A Multi-resolution Emulation + Simulation Methodology for Exascale

Laxmikant V. Kalé, Nikhil Jain, Akhil Langer, Esteban Meneses, Phil Miller, Osman Sarood, Ehsan Totoni and Eric Bohm

Parallel Programming Laboratory
University of Illinois Urbana-Champaign

November 27, 2013

Motivation (1/2)

Challenge: Achieving simulation fidelity for dynamic applications on next generation hardware.

Example software sources of dynamic behavior

- Adaptive algorithms:
  - Dynamic adaptive mesh refinement
  - Multiple time stepping
  - multiscale & multiphyics
  - Dynamic Load Balancing
- Adaptive runtime systems:
  - Work stealing
  - Prioritized adaptive overlap of computation and communication
  - Work migration triggered by heterogenous DVFS
  - Message aggregation and adaptive collective layout
  - Detection of optimal load balancing frequency
  - Workload balance between host and accelerator
  - In situ performance analysis
  - IO management

Motivation (2/2)

Challenge: Achieving simulation fidelity for dynamic applications on next generation hardware.

- Example operational sources of dynamic behavior (interference)
  - Contention for network resources from other jobs (network noise)
  - Contention for I/O resources from other jobs (I/O noise)
  - Monitoring and management overhead (daemons)
  - Virtualization management
- Example hardware sources of dynamic behavior
  - Component failures/errors
  - Dynamic CPU frequency changes
  - Thermal restrictions
  - Power constraints
  - Performance heterogeniety
  - Adaptive routing
  - Hardware threads

Solution: Emulation followed by Simulation

- Full-scale emulation of applications.
- Virtualization to emulate full platform on smaller platform.
- Capture traces and event dependency for region of interest.
- Input traces to platform simulator to observe simulated effect of platform design choices.

A Motivating Example

Can your software predict the performance of an adaptive, dynamically load balanced, multi-stepping PME, MD code, on 25k heterogeneous nodes?

How about when we add fault tolerance protocols, replica exchange, core level DVFS, and run on a million nodes?

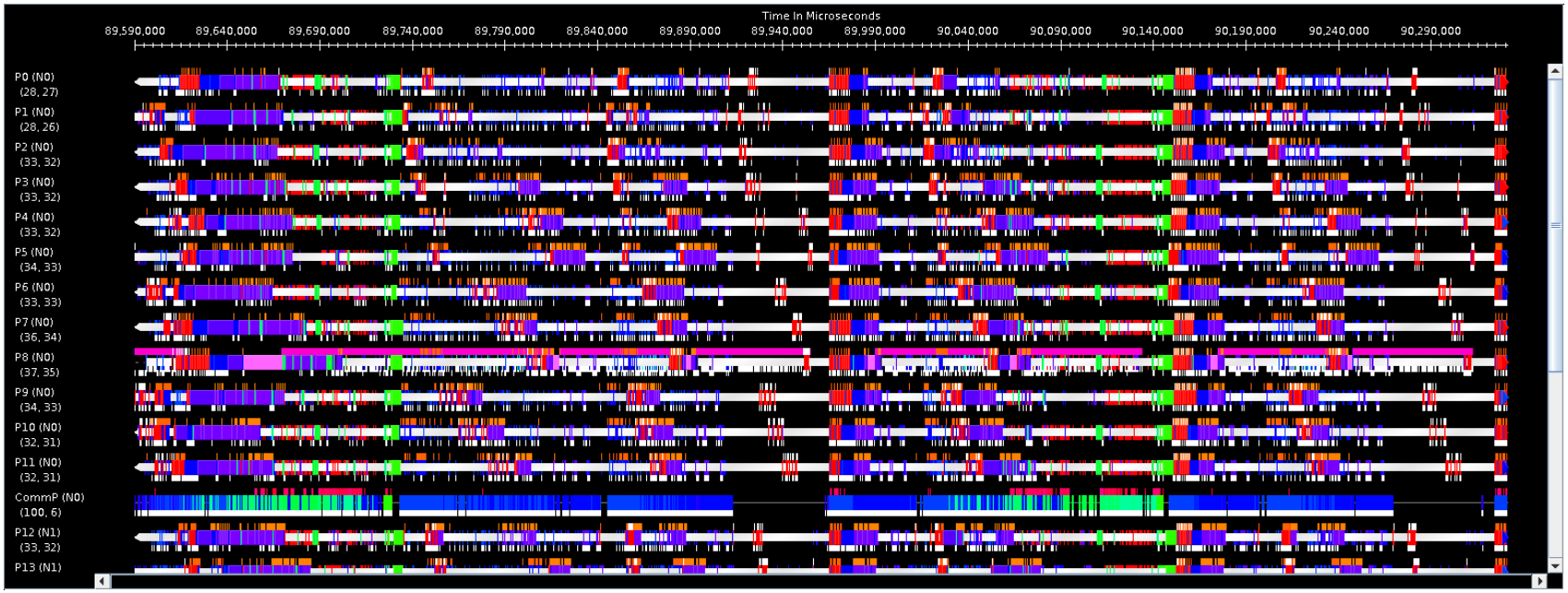


Figure : NAMD 100 Million atoms (STMV) multistepping PME with non-bonded offloaded to GPGPU on XE7

My simulator can't handle that yet either.

Emulator + Simulator Framework

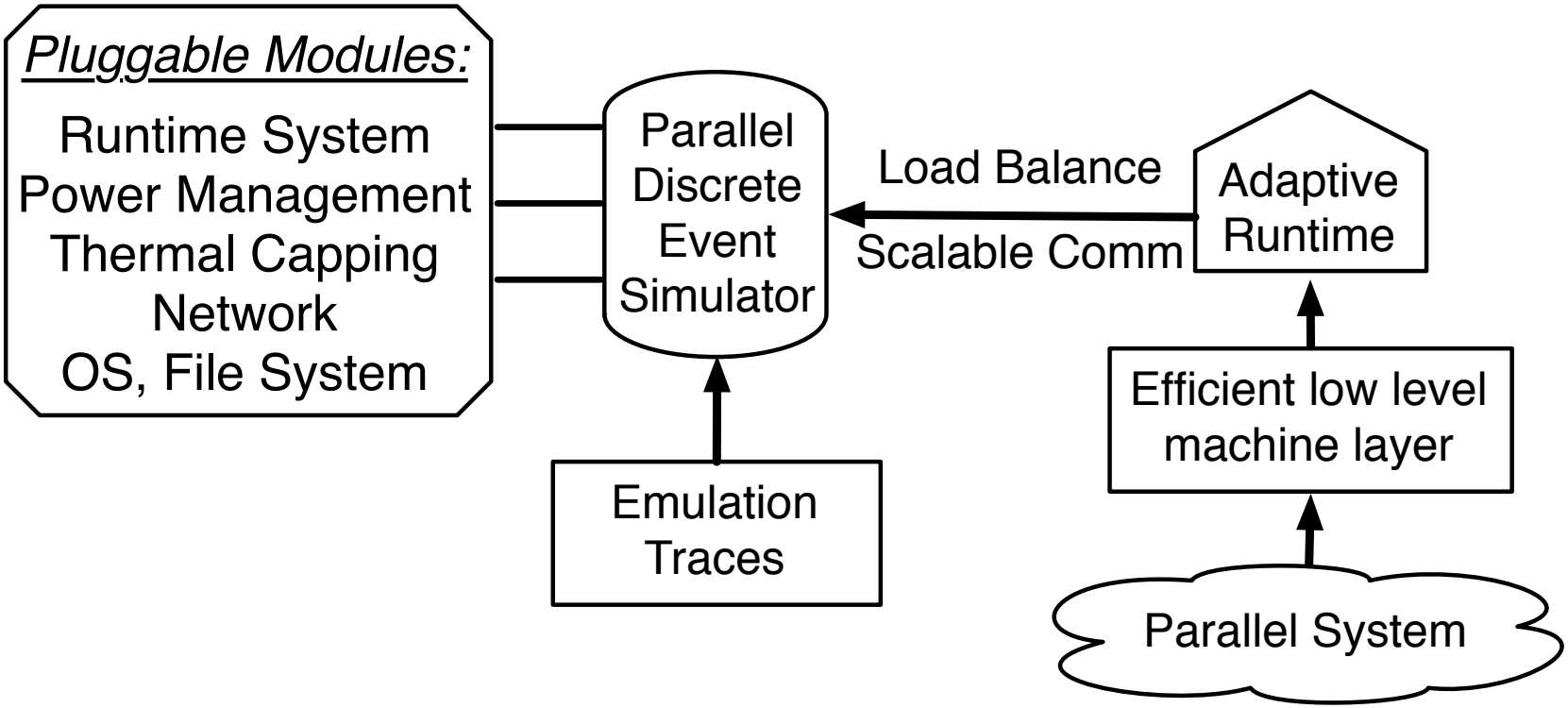


Figure : Software stack for scalable simulation of performance and non-performance application characteristics.

Scalable Discrete Event Simulation

- Co-design of simulator with Runtime System
- Efficient use of low level communication infrastructure
- Introspection feedback cycle
  - Using relative event importance to guide detail of simulation level

Network Simulation with Adjustable Granularity

Challenges:

- Scale of network
- Dynamic and complex application communication patterns
- Greater impact of faults

Solutions:

- Multi-granularity with acceptable accuracy loss - select granularity based on the component
- Increase abstraction level for network components
- Accelerated simulation of phases found to be contention free

Runtime

Challenges: Simulator must account for many possibilities

- Message Passing SPMD
- Heterogeneous platforms
- Work Stealing
- Message driven execution with migratable tasks
- Exammple: Dynamic load balancing reassigns work
  - Simulators must adjust subsequent events for units impacted by remapping
  - Modifying a trace post mortem for this purpose is highly limiting

Simulation of Dynamic Runtime Systems

- A simulator which models the runtime system can model migration use cases at varying degrees of resolution
  - Oracular message redirection
  - Explicit communication
  - Experimental load balancing policies
- Event Queue Management
  - FIFO Queuing
  - Prioritization
  - Critical path detection

Resilience

MTBF at 10s of minutes

- Interaction with power management
- Interaction with thermal and frequency performance trade offs

MTBF too simple a model for many considerations.
Resilience modeling will require:

- application communication patterns and dependencies
- characteristics of platform
- realistic distributions of fault patterns
- multiplicity of root causes (node, network, software, other).
- hard failures and soft failures

- *Studying the steady state of a fault tolerance protocol at exascale will be highly resource intensive.*
- multiple levels of resolution will be necessary to effectively scale from seconds, to hours, to days of simulation

Power

Maximizing data center performance under strict power budget
Dynamic power management is assumed, therefore it must be simulated

- Accurate models of power draw by subsystems (chip, memory, etc.)
- impact of power capping the subsystems
- component-wise analysis of applications
  - breaking into Sequential Execution Blocks (SEB)
  - characterize power profile of SEB
  - simulate mutually recursive impact of power on SEBs
  - global power management system specifies power allocation to components/subsystems
  - provide power consumed metrics as output of simulation
- network and I/O subsystems require analogous attention
  - e.g. simulate effect of turning network links on/off based on usage

Thermal

Core temperature may impact performance, reliability, and power consumption

- Dynamic thermal management will impact performance
- Similarly it will impact measured load
- Applications will be impacted in idiosyncratic ways
- Simulation of those interactions requires the development of models to simulate the interaction of dynamic management with dynamic applications and runtimes.
- Simulating these interactions will require the development of more efficient and mature models, especially in an exascale context

Introduction

- Challenges: Coping with the complexities generated by sophisticated applications and complex machines.
  - Adaptive numerical algorithms:
    - ★ Dynamic adaptive mesh refinement
    - ★ multi-time-stepping
    - ★ multiscale & multiphyics
    - ★ Dynamic Load Balancing
  - Static and dynamic variability in hardware:
    - ★ Component failures/errors
    - ★ Dynamic CPU frequency changes
    - ★ Thermal restrictions
    - ★ Power constraints
- Solution: Introspective and Adaptive Runtime Systems
- Need: Modeling Infrastructure for IARTS Development

The Big Picture

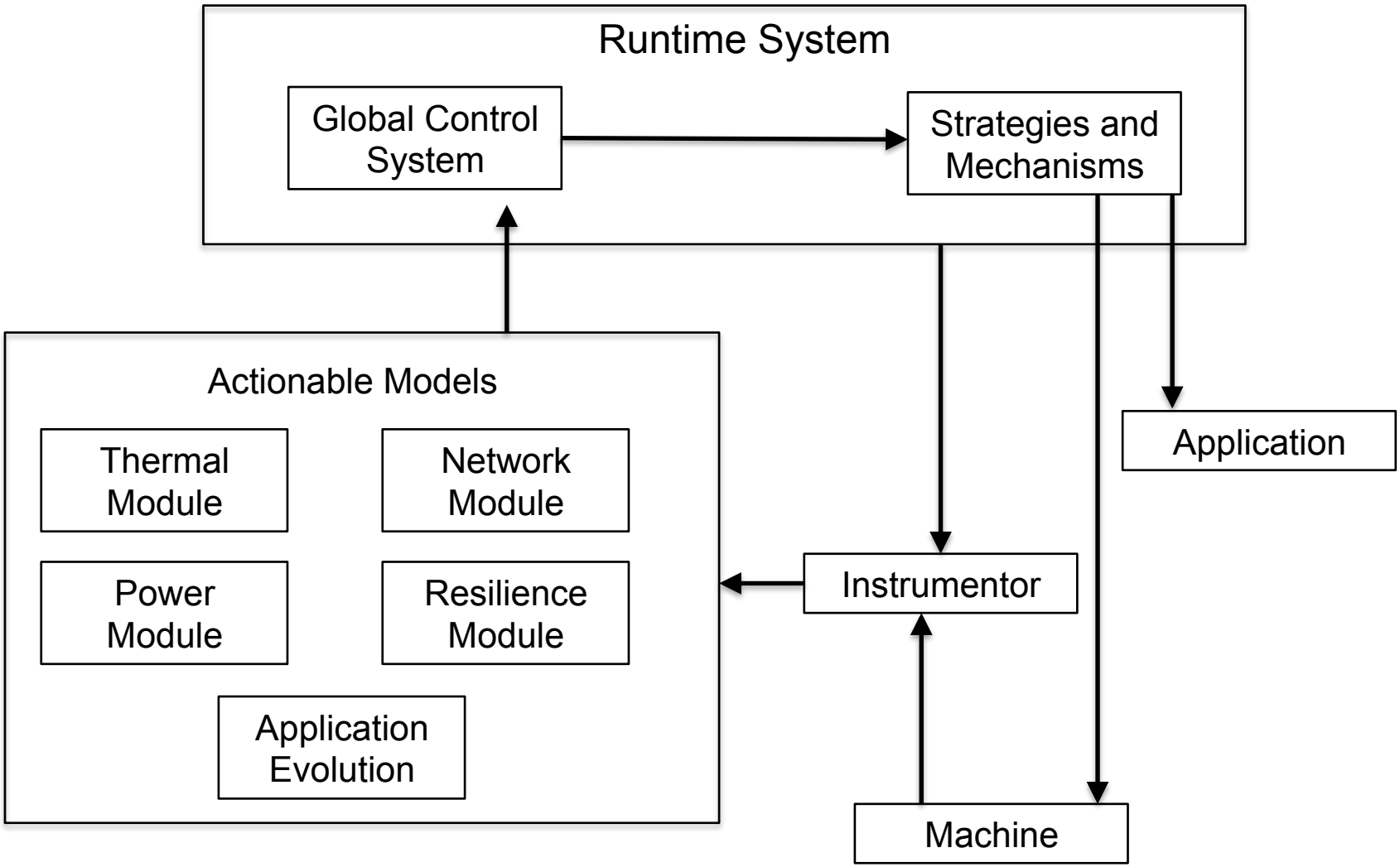


Figure : Interaction between actionable models, runtime system, application and machine.

Dynamic Workload Management

- Dynamic Load Balancing Systems (e.g. Charm++, Trilinos)
  - Provide simple models for predicting future load
  - Good, but insufficient for the dynamic exascale environment
- Necessary infrastructure improvements:
  - Fast and Accurate Load Balancing Strategies
  - Application Specific Behavior Models
  - Automatic Load Balancing Strategy Determination
  - Automatic Load Balancing Frequency Determination
    - ★ Cost benefit analysis for when (& how) to assess, decide, and apply load balancing
  - i.e., Metabalancer

Metabalancer Results

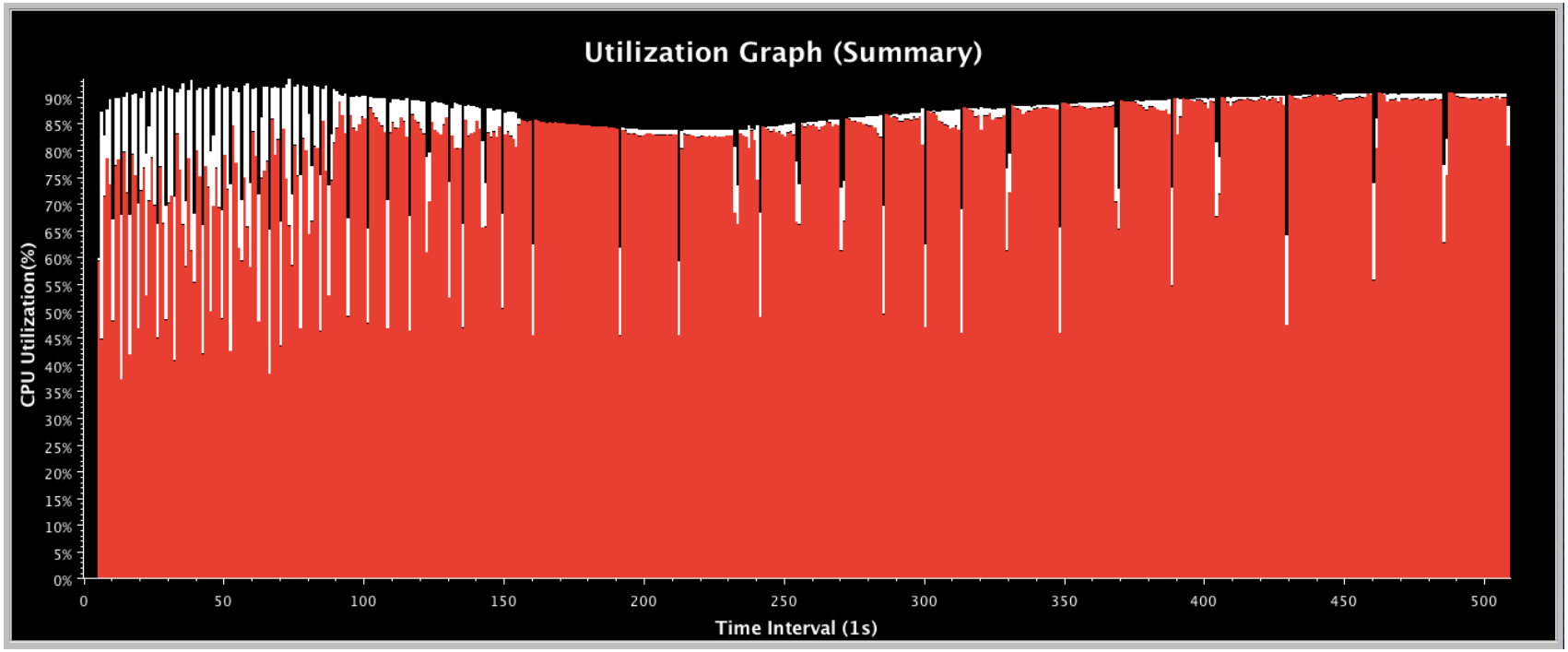


Figure : Introspective frequency of LB decision using Meta LB

Dynamic Power and Temperature Management

- Dynamic Voltage and Frequency Scaling (DVFS)
  - Runtime system can adjust frequency to restrict package and memory power consumption.
  - Reduce power use $->$ Reduced thermal output $->$ reduced cooling energy
- However, power and thermal variation across platform will create dynamic load imbalance.
  - May invoke dynamic load balance to compensate
  - Must be integrated with other load balancing considerations

HPC Goes On a Power Diet

20 MW Power Cap?
Whatever the actual number, there will be a power cap.

- Thermal Design Power (TDP) is the max power draw of a package or component
- Sum of all TDPs is the data center's theoretical peak power requirement
- Problem: Probably too much power at exascale

Solution: Over Provisioned Platforms

- TDP should be configurable for the key components
  - Already true for new architectures (e.g. Intel Sandy Bridge)
- Over-provisioned platform would have TDP $>$ data center power
  - Set TDP below peak
  - At good power/performance ratios
  - Within power budget

Optimizing Performance Within a Power Cap

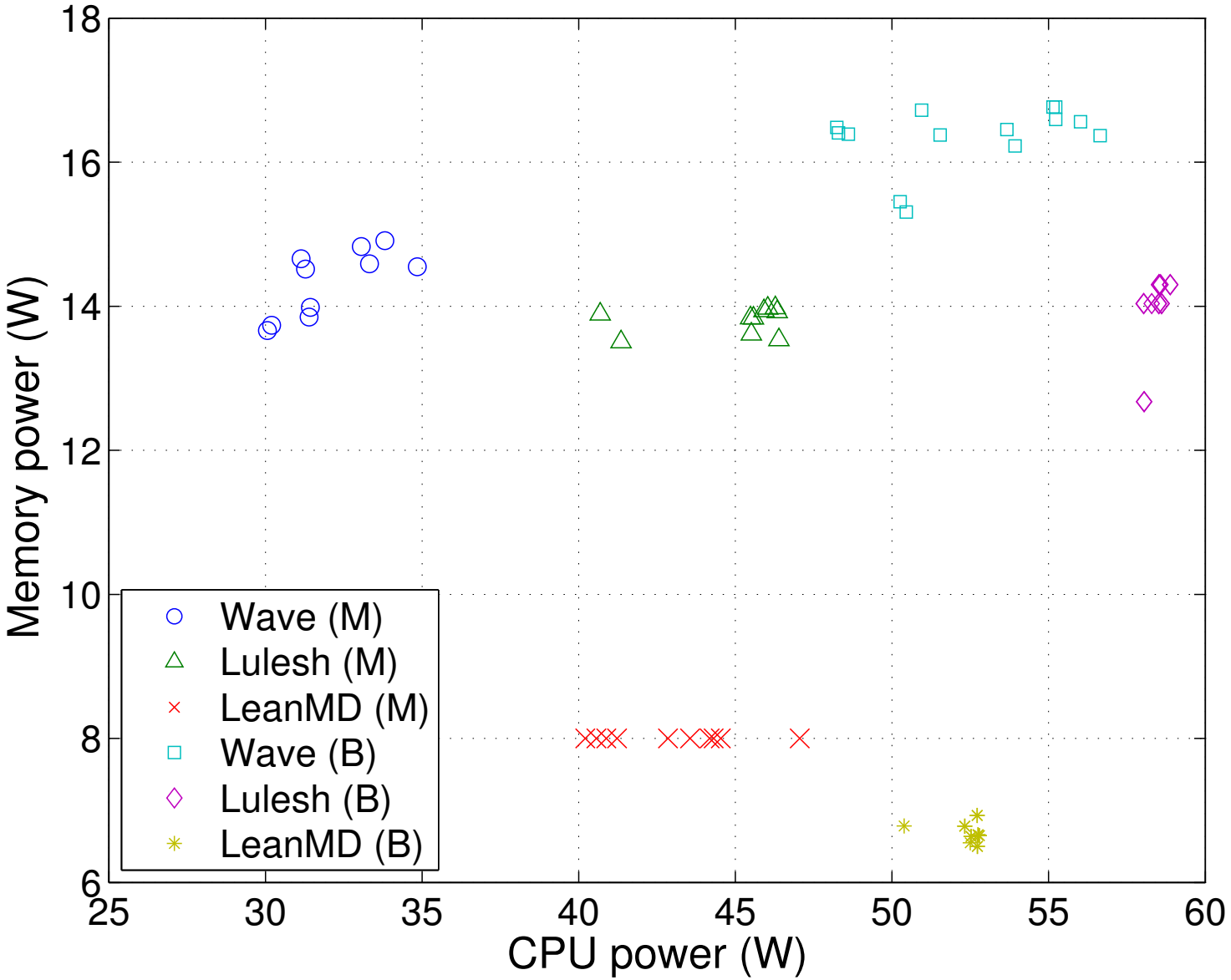
How to minimize time to solution without exceeding the cap?
Answer: Embrace non-uniformity!

- Power requirements vary over time within each application
- i.e., Intensive computation, synchronization, memory accesses, etc.
- Any application can be segmented into Sequential Execution Blocks
- SEBs have no remote dependency within them
- The Modeling system can profile SEBs to characterize their power characteristics
  - Perhaps at runtime, perhaps offline
  - Runtime model can assess SEBs at different power levels
  - Apply machine learning models
  - Build lookup tables of performance vs power by SEB
- Strategies may then be applied dynamically to optimally allocate power for each SEB

Adaptive Jobs: Optimizing Throughput Within a Power Cap

- Much as application phases can be characterized, so can entire applications.
- Resources can be scheduled to balance high and low power applications on the platform.
- Adaptive jobs may *shrink* or *expand* their allocation
  - based on scheduler commands for capping concerns
  - by application request to accomodate dynamic behavior (i.e. multiphysics transitions, complex workflows, speculative sub computations, re-meshing, adaptive mesh refinement etc.)
- Requires integration of linear optimization problem solving in job scheduler

Power Setting Optimality by Application



Improving Resilience

Petascale platform MTBF 2-11 hours
Exascale platform MTBF **35-40 minutes**?

- Failure rate of a node doubles for every 10°C increase in temperature
- Variation in temperature due to variation in power usage creates variation in fault frequency.
- Restraining core temperature will allow the data center operator to *choose* a reliability level
  - Selectable by application to improve reliability
  - Reliability model for a given allocation of resources can be improved by tracking temperature history of each component in the allocation
  - Relatively inexpensive to monitor temperature
  - Predictive models for failure not difficult to extend to include temperature history
  - Applications with high failure tolerance could use lower reliability resources (i.e. older nodes)

Temperature Aware Fault Tolerance

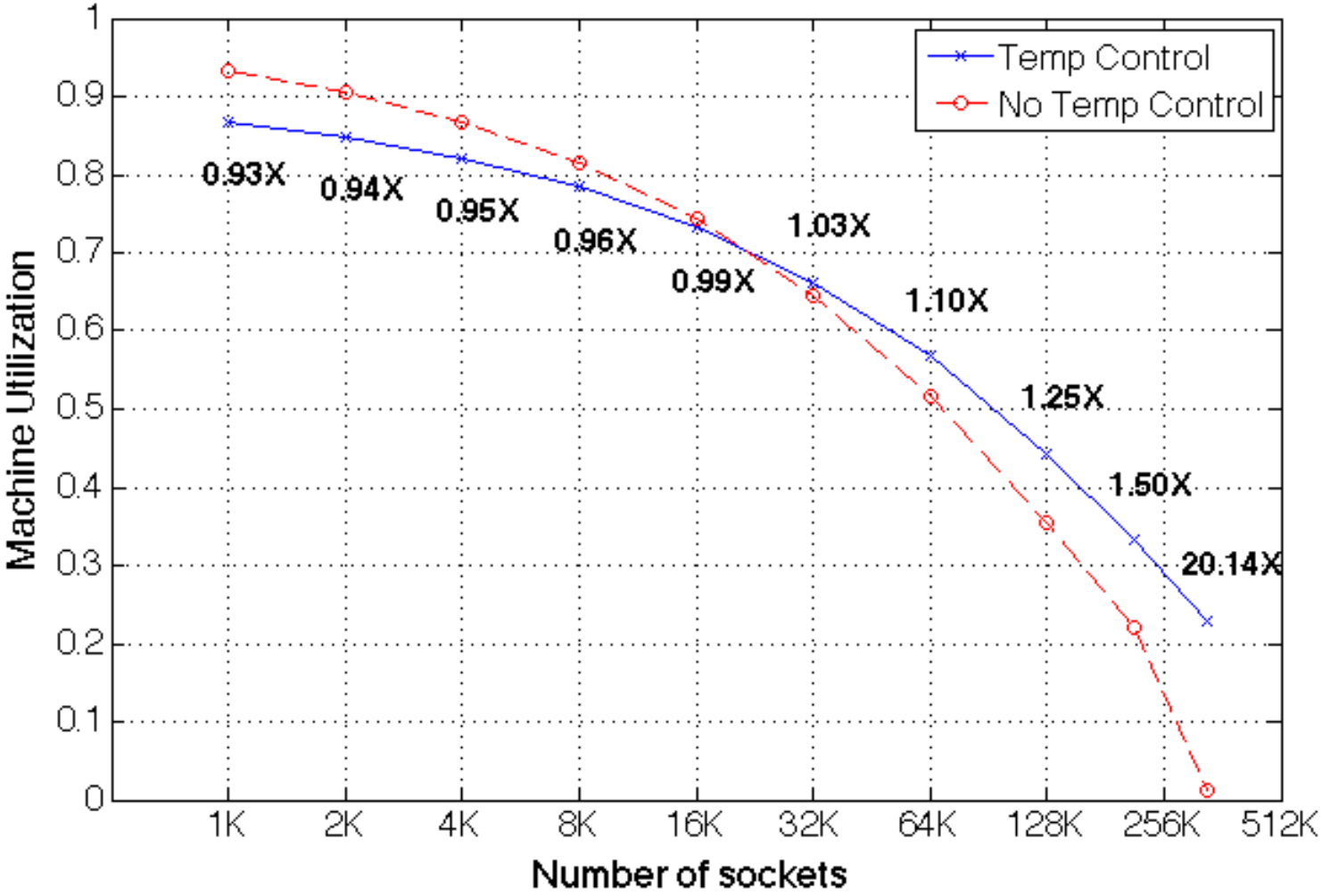


Figure : Estimation of Impact of Temperature Control on Improve Resilience

Automatic Network Topology Aware Placement

- Many applications exhibit patterns which benefit from network topology aware placement
- Introspective runtime can capture the communication graph(s) of applications
  - Integration with network modeling software can predict performance impact of alternate layouts
  - Congestion traits of the application and platform interaction can be captured
- Layouts with superior predicted performance may be applied by a dynamic runtime
- Superior overlay trees for collectives can be selected
- Alternate routing schemes can be requested

Conclusion

- Exascale platforms will present a variety of new challenges for both runtime systems and simulation.
- Emulation + Simulation can be used to capture application behaviors and study the impact of many different platform characteristics
- Specialized simulators can be applied post-mortem to gain insight wrt platform and algorithmic choices
- Multiple levels of resolution will be necessary to gain insights in reasonable time