




 National Science Foundation  
WHERE DISCOVERIES BEGIN

 Sandia National Laboratories
 

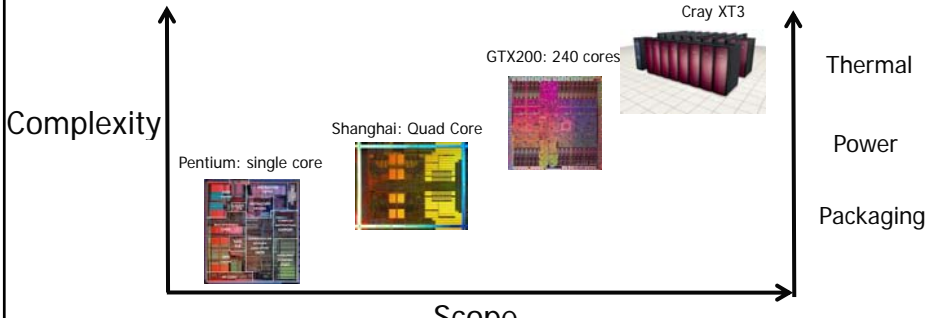
## Scalable Simulation Frameworks

Sudhakar Yalamanchili, George Riley, Tom Conte, Hyeosoon Kim  
and many students....

Center for Experimental Research in Computer Systems  
School of Electrical and Computer Engineering and School of Computer Science  
Georgia Institute of Technology

### Modeling and Simulation Demands



- System complexity is outpacing simulation capacity
  - Cannot perform analysis at scale
- The problem is getting worse faster → **Simulation Wall**
- Islands of simulators and simulation systems
  - Customized interactions
  - Difficult to leverage individual investments

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING | GEORGIA INSTITUTE OF TECHNOLOGY

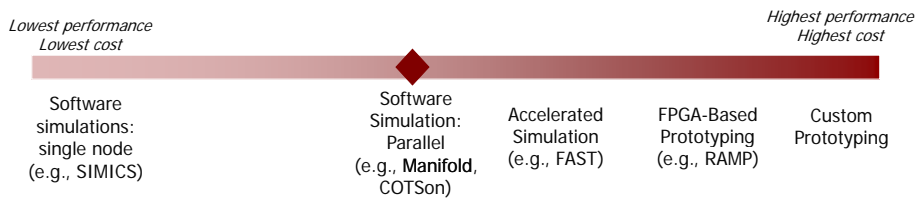
CASL 2

## Prioritized Major Challenges\*

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*, (Co-Chairs C. Janssen (SNL) and S. Yalamanchili (GT) Interconnect Workshop, DoE Institute for Advanced Architectures, July 2008

## Spectrum of Solutions



- Simple Premise: Use parallel machines to simulate/emulate parallel machines
- Leverage mature point tools via standardized API for common services
  - Event management, time management, synchronization
- Support Sandia's Structural Simulation Toolkit (SST)

## Coarse Grain Parallel Simulation

Example Modeled System

**Key Challenges**

1. Exploit program semantics
2. Exploit architecture behaviors

1. D. Burger and D. Wood, "Accuracy vs. Performance in Parallel Simulation of Interconnection Networks", *ICPP* 1995
2. A. Falcon P. Faraboschi D. Ortega, "An Adaptive Synchronization Technique for Parallel Simulation of Networked Clusters," *ISPASS*, 2008
3. Parallel SST, SNL

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING | GEORGIA INSTITUTE OF TECHNOLOGY CASL 5

## Separation of Timing and Functionality

- Granularity of the timing model & time synchronization
  - Sampling interface
- Direct interface into the code cache
- Timing model feedback
  - Synchronized advance of functional and timing models rather than roll-back and recovery
- Modeling I/O

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING | GEORGIA INSTITUTE OF TECHNOLOGY CASL 6

## Manifold Execution Model *NSF Manifold: S. Yalamanchili, T. Conte, G. Riley*

The diagram illustrates the Manifold Execution Model across two cores, Core 0 and Core 1. On Core 0, a 'Component' contains a 'VM' (Virtual Machine) and a 'Timing' block, connected by a 'VM - Timing API'. This component interacts with a 'Manifold Kernel' via a 'Kernel - Component API'. On Core 1, three 'Component' blocks are connected sequentially, with a 'Non-timing API' connecting the first and last components. Each component on Core 1 also interacts with its own 'Manifold Kernel' via the 'Kernel - Component API'. The two 'Manifold Kernel' blocks are connected to each other via an 'Inter-Kernel API'.

- Components, virtual machines, models, and logical processes
- Scalable parallel kernel to manage progression of time
- APIs → key to integrating mature point tools
  - Introspector API not shown (later)

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING | GEORGIA INSTITUTE OF TECHNOLOGY **CASL** 7

## Extensions to the Core Framework

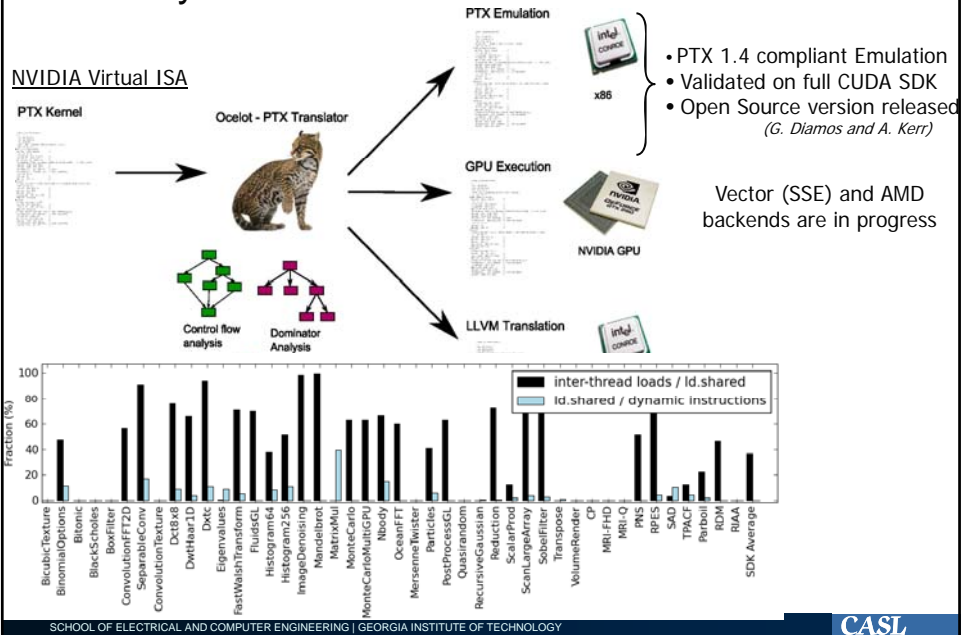
*Interface to Thermal Models (joint with S. Mukhopadyay)*

The diagram shows the core framework (Component, Manifold Kernel) being extended. Red arrows point from the component chain to three different model types:

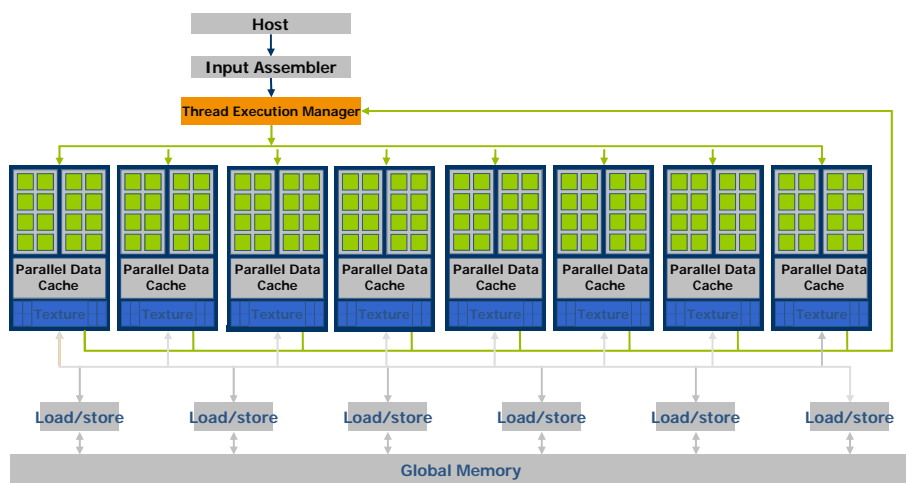
- Interface to Thermal Models (joint with S. Mukhopadyay):** Represented by two heatmaps showing thermal distribution.
- GPGPU Models (H. Kim):** Represented by a detailed block diagram of a GPU architecture, including MP (Multiprocessor), DRAM, Interconnect, Texture, Constant Cache, Instruction Cache, Memory Interface, Warp State, Warp Scheduler, Register File, and Shared Memory.
- Fast Simulation Models:** Represented by a 3D rendered image of a dragon-like creature.

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING | GEORGIA INSTITUTE OF TECHNOLOGY **CASL** 8

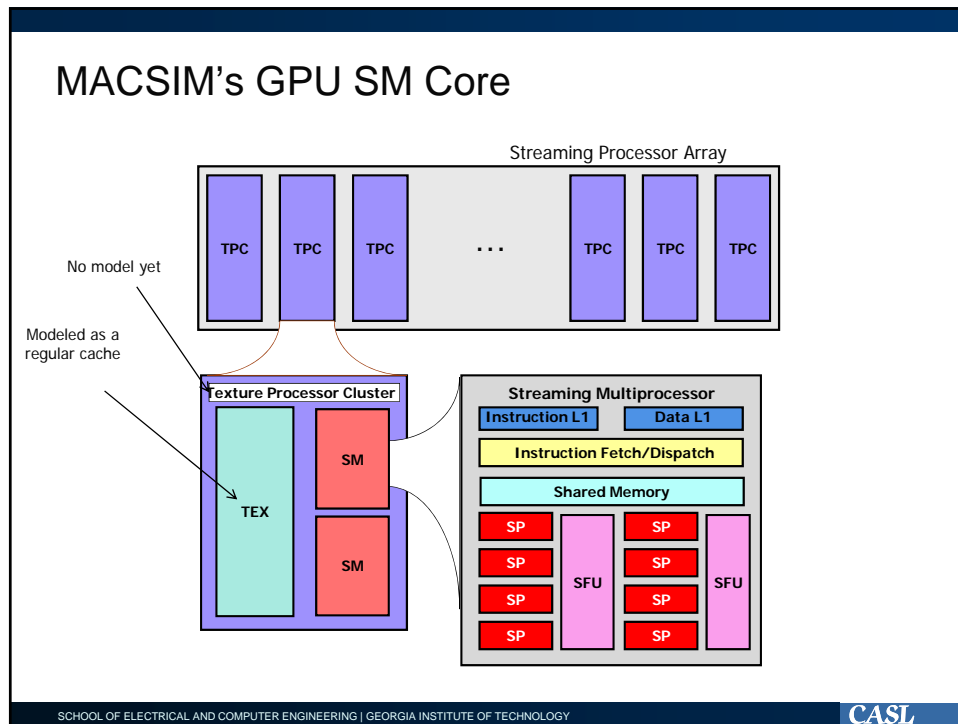
# Ocelot: Dynamic Execution Infrastructure



# Simulating the Tesla Architecture – MACSim (H. Kim)



© David Kirk/NVIDIA and Wen-mei W. Hwu, 2007 ECE 498AL, UIUC



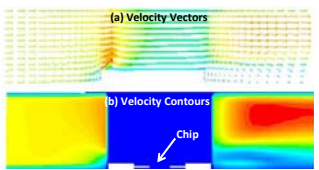
## Coordinated Power and Thermal Modeling

*S. Mukhopadhyay(ECE), S. Kumar(ME), and S. Yalamanchili(ECE)*

- Scope of architecture simulation has changed
- Need on-line (simulation) analysis of performance, power, and thermal behaviors
  - Architectural techniques for energy management driven by on-line analysis of the thermal field
- Need to couple physics of heat flow with detailed architecture simulation
  - System energy optimization requires trade-offs between compute energy and cooling energy
  - Coupling these models is a challenge!

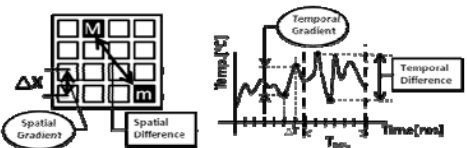
*S. Mukhopadhyay's work was supported in part by Intel and a IBM Faculty Partnership Award*

## Modeling and Simulation Challenges



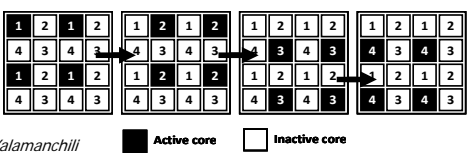
(a) Velocity Vectors  
(b) Velocity Contours

*S. Kumar*



Spatial Gradient, Temporal Gradient, Time[ns]


*S. Mukhopadhyay*



Active core, Inactive core

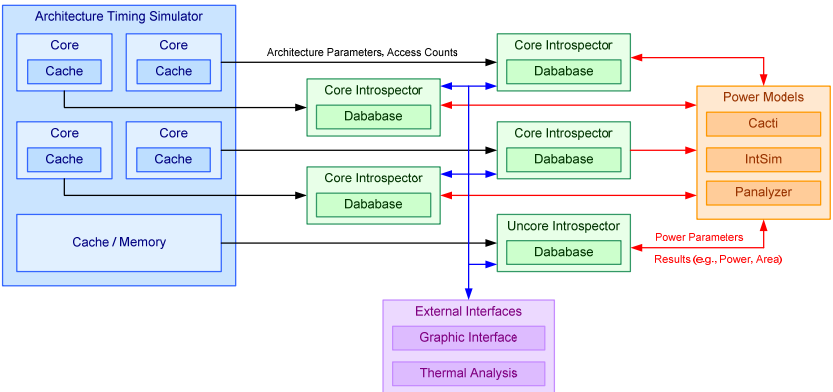
*S. Yalamanchili*

- How do you couple different simulation models?
  - Differing physical time scales
  - Reduced order computational techniques
- How do we model coordinated mechanisms in a single environment?
  - Time management
- What interfaces do we provide?
  - Manifold/SNL API extensions?




## Introspector API

- An Introspector is a pseudo component that monitors a set of model components
- An Introspector provides a uniform interface to physical properties of of a monitored component

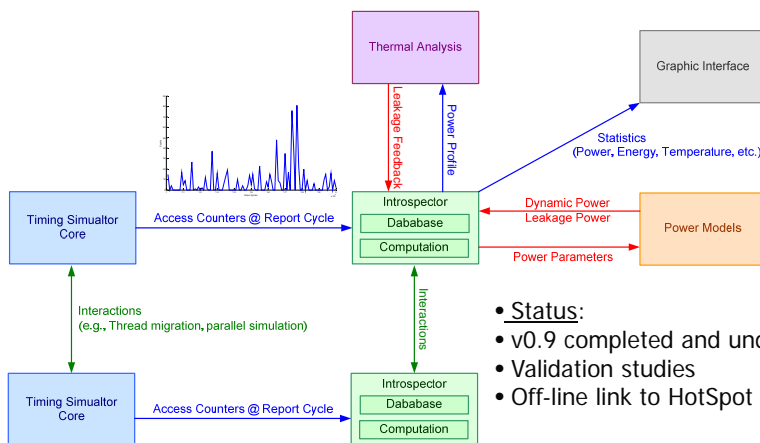


Architecture Timing Simulator, Core, Cache, Core Introspector Database, Power Models (Cacti, IntSim, Panalyzer), External Interfaces (Graphic Interface, Thermal Analysis)



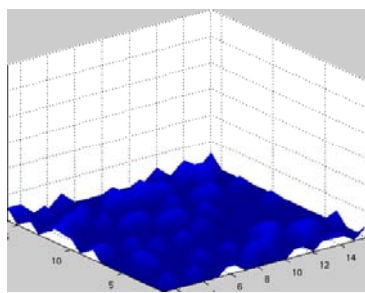
## Introspector Functionality: Example

- Timing simulator reports the access counts of the components to the Introspector
  - User selected models
- The Introspector records energy values based on selected models



- **Status:**
- v0.9 completed and under test
- Validation studies
- Off-line link to HotSpot

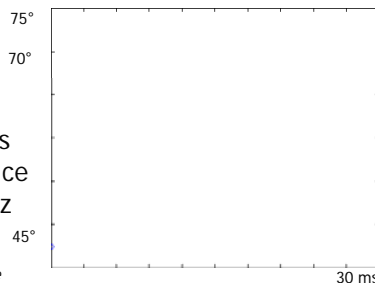
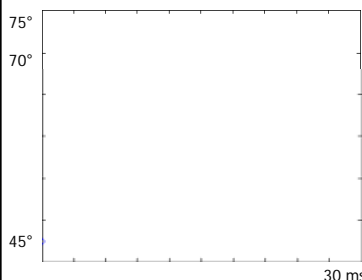
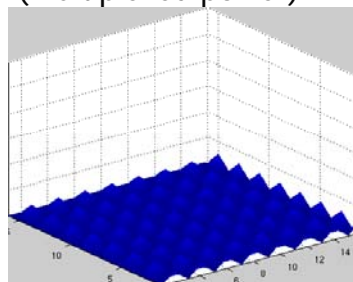
### Unmanaged Thermal Behavior



Spatial gradient:  
10.5°@0.75mm  
Temporal gradient:  
2.5°@100Kcycles

Spatial gradient:  
2.5°@0.75mm  
Temporal gradient:  
1.99°@100Kcycles

### Managed Thermal Behavior (multiplexed power)



- 64 on-tiles
- 256 total tiles
- 100K time slice interval@3GHz

Courtesy: Nikil Sathe



## Some Research Problems

- Integrating interconnection and memory subsystems
  - #memory controllers, address space mappings, routing, topology
- Energy management (with S. Mukhopadhyay)
  - Application signatures → runtime management
  - Integration of non-volatile memory technologies
  - Integrated package-level power and thermal management
- Algorithm-architecture analysis
  - Data mining and (fast) model construction

## Concluding Remarks

- Integration with Sandia SST in Q4/2010
  - X86 processor models
  - Execute code generated by stock compilers and boot (version) Linux
- Keep simulation capacity scaling with Moore's Law
- Coordinated with the Manifold Project (NSF)
- A focus on validation infrastructure in 2011
  - Leverage the GreenIT infrastructure

*Thank You*