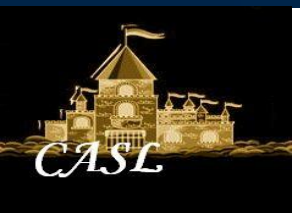


Architectural Performance Modeling for Exascale: Challenges and Opportunities

S. Yalamanchili

School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA. 30332

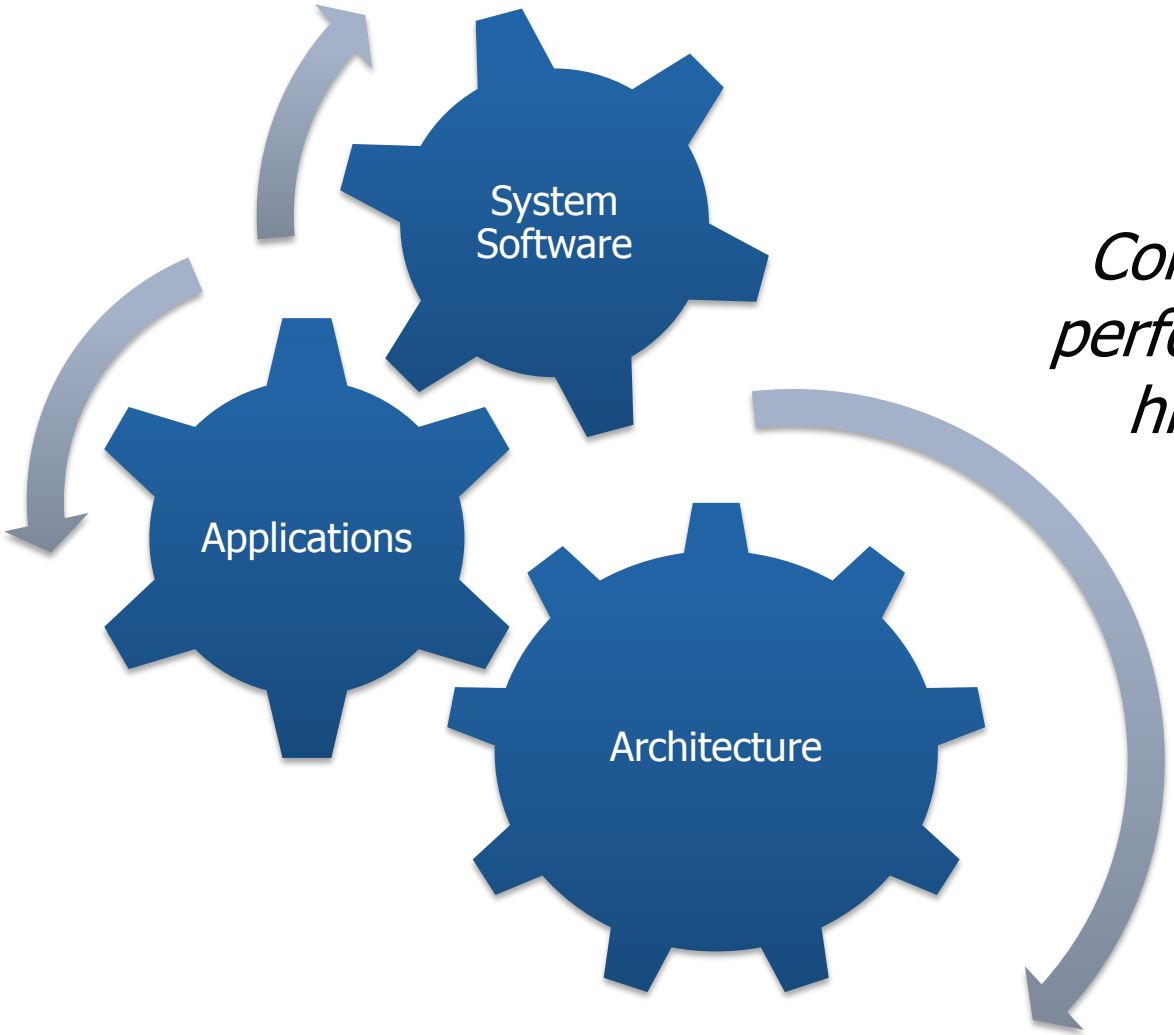
Sponsors: National Science Foundation, Sandia National Laboratories, and Semiconductor Research Corporation

Motivation

"If you can not measure it, you can not improve it."

Lord Kelvin

Goal



Construct integrated performance models of high performance systems

Needs and Capabilities

Need to distinguish between modeling and engineering

Modeling

- Performance models of complex phenomena
- Abstract behaviors of interest
- Draw upon a palette of mathematical and simulation techniques

Engineering

- Construction of software or hardware implementations
- Modularity, composition, interoperability
- Practical determinant of ease of use

Energy Scaling is the Key Driver

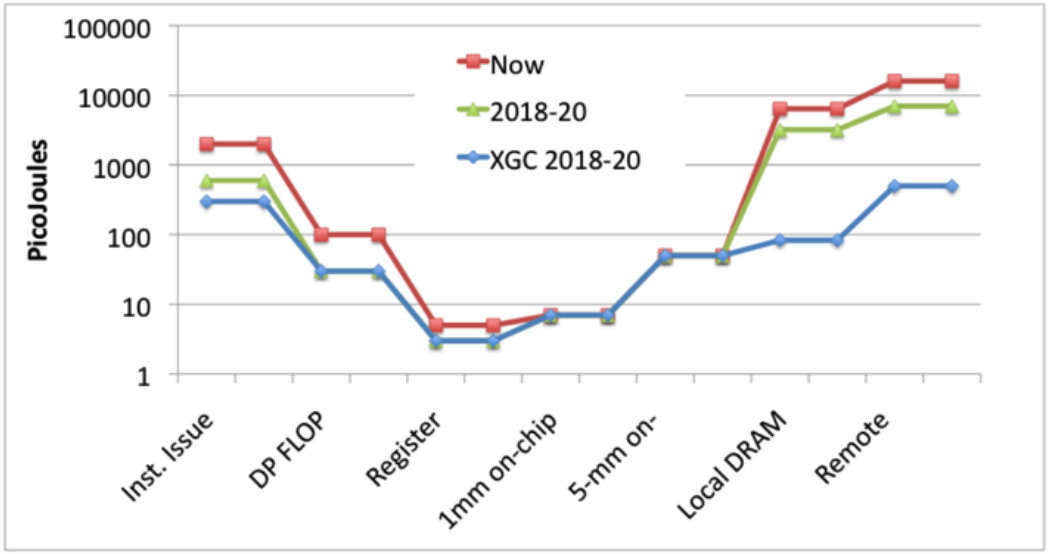
Embedded Platforms

Big Science: To Exascale

Goal: 1-100 GOps/w

Goal: 20MW/Exaflop

Cost of Data Movement

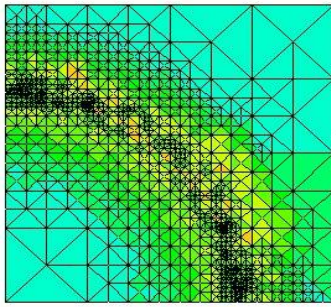


Courtesy: Exascale Grand Challenge Program (Sandia National Labs :R. Murphy).

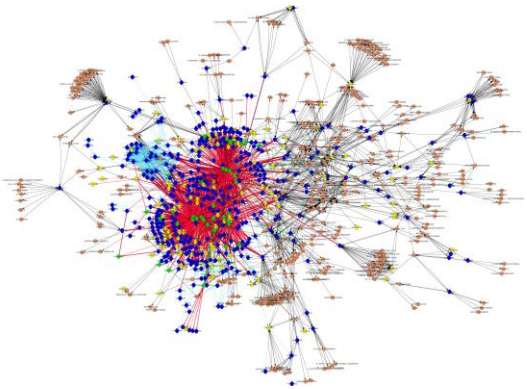
- Sustain performance scaling through massive concurrency
  - New execution models
- Data movement more expensive than computation

Fidelity vs. Scale

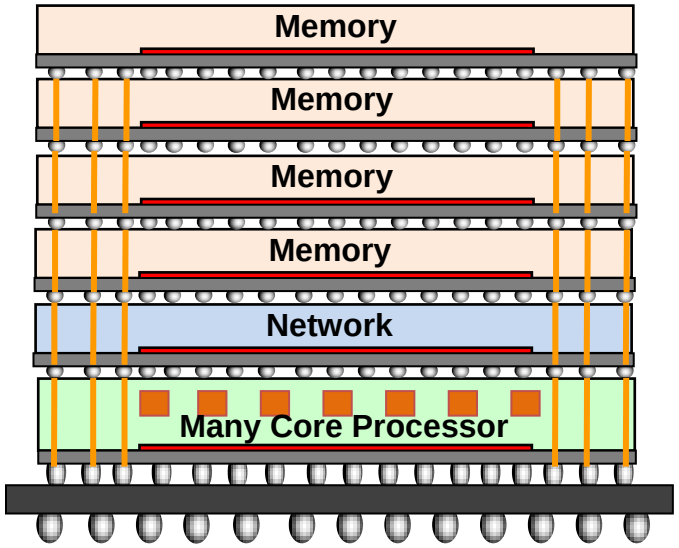
Multi-resolution



Large Graphs

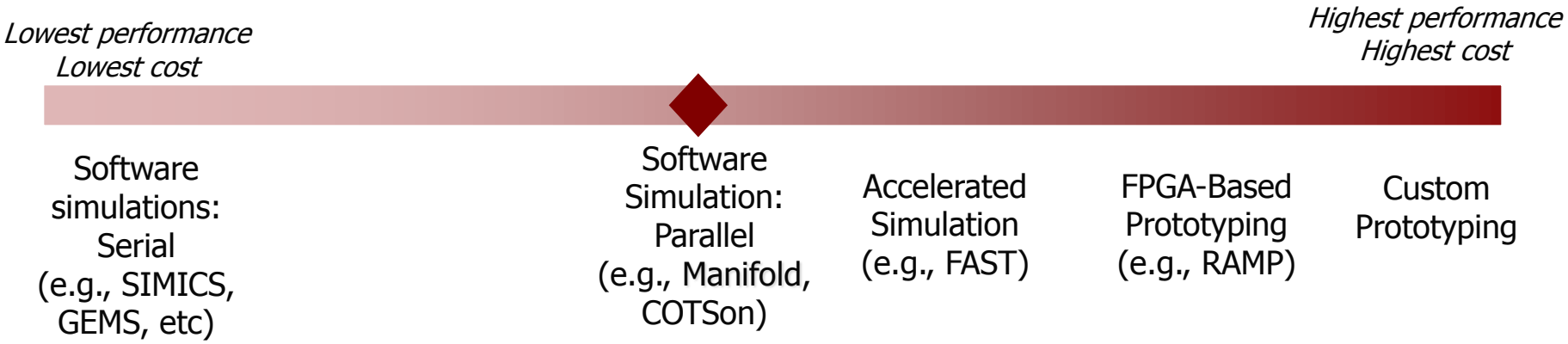


System Software



- Interactions between core, network, and memory hierarchy
- Full System Simulation
  - Boot OS, compiled application binaries, cycle-level timing models
- High Level Models
  - Analytic models of system behavior and application demand
- Intermediate Models
  - E.g., Trace-based, dependency-based, statistical etc.

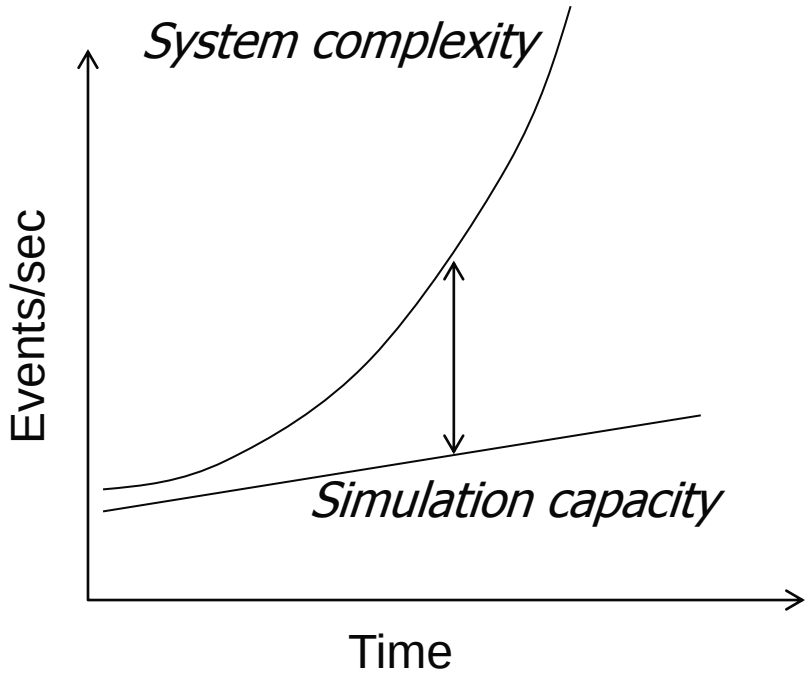
Spectrum of Solutions



- Models should be portable across methodologies
  - Mature point tools, e.g., cores, memory systems, etc.
- Cull the design space prior to committing to hardware prototyping or hardware acceleration strategies

The State of the Practice

- System complexity is outpacing simulation capacity
  - Cannot perform analysis at scale
- Islands of mature models/simulators
  - Monolithic, custom, not built for composition or re-use
- Need to move to the new performance scaling curve

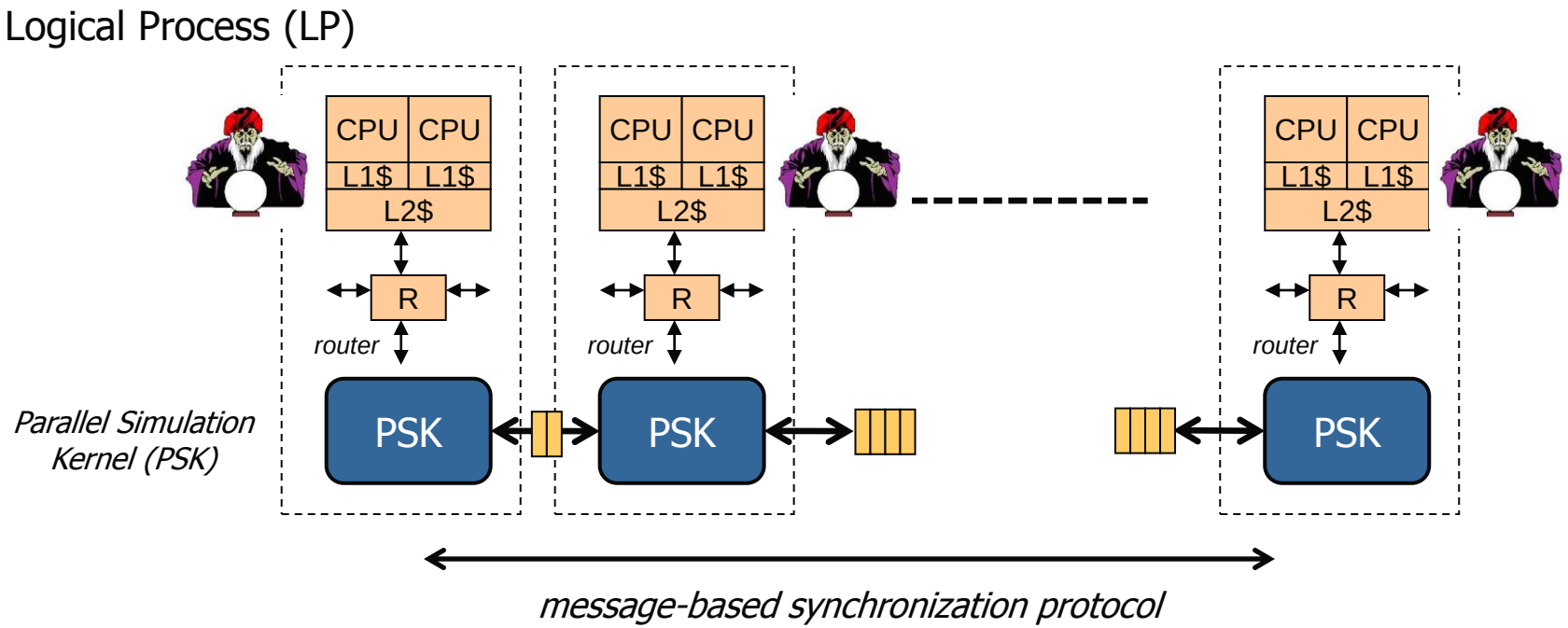


Need to scale simulation capacity

Challenges and Opportunities

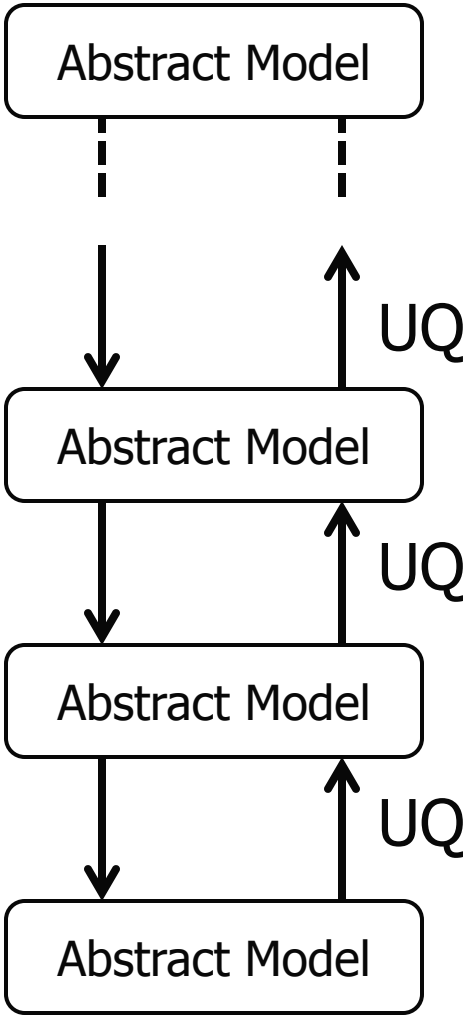
1. Parallelism
2. Model Accuracy and Validation
3. Heterogeneity and Parallelism
4. Multi-scale
5. Energy/Power Stack
6. Coupling Physical Models
7. Workloads and Characterization
8. Engineering a Simulator

1. Need for Parallelism



- Processors have moved to new performance scaling regime → parallelism
- Decouple parallel execution from the model

2. Model Accuracy and Validation

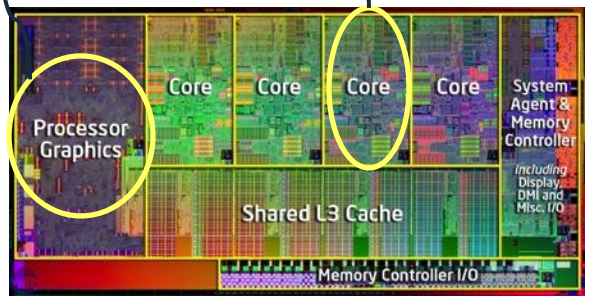


- How do we assess accuracy?
  - No baseline
  - Borrow concepts from scientific models
  - Uncertainty Quantification UQ)
    - Model, and parameters
- Validation
  - Trend analysis against real machines
  - Invariants
  - Prototyping
  - Model correlation techniques from EDA

3. Heterogeneity and Massive Parallelism

Programmable Pipeline (GEN6)

Vector Extensions AES Instructions



Intel Sandy Bridge

Programmable Accelerator

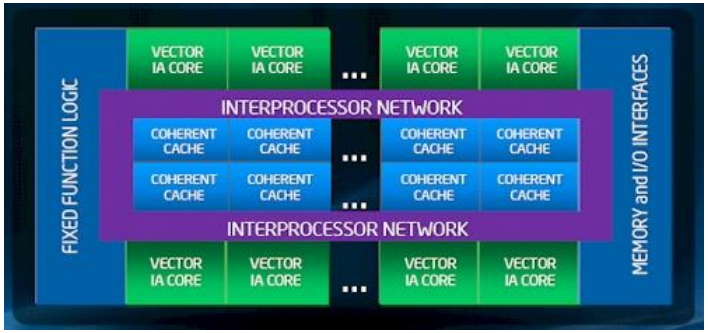
Multiple Models of Computation
Multi-ISA

16, PowerPC cores

Accelerators
- *Crypto Engine*
- *RegEx Engine*
- *XML Engine*
- *CP<[press Engine*

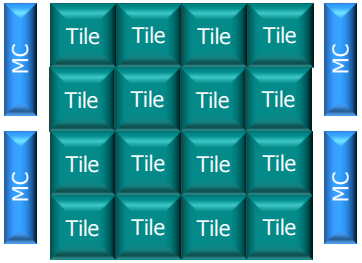
IBM *PowerEN*

Intel Knights Corner



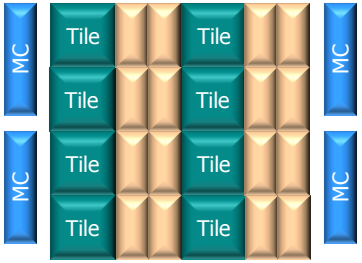
Asymmetry vs. Heterogeneity

Performance Asymmetry



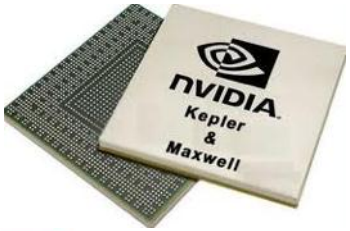
- Multiple voltage and frequency islands
- Different memory technologies
  - STT-RAM, PCM, Flash

Functional Asymmetry



- Complex cores and simple cores
- Shared instruction set architecture (ISA)
  - Distinct microarchitectures

Heterogeneous



- Multi-ISA
- Microarchitecture
  - Memory & Interconnect hierarchy

Uniform ISA

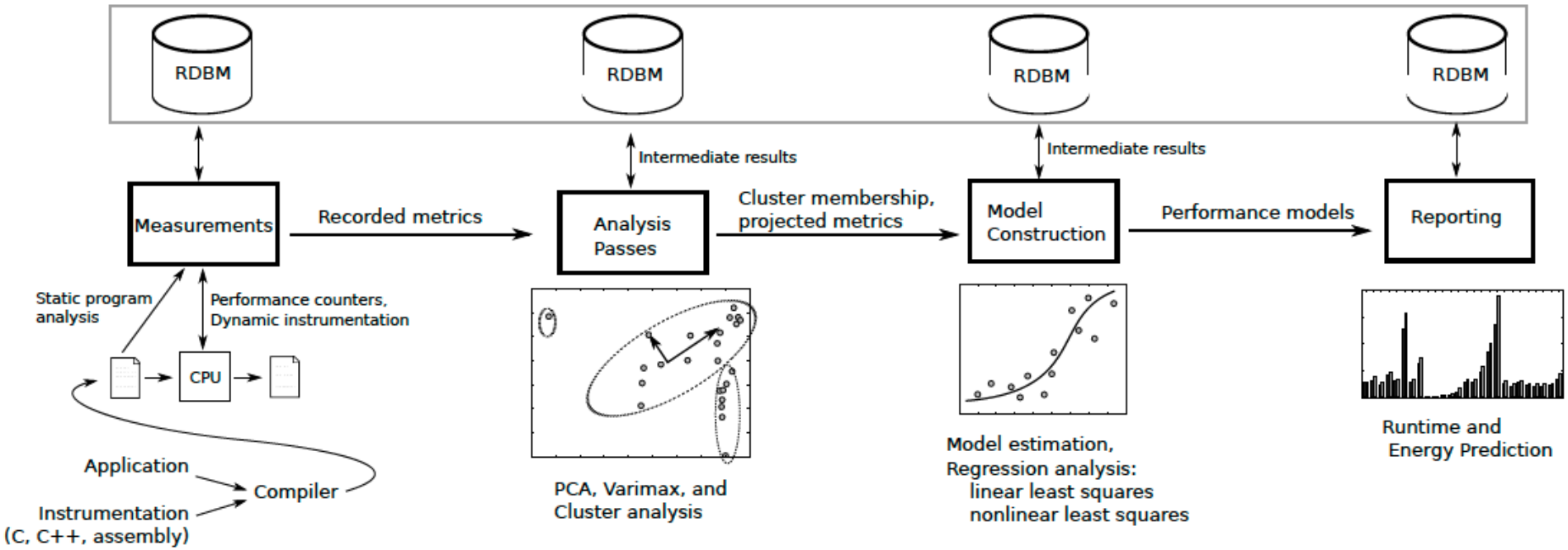
Multi-ISA

[1]Li., T., et.al., "Operating system support for shared ISA asymmetric multi-core architectures," *in WIOSCA,* 2008.

4. Multi-Scale

- Hierarchy of application representations
  - Applications → mini-apps → kernels → traces → analytic models
- Hierarchy of Architecture models
  - Macro → meso → micro → RTL → gate → device
- Need abstract models of key features
  - Time, space, and energy
- Ability to trade fidelity for scale
- Hybrid simulation
  - Time stepped, discrete event, continuous time
  - Getting to the end of CMOS

Eiger: Performance Model Synthesis



- Collect instrumented data on simulators, FPGAs, or real implementations
- Data analysis to synthesize performance models
  - Execution time, energy, etc.

CODEX: A Hardware/Software Co-Design Process for the Exascale Era

5. Energy/Power Stack

Applications

Libraries

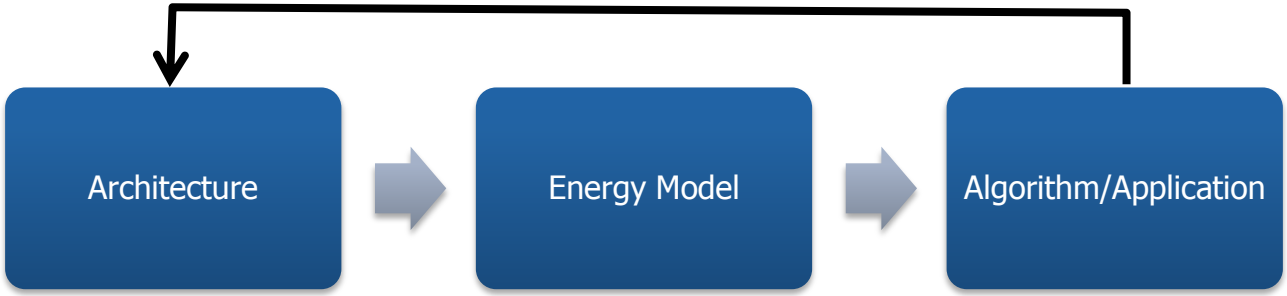
System Software

Microarch

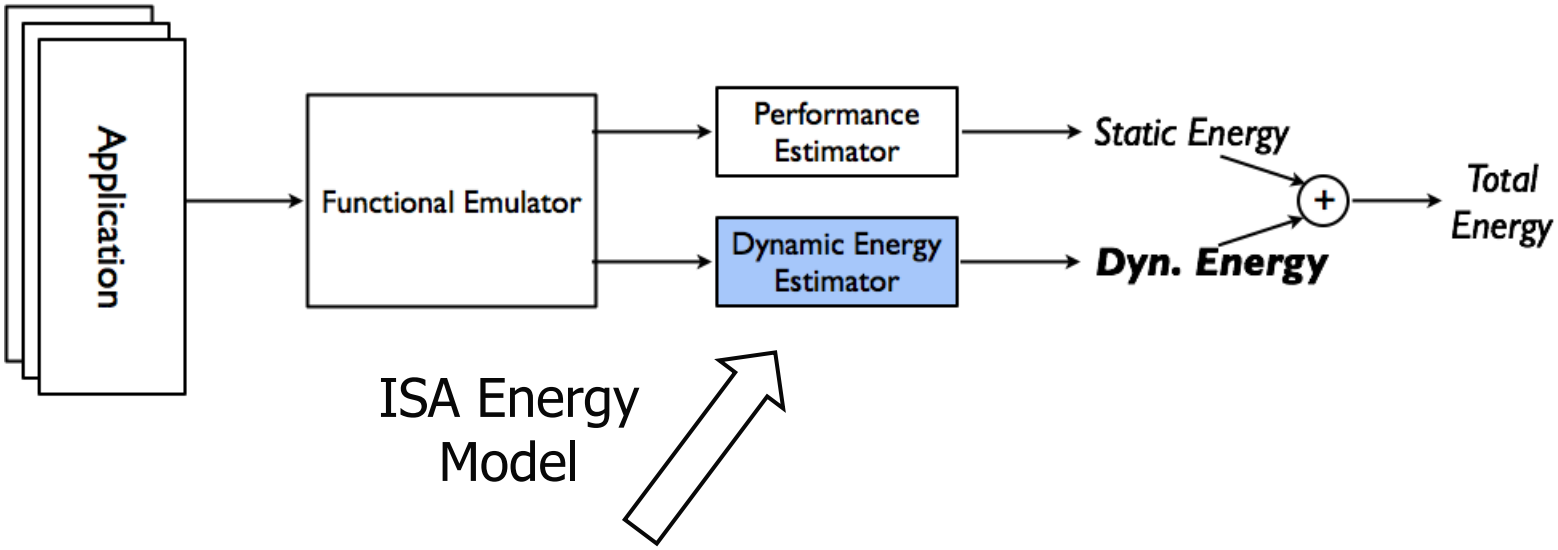
RTL

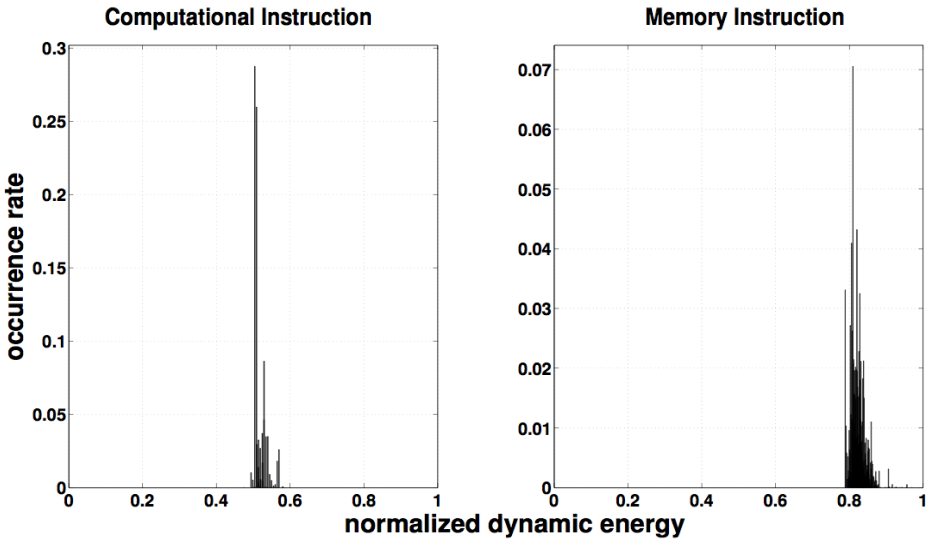
Circuit

- Need robust energy/power models at all levels of abstraction
- Need to be able to audit the energy/power behavior of software and hardware
- How can we promote industrial collaboration while protecting IP?
- Creation of energy/power models that can be used by algorithm and software developers



Instruction-Level Energy Modeling

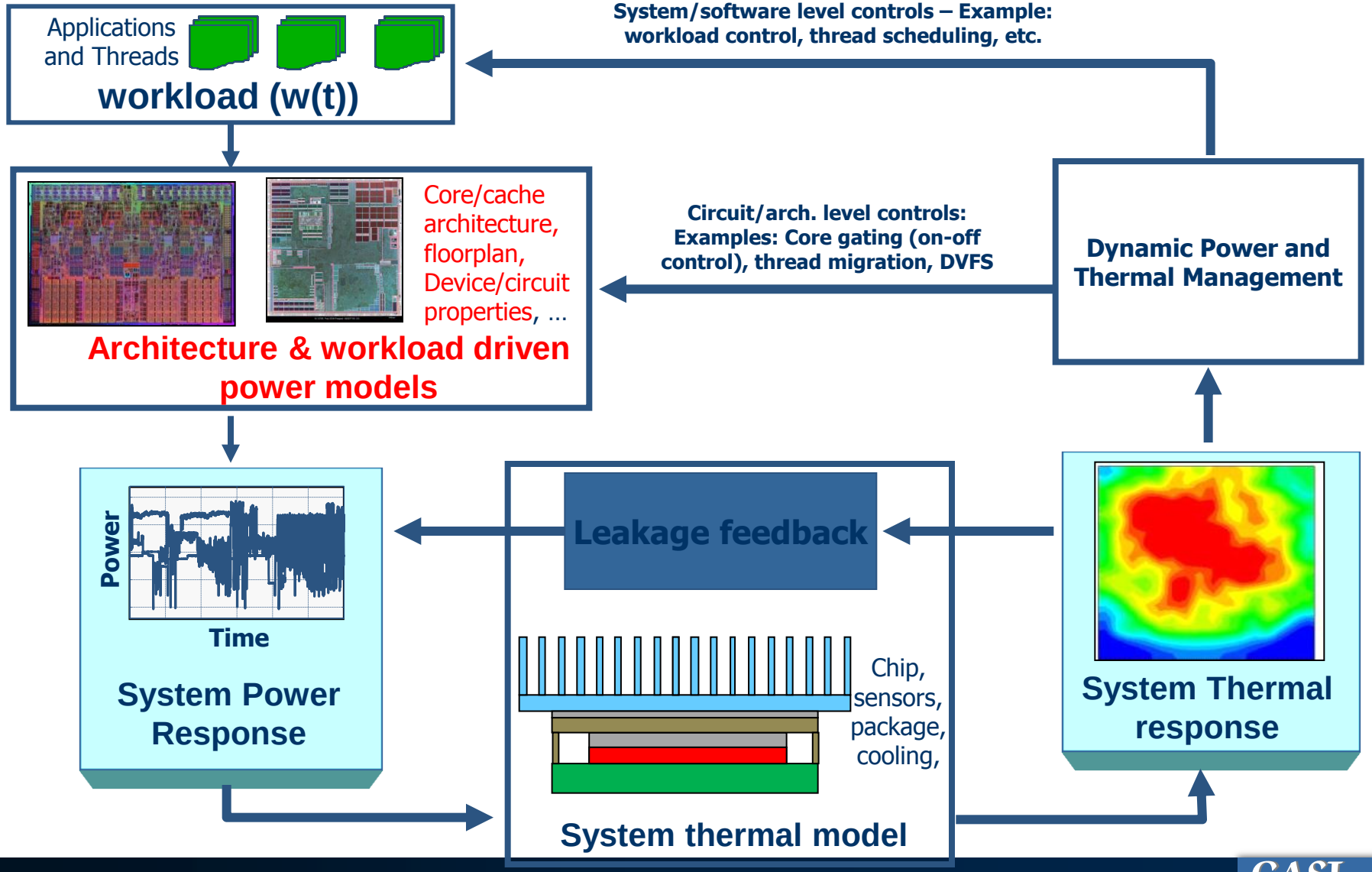




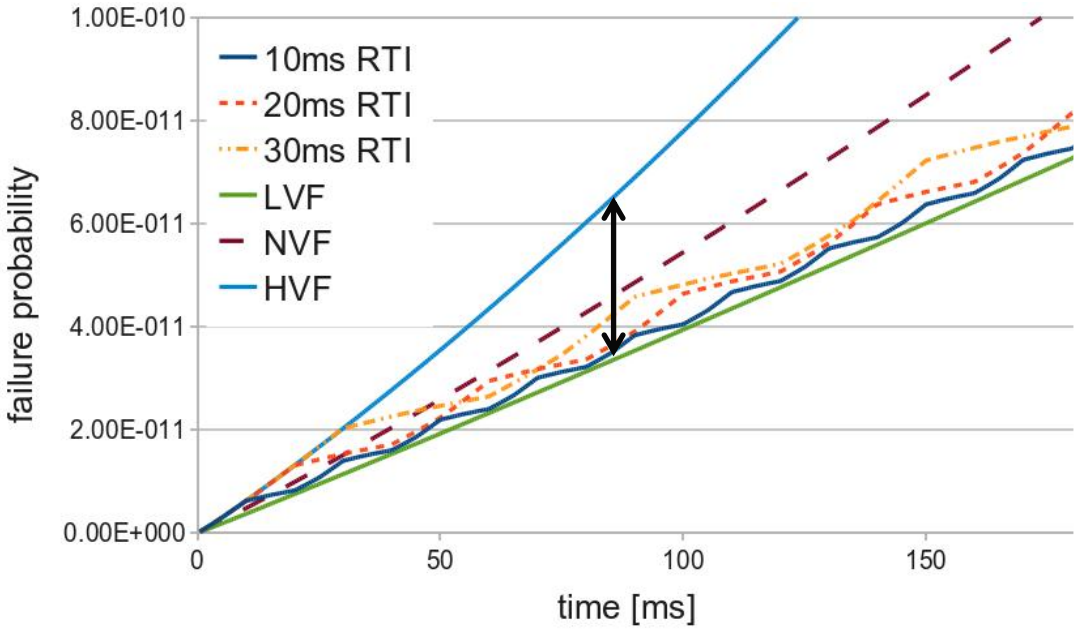
Dynamic energy variation of instructions is smaller and easier to predict by analyzing the execution datapath.

6. Coupling Physical Models

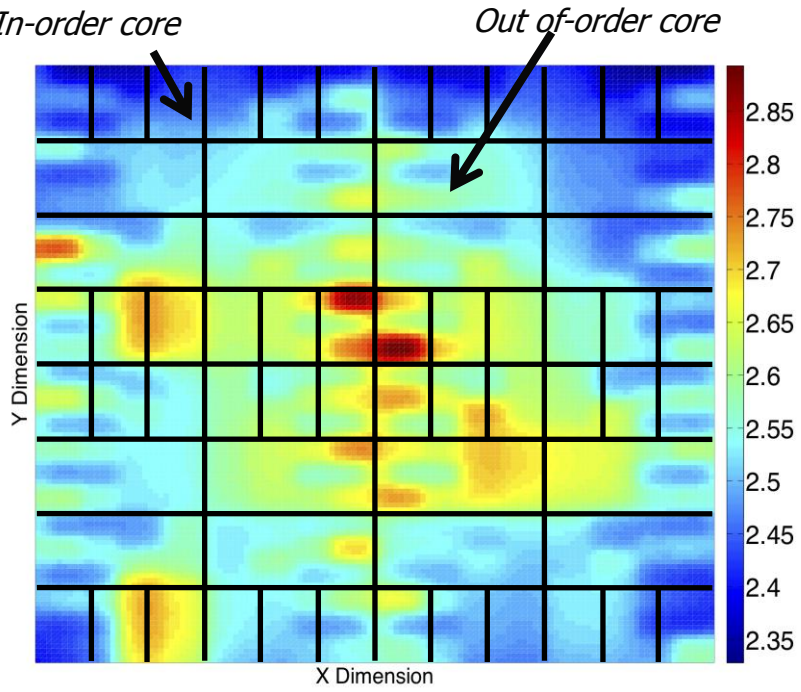
Joint work with S, Mukhopadhyay
Sponsors: SNL, SRC



Interaction Between Reliability and Thermal Fields



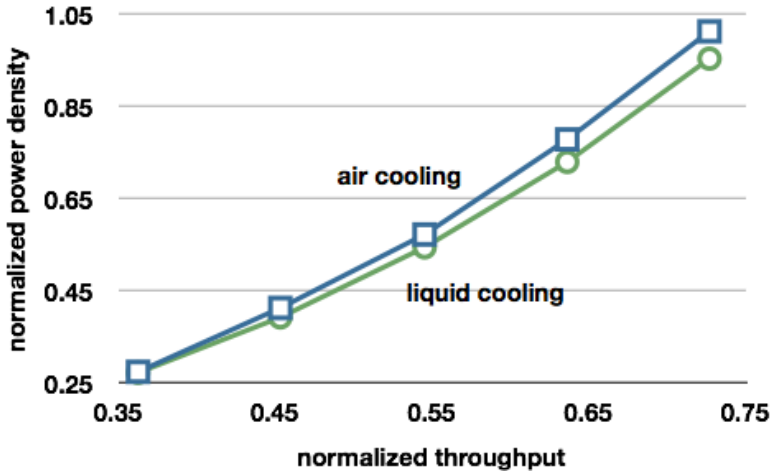
Failure probability trace of a grid cell with variable race-to-idle periods, compared to three continuous execution modes; LVF (0.8V, 2.0GHz), NVF (1.0V, 3.0GHz), and HVF(1.2V, 4.0GHz)

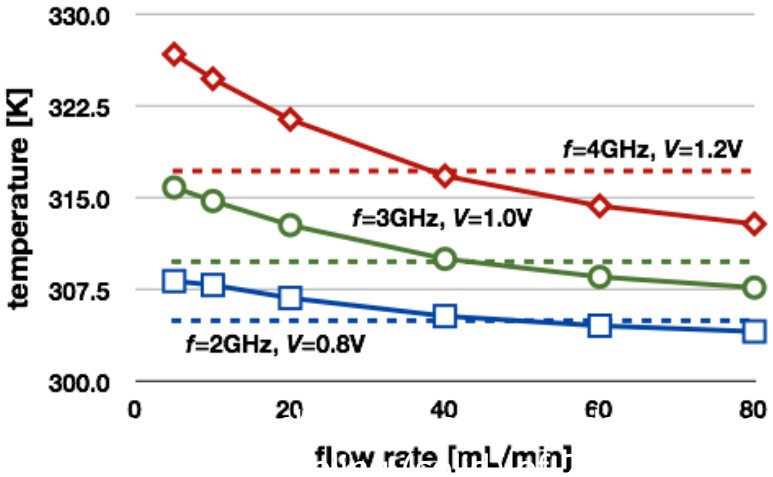


64-core asymmetric chip multiprocessor layout and failure probability distribution

- Non-uniformity in workload leads to non-uniformity in degradation across the die
- Amplified by non-uniform core types

Interaction Between Cooling and Power





Simple Testing Results

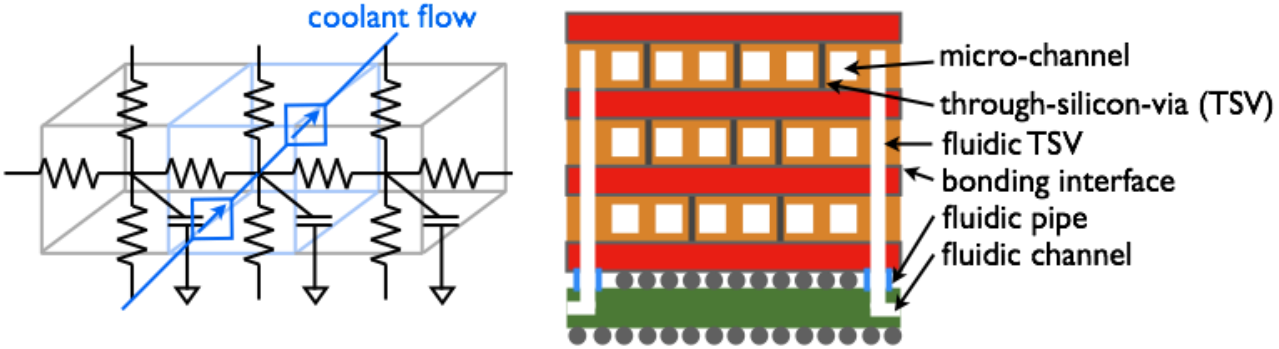
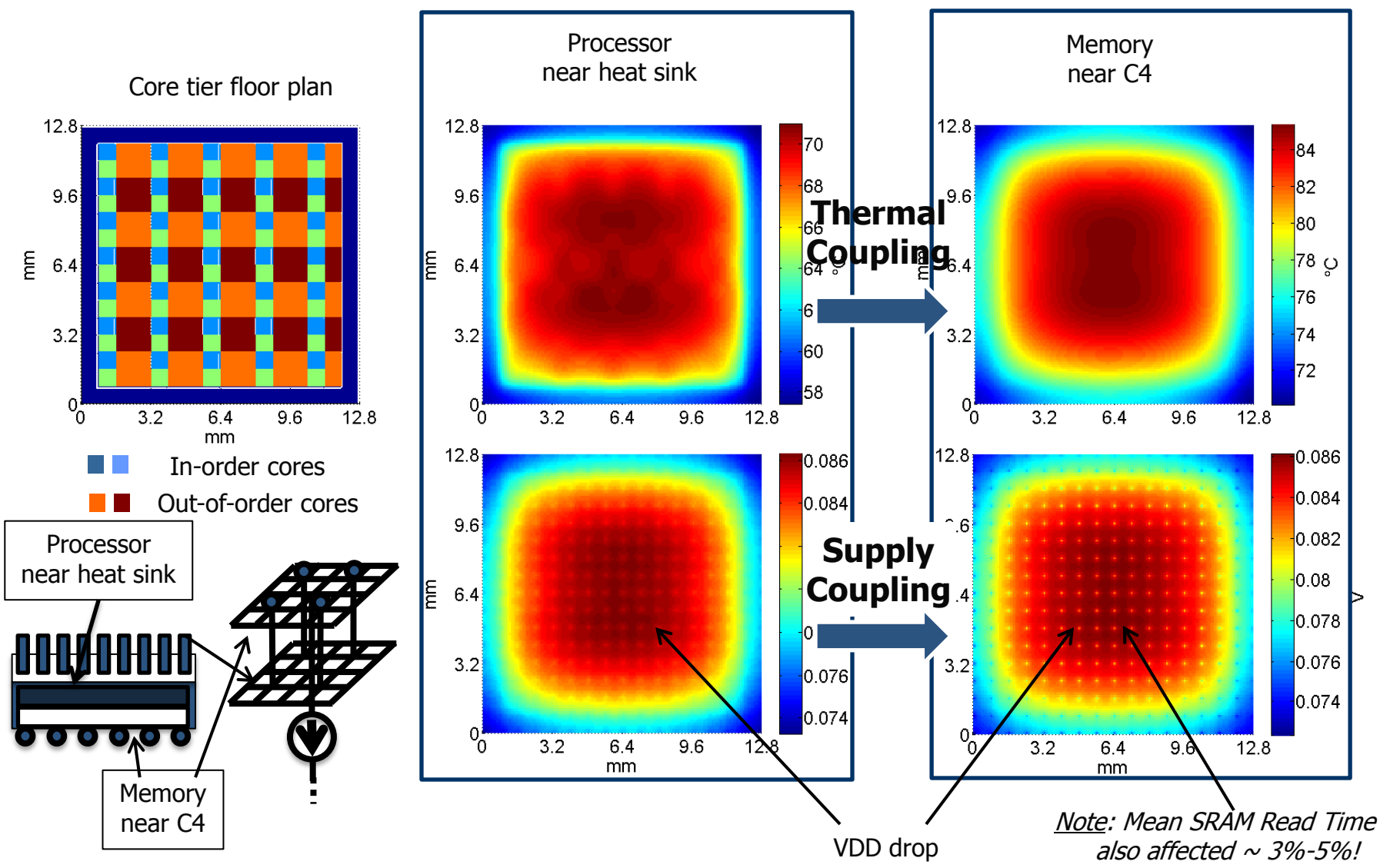


Figure. A 3D thermal grid cell and package model with liquid cooling, and partitioned die stacks.

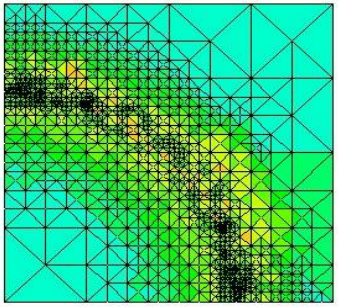
Energy Introspector: Modeling of power, thermal, energy, reliability and cooling in multicore designs

Illustrative Results: 3D Coupling

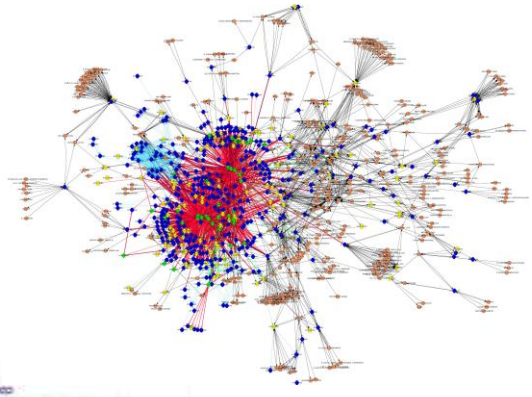


7. Workload Modeling & Characterization

Multi-resolution

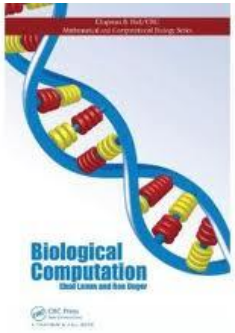


Large Graphs



C++ AMP

Database and Data Warehousing



- Emergent applications are increasingly irregular, unstructured and hard to predict
- Workload models should be multifaceted, and multiple levels of resolution

8. Engineering a Simulator

1. Cost of building a validated useful simulator
   - Composable
   - New methodologies for building simulators
2. Accuracy
   - Need for calibrated models
   - Methodologies for constructing calibrated models
3. Performance
   - Parallelism, multiscale, and hardware acceleration
4. Power and thermal models
5. Ease of use: Productivity and Management Tools
   - Visualization, deployment, debugging, etc.
   - Documentation & deployability

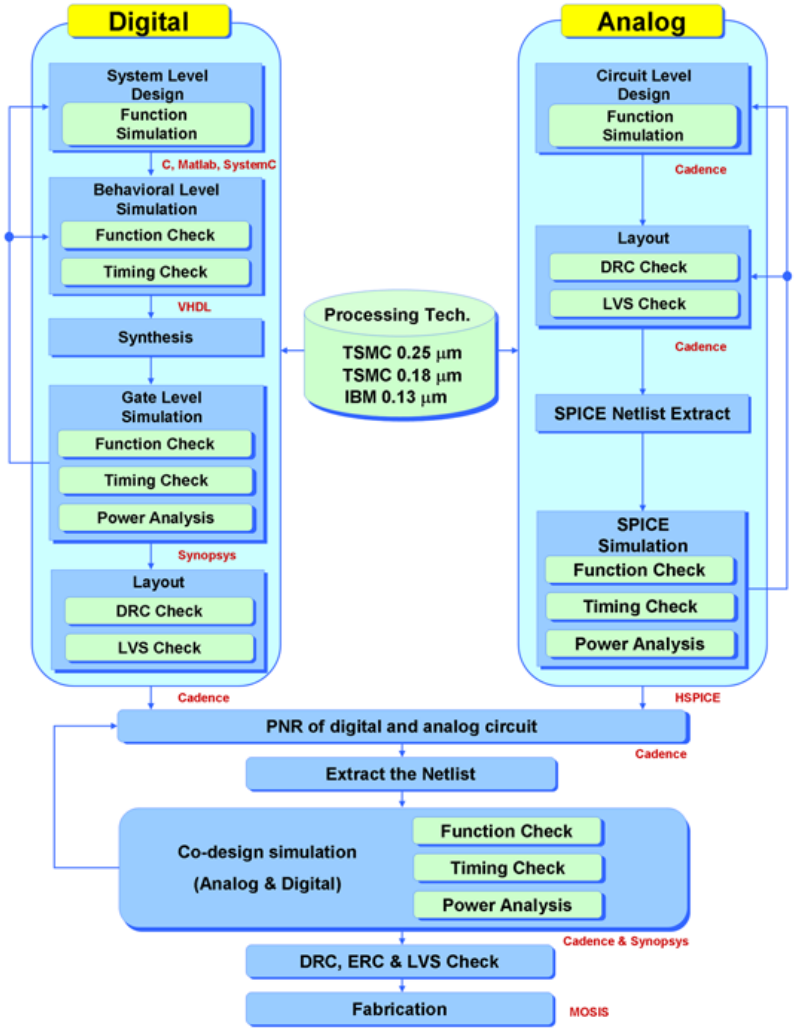
*From Outbrief: *Performance Prediction and Simulation for Exascale Interconnection Networks,* Interconnect Workshop, DoE Institute for Advanced Architectures, July 2008

Composability and Portability

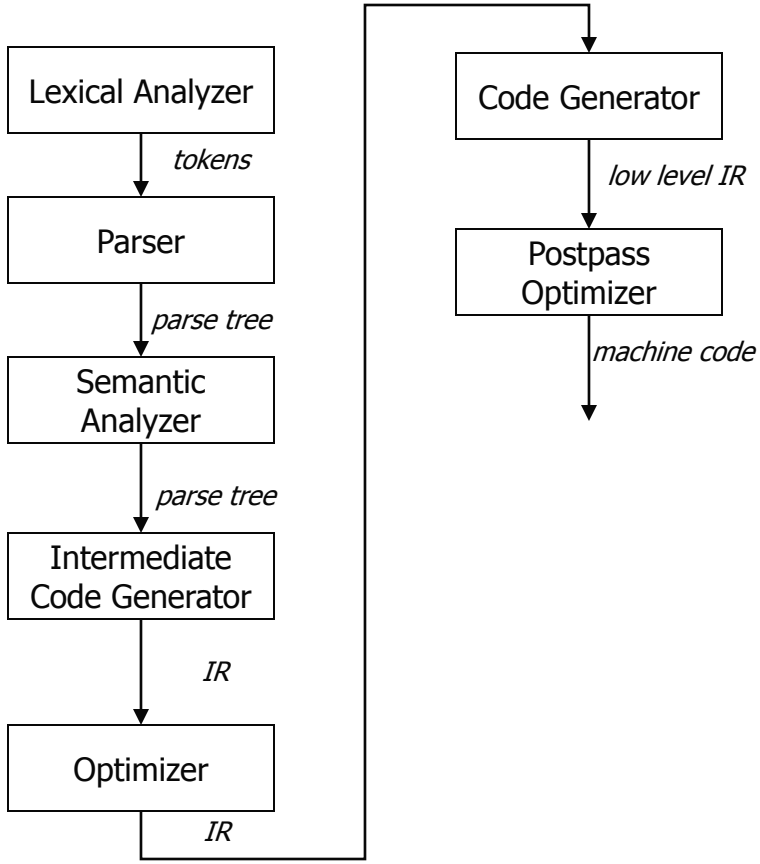
- Separate timing, synchronization, and event management from models → simulator kernels
- Model composition
  - Assemble system models from component models
  - Integrate third party tools
  - Trade-off fidelity vs. scale
- Leverage extensive depth of point tools
  - Interactions with industry
- Stability, testing, regression, documentation, release engineering

A Thought: Learn from Design Flows

EDA



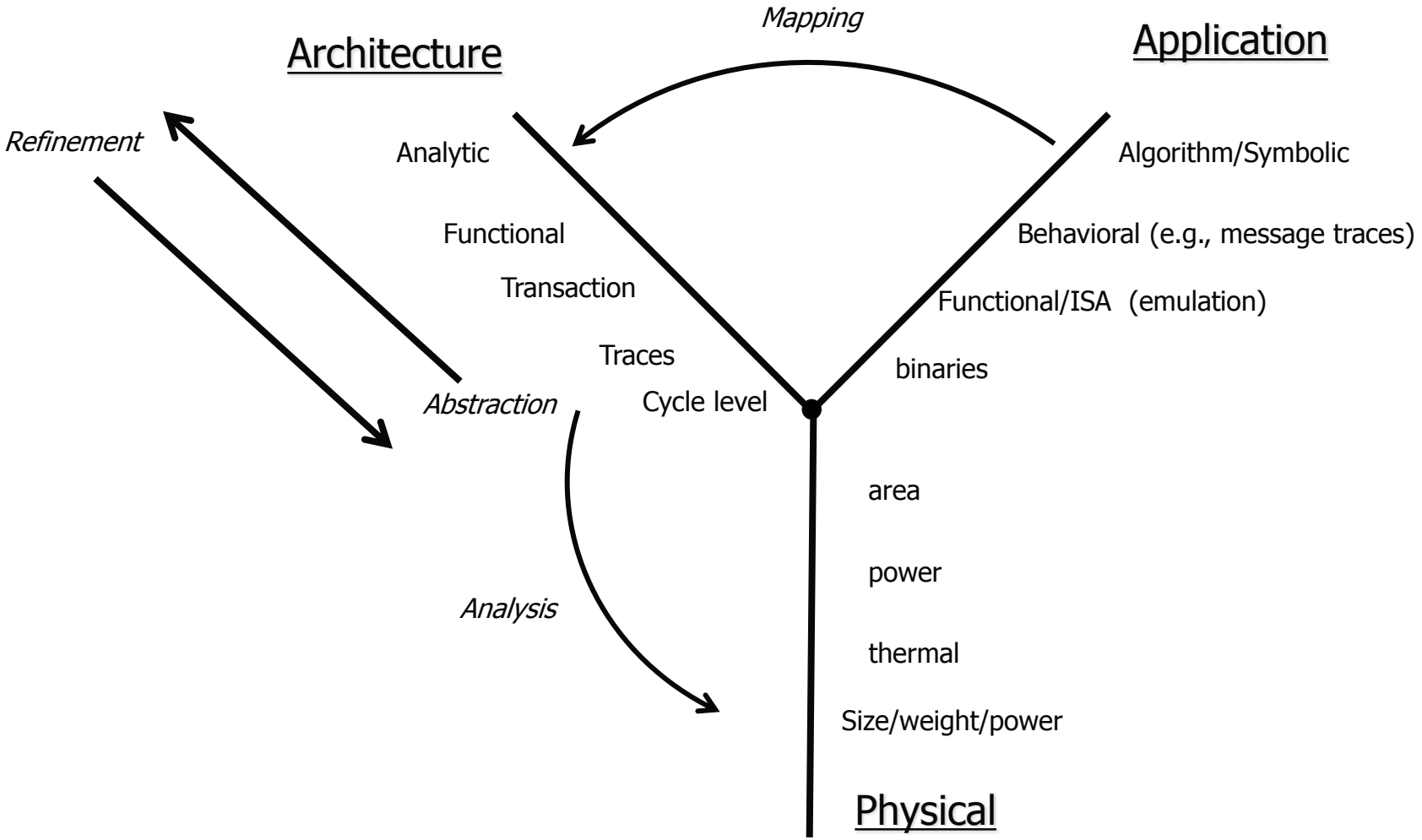
Optimizing Compiler



Implications

- Need a hierarchy of representations
  - Accompanied by successive refinement
- Some example simulation flow steps
  - Parsing a system description language
  - Model selection & instantiation
  - Interconnection
  - Partitioning & assignment,
  - Design rule (model) check

Example Y-Chart Based Design for Simulation



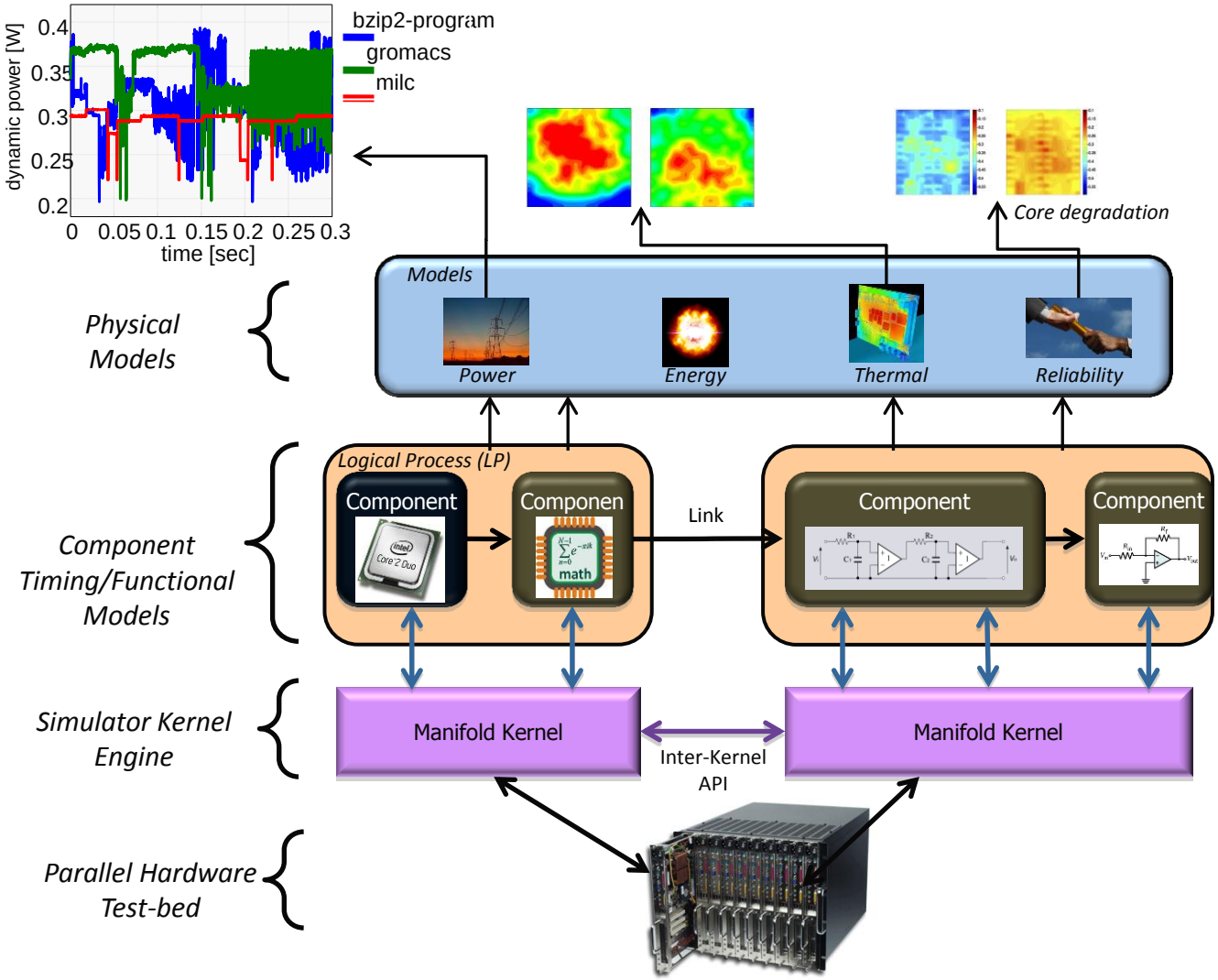
Based on the Y-Chart from D. Gajski, Silicon Compilers,Addison Wesley, 1987.

Manifold

S. Yalamanchili, T. Conte, G. Riley

Sponsors: NSF, Sandia Labs, HP Labs, Oracle

A composable parallel simulation system for heterogeneous, many core systems.



- Component-based and extensible
- Mixed discrete event and time stepped simulation
- From full system HW/SW models to abstract timing models
- From detailed cycle-level to high level analytic models
- Integration of third party tools

Summary: Challenges and Opportunities

1. Parallelism
2. Model Accuracy and Validation
3. Heterogeneity and Parallelism
4. Multi-scale
5. Energy/Power Stack
6. Coupling Physical Models
7. Workloads and Characterization
8. Engineering a Simulator

Thank You

Questions?