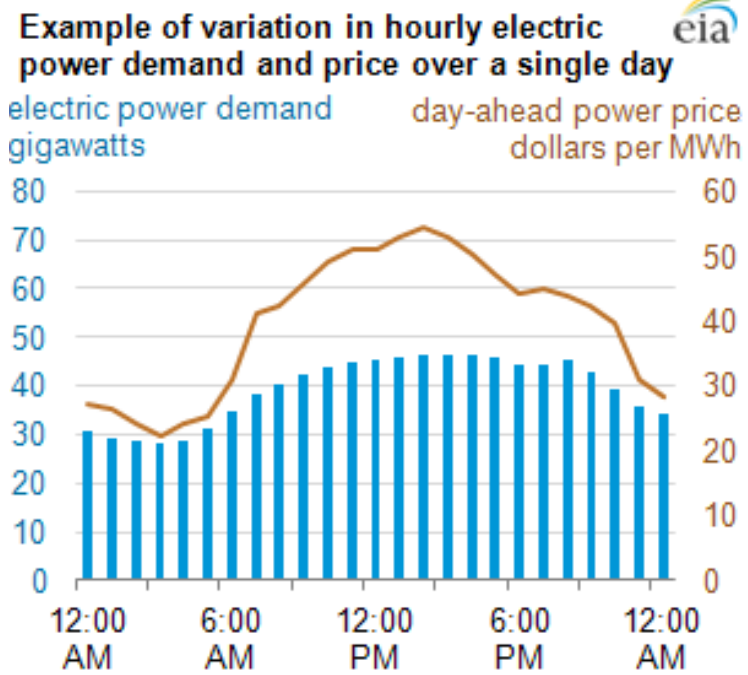
Multiple cost functions
- Energy pricing, carbon taxes, varying availability
- Hardware, data transfer bandwidth/latency …

Multivariate optimization and prediction
- Workload demand (diurnal and seasonal)
- Workload location subject to service level agreements (SLAs)
- Weather and seasonal models
- Auction-based energy pricing
- Infrastructure (UPS, optical fiber and computing)

Scheduling subject to energy and reliability
- Cost, availability, resilience …



Energy constrained scheduling



www.eia.gov

Two options, generalizable to auction-based pricing

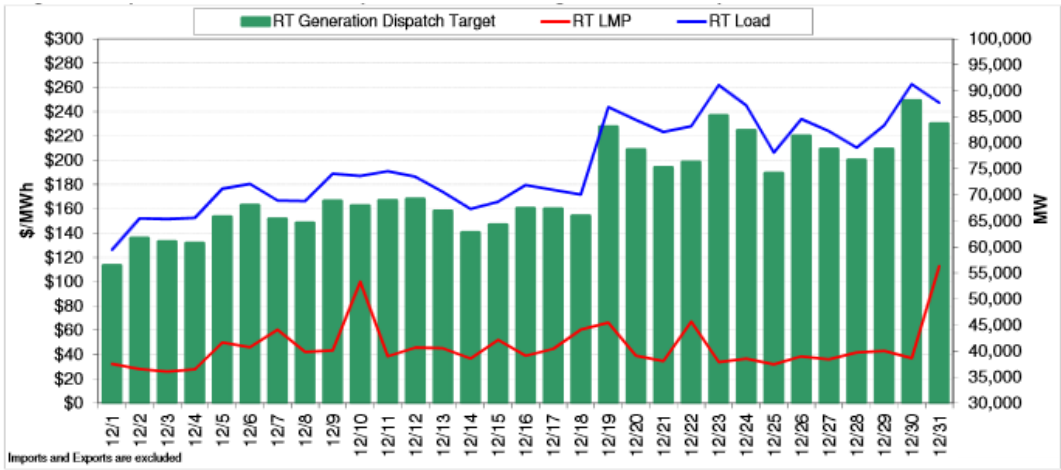
- Peak/off-peak pricing but no limit on energy availability
- Fixed pricing but peak/off-peak energy availability

Given

- System size *N* nodes
- Total system energy budget *B*
- Sequence of jobs $J = j_1, j_2, \ldots, j_M$

Choose an optimal subset *S* of *J* such that

$$\sum_{j_i \in S} Size(j_i) \leq N \qquad \sum_{j_i \in S} Energy(j_i) \leq B$$



December 2013 MISO

HPC resource allocation and policies

User allocations are now of substantial value

- Depreciated capital cost over system lifetime
- Energy consumption for parallel job

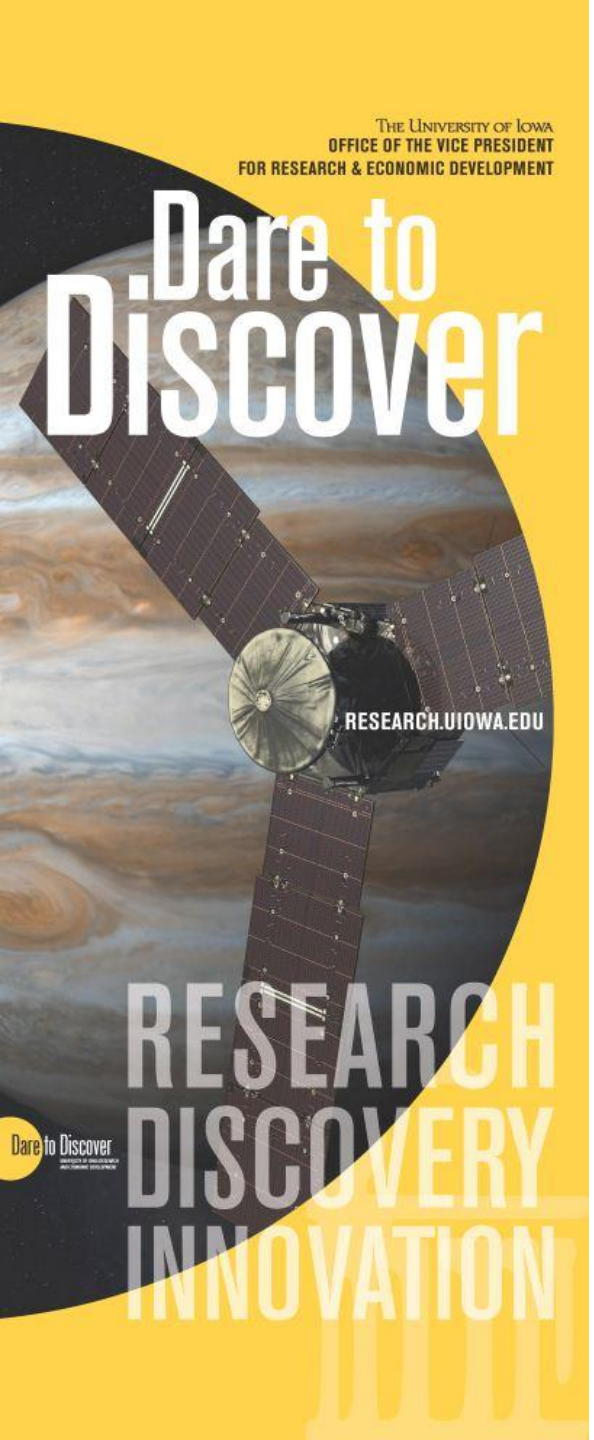
Limited incentives for user responsibility

- Capital or operating cost efficiency

Exacerbated by

- Sub-linear parallel speedups
- Accelerators and node heterogeneity
- Multiple energy envelopes

New reward models must be explored …



Discussion

The University of Iowa