



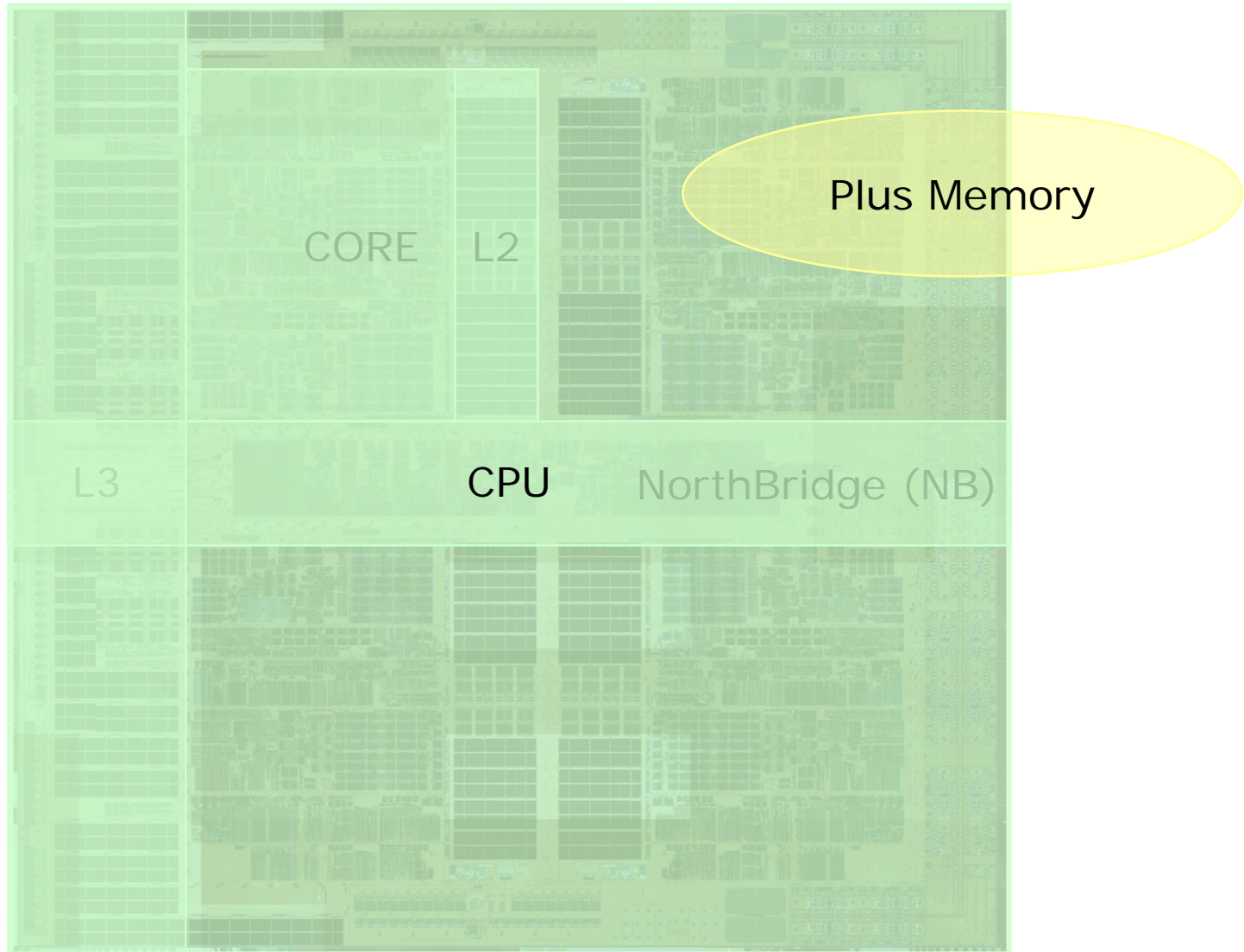
# Performance Modeling and Analysis at AMD: A Guided Tour

Leslie Barnes, AMD Fellow  
ISPASS 2007  
April 27, 2007

# Outline

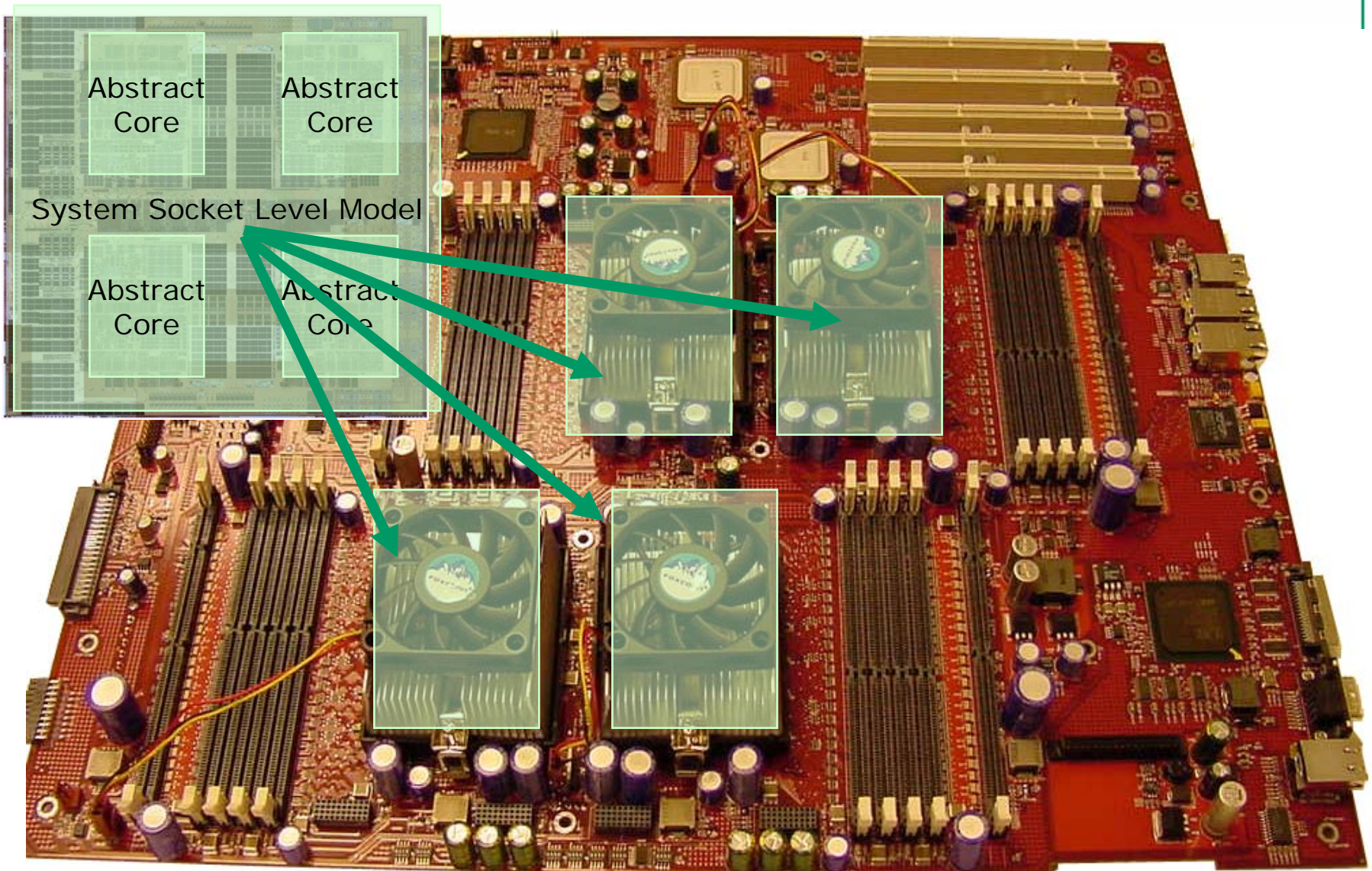
- Nomenclature
- Performance Modeling for Products: Quick overview
- Performance Analysis Tools
- Workloads and Workload Analysis
- CPU Modeling
- System Modeling
- Power Modeling – brief outline
- Sample Recent Modeling Applications
- Future Challenges
- Acknowledgments
- Conclusion

# Nomenclature: CPU model





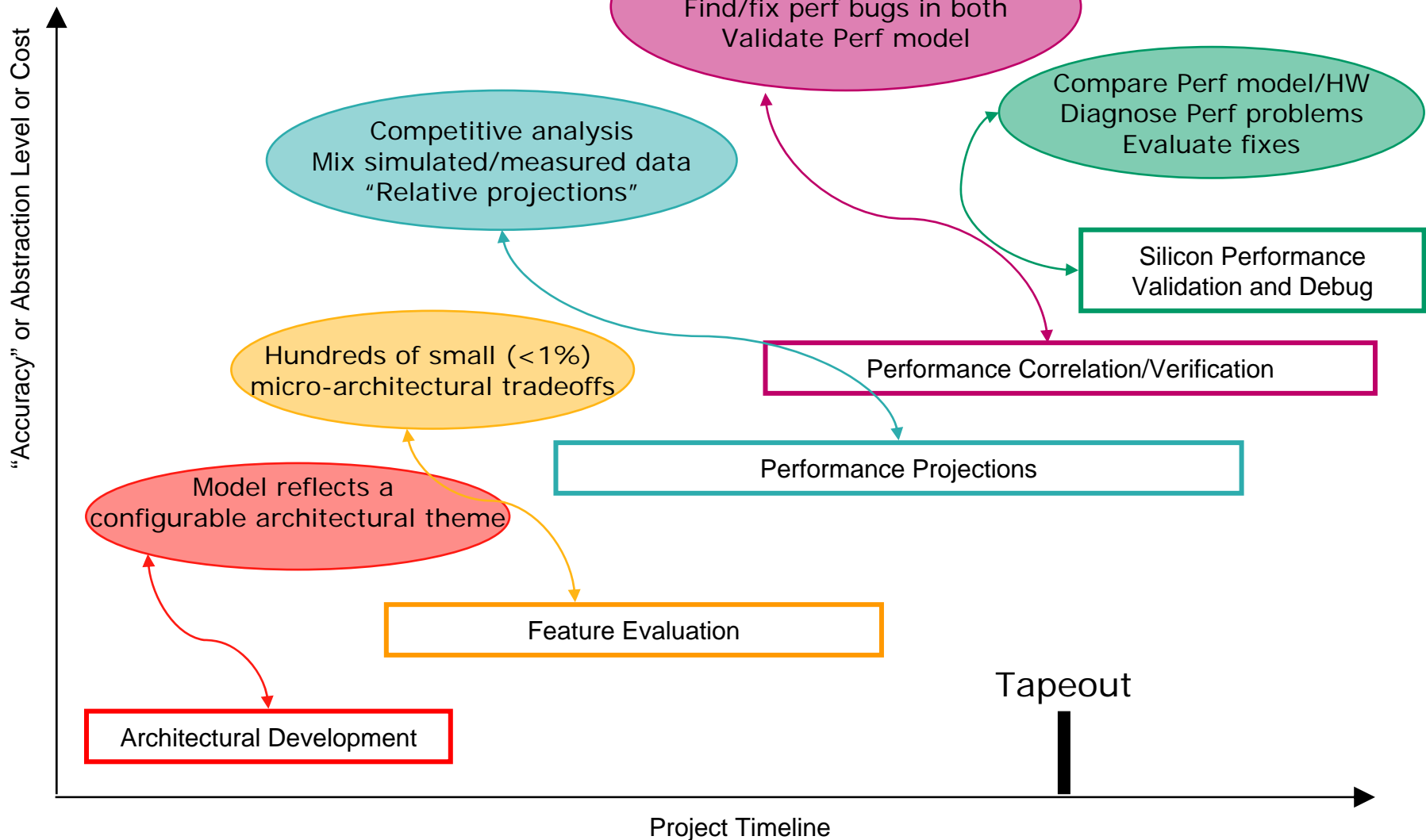
# Nomenclature: System model



# Performance Modeling for Products: Quick Overview

# Performance Modeling for Products

My life for the past few years



# Performance Modeling for Products: What's "new"?

## **Power modeling and power correlation**

- Power and performance are intimately linked
- New role for the performance model
- Huge, cross-disciplinary effort to do this well

## **Virtualization**

- New workloads
- Challenges our performance modeling and tracing infrastructure

## **Graphics performance analysis**

- Discrete, UMA and Fusion!

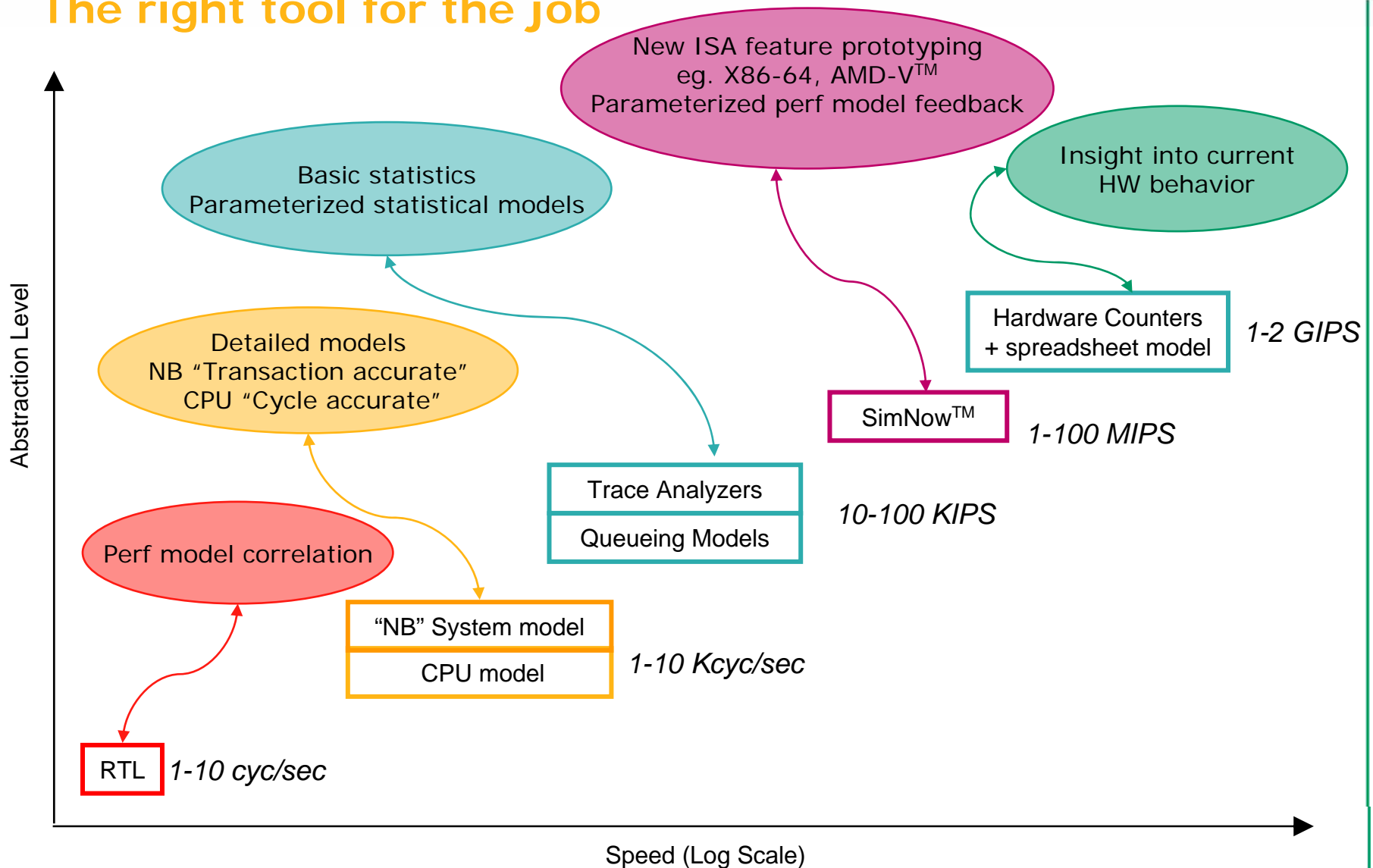
## **More MP, all the time!**

- Cycle-accurate MP simulation is uniquely challenging
- More and more CPUs on a single chip

# Performance Analysis Tool Chain

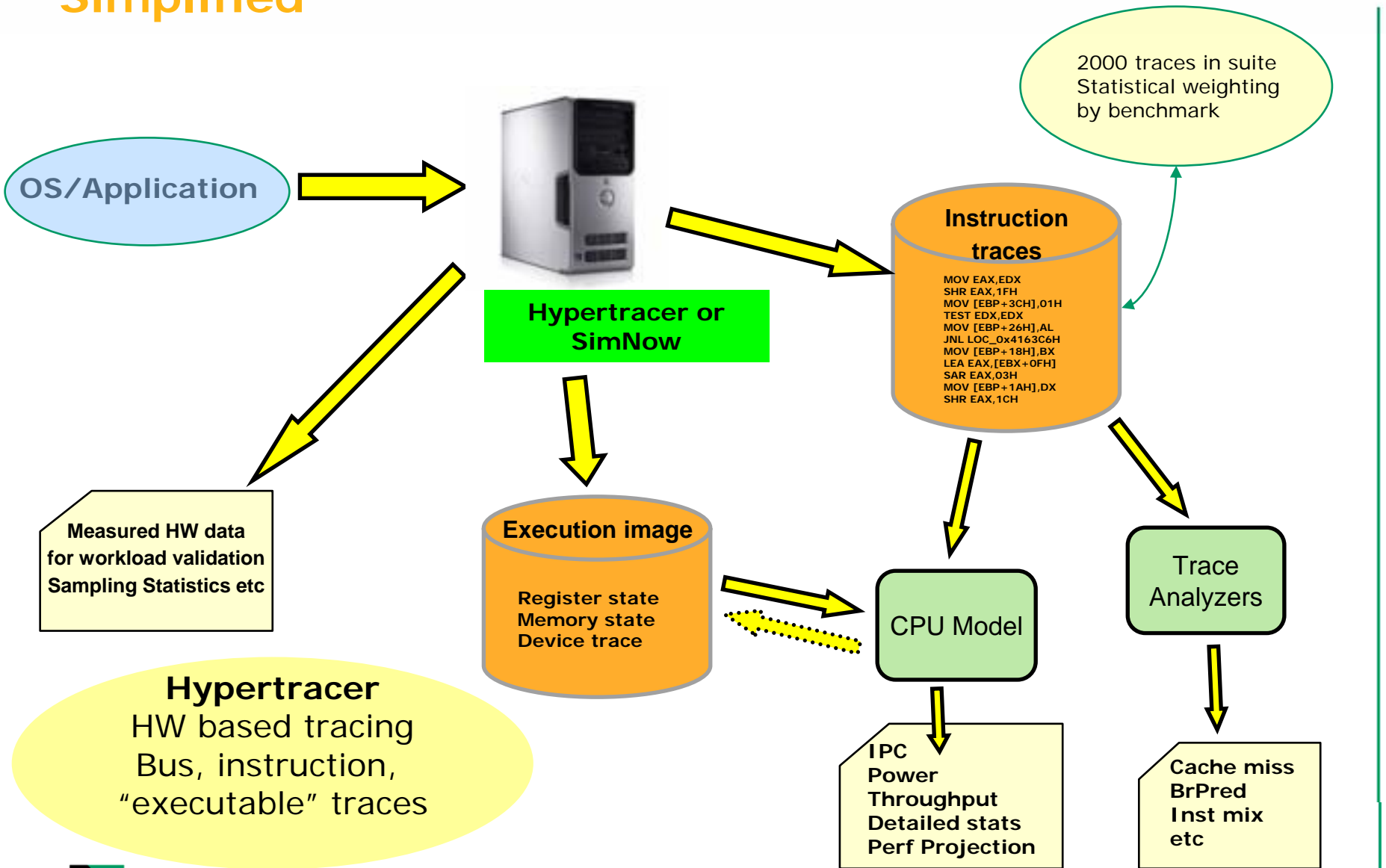


# Performance Analysis Tools: The right tool for the job

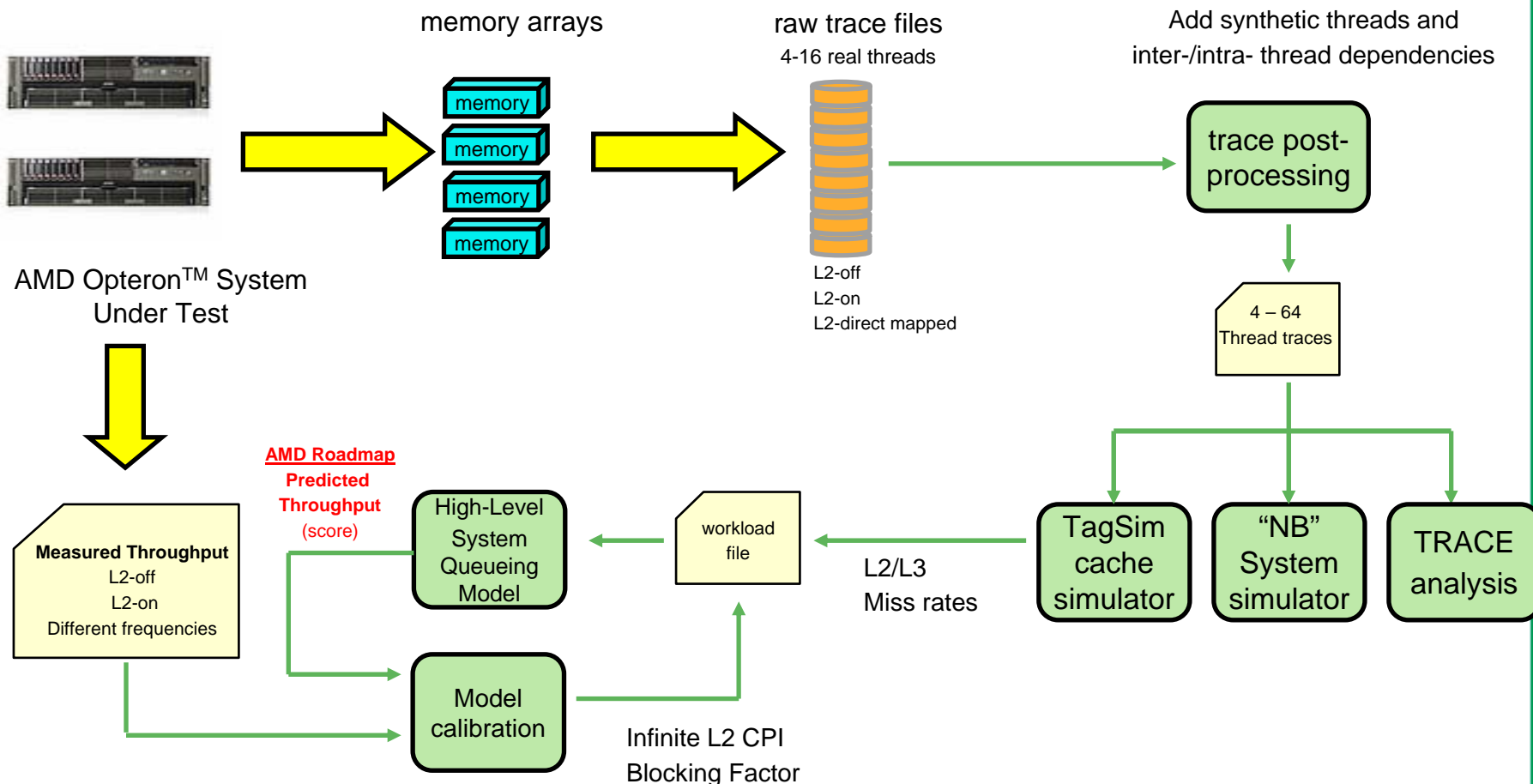


# CPU Performance Modeling Tool Chain

## Simplified



# System Performance Modeling Tool Chain



# SimNow™: Perf/Arch applications

- Fast and configurable x86 and AMD64 instruction-level platform simulator
- Evaluate ISA extensions
  - X86-64, AMD-V™
- Produce instruction traces
- Produce execution-driven workload inputs
- On-the-fly trace analysis
- Golden model for exec-driven CPU perf models
- Playback executable-traces from real-hw
  - Hookup to perf model also
- Network support
- Graphics devices
- Many other uses besides Perf/Arch
  - BIOS, Driver, OS development
  - Compiler development (ISA extensions)

# SimNow™ Screenshot

The screenshot displays the SimNow software interface, which is used for hardware simulation and debugging. The interface is divided into several main sections:

- Debugger (Top Left):** Shows assembly code for a CPU. The register values are:
 

```

            EAX=752D3D40 EBX=00DFF978 ECX=752D0008 EDI=00000070
            ESI=00000000 EDI=00DFFA58 ESP=00DFF8F0 EBP=00DFF908
            CS=001B DS=0023 ES=0023 FS=0038 GS=0000 SS=0023 EFLAGS=odItszapc
            
```

 Below the registers, assembly instructions are listed with their addresses and hex values.
- Simulation Display (Middle):** Contains several performance graphs:
  - CPU: Translation Graph:** Shows a series of blue bars representing instruction translations.
  - CPU: Real MIPS Graph:** Shows a fluctuating line graph of real MIPS.
  - CPU: Invalidation Rate Graph:** Shows a red line graph representing the invalidation rate.
  - CPU: Exception Rate Graph:** Shows a red line graph representing the exception rate.
  - CPU: PIO Rate Graph:** Shows a purple line graph representing the PIO rate.
  - CPU: MMIO Rate Graph:** Shows a blue line graph representing the MMIO rate.
- Simulation Stats (Middle Right):** A table providing performance metrics:
 

Sim Stats	IDE Primary byte counts	IDE Secondary byte counts	Diagnostic Ports	Floppy byte counts
22.60 Host Seconds	293,548,620 master read	8,243 master read	00 00 00 FF 83 - 80	754 read
13,994.56 Sim Seconds	38,247,727 master written	81 master written	00 00 00 00 87 - 84	266 written
2.367 Avg MIPS	8,238 slave read	8,238 slave read	00 00 00 00 e3 - e0	
23 MIPS	71 slave written	71 slave written		
	PIO mode	PIO mode		
- Hardware Diagram (Bottom Left):** A block diagram showing the system architecture. Components include:
  - AMD Debugger Example
  - AMD Soft Graphics Device
  - AweSim Processor
  - Dimm Device
  - Golem Device
  - IGD4 Rev. B0 Northbridge
  - Lokar Device
  - Memory Device
  - PCI Bus
  - Pogo Device
  - SH/CH NorthBridge
  - SMB Hub Device
  - Thor Lite Southbridge
  - Winbond W83627HF SIO
  - Thor Southbridge
  - PCI Bus #7
  - Dimm #5
  - SH/CH NorthBridge Node #4 Dev #3
  - Lokar Device #2
  - AMD Soft Graphics AGP
  - System BIOS: FFFC0000 - FFFFFFFF
- Windows XP Desktop (Background):** Shows a standard Windows XP desktop with a Test Manager dialog box open. The dialog box is titled "Manual Power State Change Test" and contains instructions for testing the power state change. It includes a "Test in Progress" button and a "Start Tests" button.

# Workloads and Workload Analysis



# Workload Overview: Client

## Varied and rapidly changing landscape

### Digital Media

- Multimedia Content Creation Winstone® 2004 (Ziff Davis Media, Inc.)
- SYSmark® 2004 Internet Content Creation (BAPCO®)
- Panorama Factory, Sony Vegas Studio, Microsoft® Movie Maker, Apple iTunes
- ...

### Computer Gaming

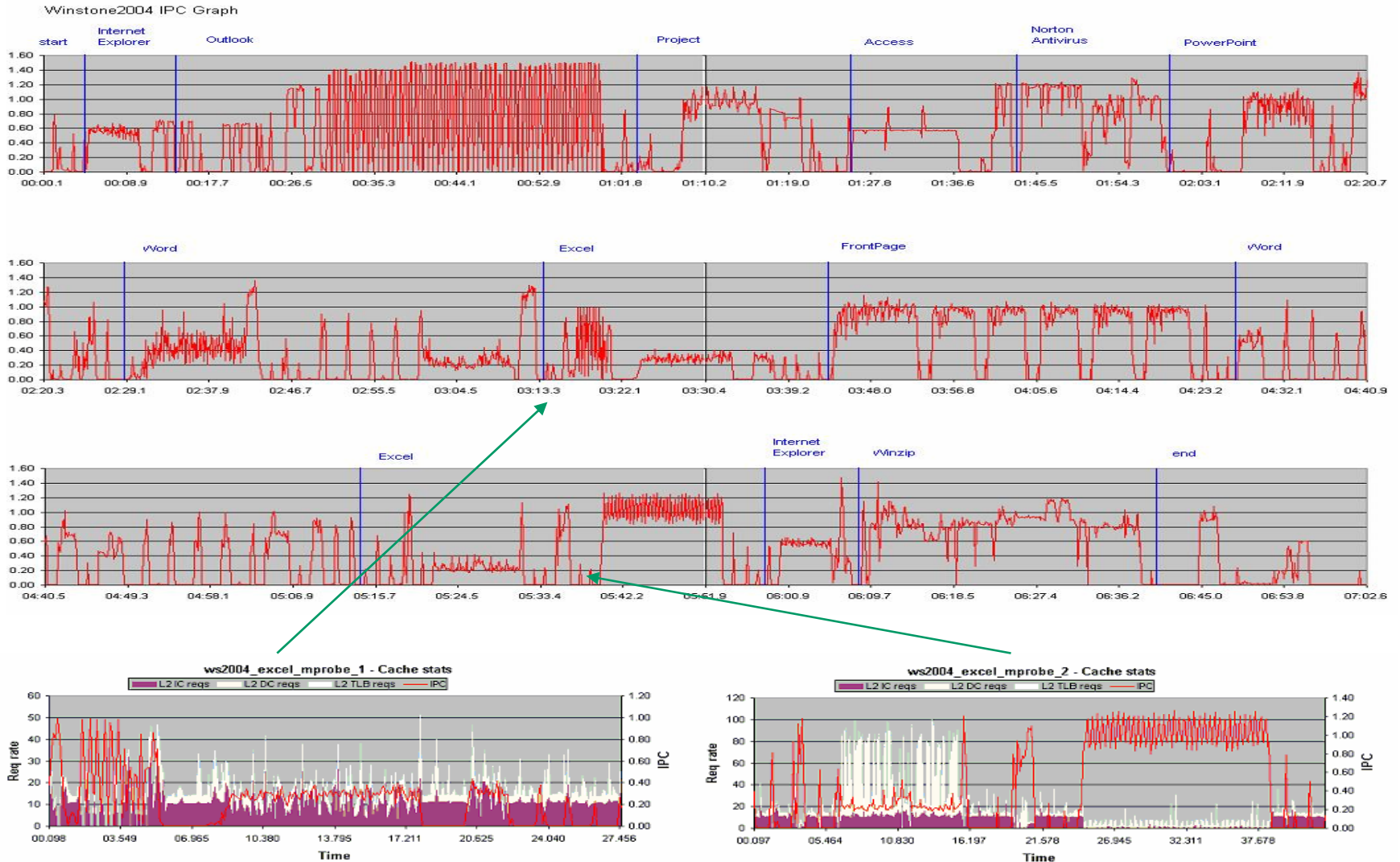
- 3DMark™ 2005/2006 (Futuremark Corporation)
- Doom, Farcry, Halflife 2, ...

### Office Productivity

- Business Winstone® 2004 (Ziff Davis Media, Inc.)
- SYSmark® 2004 Office Productivity (BAPCO®)
- PC Worldbench
- WinRAR
- Remote Collaboration Scenario
  - multi-application benchmark that combines Microsoft® NetMeeting and Windows® Media Encoder
- Travel Ready Scenario
  - multi-application benchmark that combines Microsoft® Publisher 2003 and Nero Recoder
- ...

[http://www.amd.com/us-en/assets/content\\_type/white\\_papers\\_and\\_tech\\_docs/31366.pdf](http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/31366.pdf)

# Client workloads can be complicated!



# Workload Overview: Server/HPC

## SPEC CPU2006

- Many compiler, 32->64-bit and OS variants
- Microsoft®, PGI, Pathscale, Sun, Gnu, Intel compilers
- Windows® XP, Windows Vista™, Linux®, Solaris OS's
- 32-bit and 64-bit

## High-performance computing

- DGEMM (matrix multiply aka HPL), FFT
- LS-Dyna3D, Ansys, ...

## Server

- OLTP with various databases
- SPECweb99, SPECweb99\_SSL, SPECweb2005
- SPEC JBB2000, JBB2005
- Microsoft® Terminal Services
- ...

## Virtualization

- VMmark (from VMware)
- AMD internal benchmarks

## Future Workloads

- Look at current workloads
- Look at industry trends
- Internally develop benchmarks

**Developing accurate workloads  
for simulation remains  
one of our biggest challenges  
for performance projection**

# Workload Sampling

Employ various profiling/analysis techniques to select and validate representative execution strips or traces

- Validate final sampled workloads against measured HW counter data and profile information
- Method employed depends on the workload
  - Some workloads are difficult to trace (eg. Sysmark04)
  - Some workloads are difficult to run on SimNow™ (eg. large-scale multi-tier server)
- Small number of large samples
  - Good for server traces from real HW systems
- Large number of small samples
  - Automated via SimNow
  - Good for straightforward benchmarks such as SPECcpu
  - Trace or execution driven
- Phase analysis
  - EIP/PC monitoring
  - Basic block monitoring
  - Loop analysis
- Simpoint type methods also employed
- Fast-warmup mode for key structures

# Server Workloads

Many server workloads require clients and network modeling

- In the lab, clients are 6-25+ additional machines generating transactions to stimulate the server
- Do we need to simulate the clients?
- Do we need to simulate the network?

Server workloads are really big

- Gigabytes of memory, terabytes of disk space
- Have explored scaled-down server vs full-scale server workloads for execution-driven simulation
  - Single-tier setups can be useful
  - Calibrate against HW data from large-scale systems
- MP traces from real HW
- Overall, take a pragmatic approach
  - Use what we have and move forward

# CPU Performance Modeling



# Cycle-accurate CPU performance model

Include detailed CPU core model, NB, memory controller

- Share NB and memory controller with NB System model

Goal is cycle-accurate simulation against RTL

- Also execution correctness to the level it matters for performance analysis

C++ model with higher-level of abstraction than RTL

- 100K lines of uarch specific code
- 400K lines of shared infrastructure and library code
- Modular structure (SimModules) with timing-aware interfaces (ComPipes)
- Highly parameterized at both the macro and micro level
  - Many, many configuration switches for structures, queues, algorithms, policies
- Don't model everything in the simulator
  - Exceptions, power states, many rare conditions, etc.

Workhorse simulator for core microarchitecture

- Core architecture tradeoffs
- Correlation with full-chip RTL
- Small scale MP simulations

# Trace or Execution-driven?

## Common features

- OS and application code always included
- Instruction stream and memory accesses recorded or generated

## Trace-driven simulation

- Simulates faster
- Sometimes easier to model
- We have thousands of traces from real HW systems

## Execution-driven

- Execute all instructions in simulator
- More accurate simulation model – control and data speculation
- MP interactions can be accurately represented
- Required for accurate power modeling
- Large workloads difficult

## Support both in the same model

# Cycle-accurate CPU model: Investment

- These models represent our biggest investment in modeling from a resource perspective
- People resources
  - Many man-years invested in infrastructure
    - Amortized over projects
  - Many man-years invested in detailed core modeling
    - Specialized to a particular core
    - Modeling and RTL teams work hand-in-hand on uarch
- Simulation resources
  - 1000 high-end AMD Opteron™ CPU's typical, 4x or more for peak
  - 90% or higher utilization on an ongoing basis, month-in, month-out

# Detailed Small-scale MP: Simulation Strategies

## Still a challenging task

Very detailed, cycle-accurate simulation

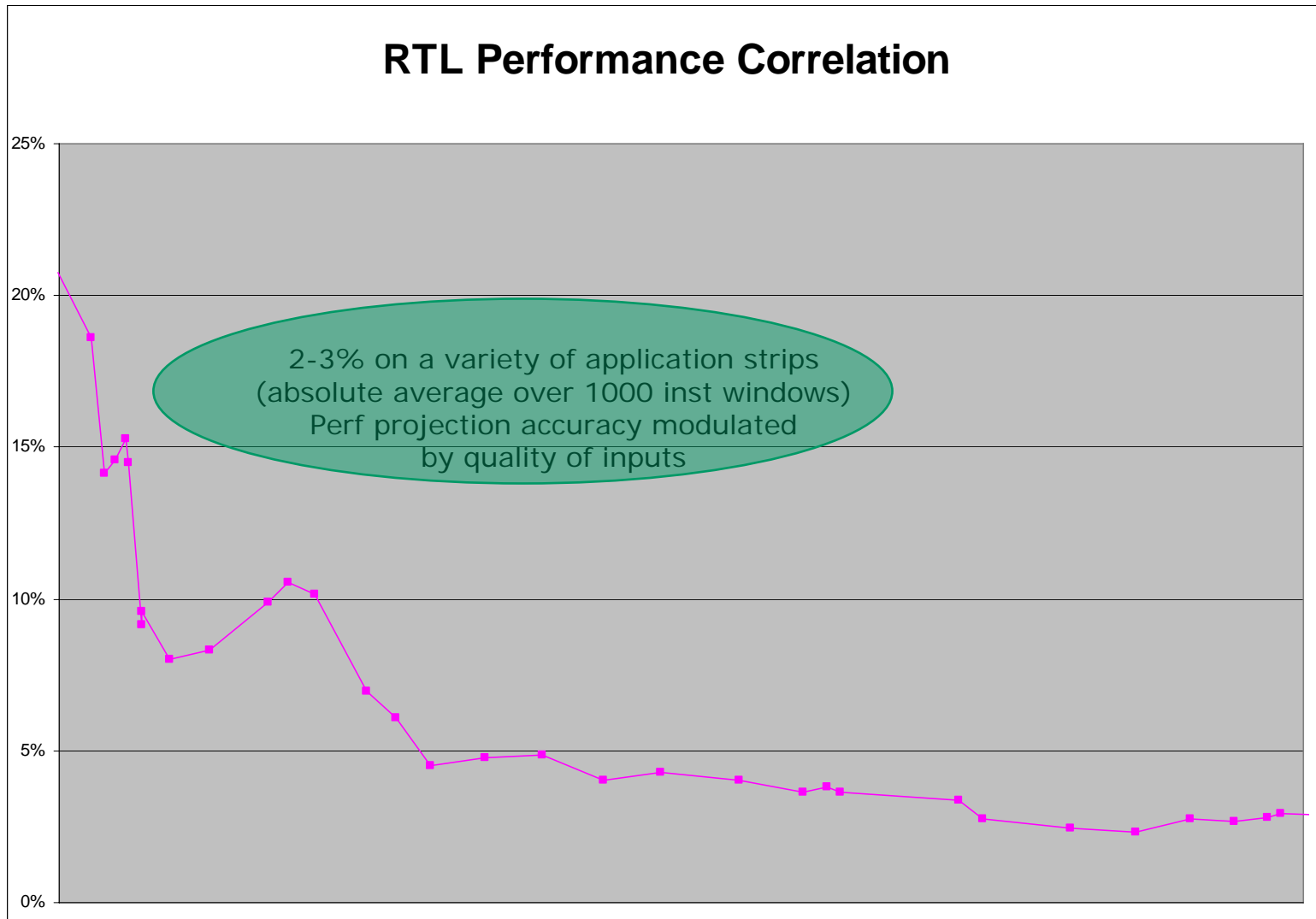
- Simulates the cores, plus the NB/L3/DRAM in detail
- Requires a lot of functional correctness in the model
- Used to examine locking, thread interactions, cache sharing, coherency policies etc

Determinism solutions

- “Trace-driven with Memory Disambiguation”
  - Force multiple threads into different address spaces
  - If they never interact, simulation become deterministic
  - Appropriate for multi-programmed workloads (eg. SpecRate)
- Fixed-transaction simulation
  - Change metric from IPC to a high-level metric such as transactions completed
  - Have to understand and instrument benchmark to measure this metric
  - Run long enough to wash out noise from different transactions completing
- MP-XTR (“Deterministic MP”)
  - Record trace of coherence interactions (executable-trace)
  - Force all simulations to follow the same coherence trace
  - Stall if necessary to force ordering (and measure stall time)
  - Appropriate for evaluating features that don’t interfere with the coherence

# How cycle-accurate?

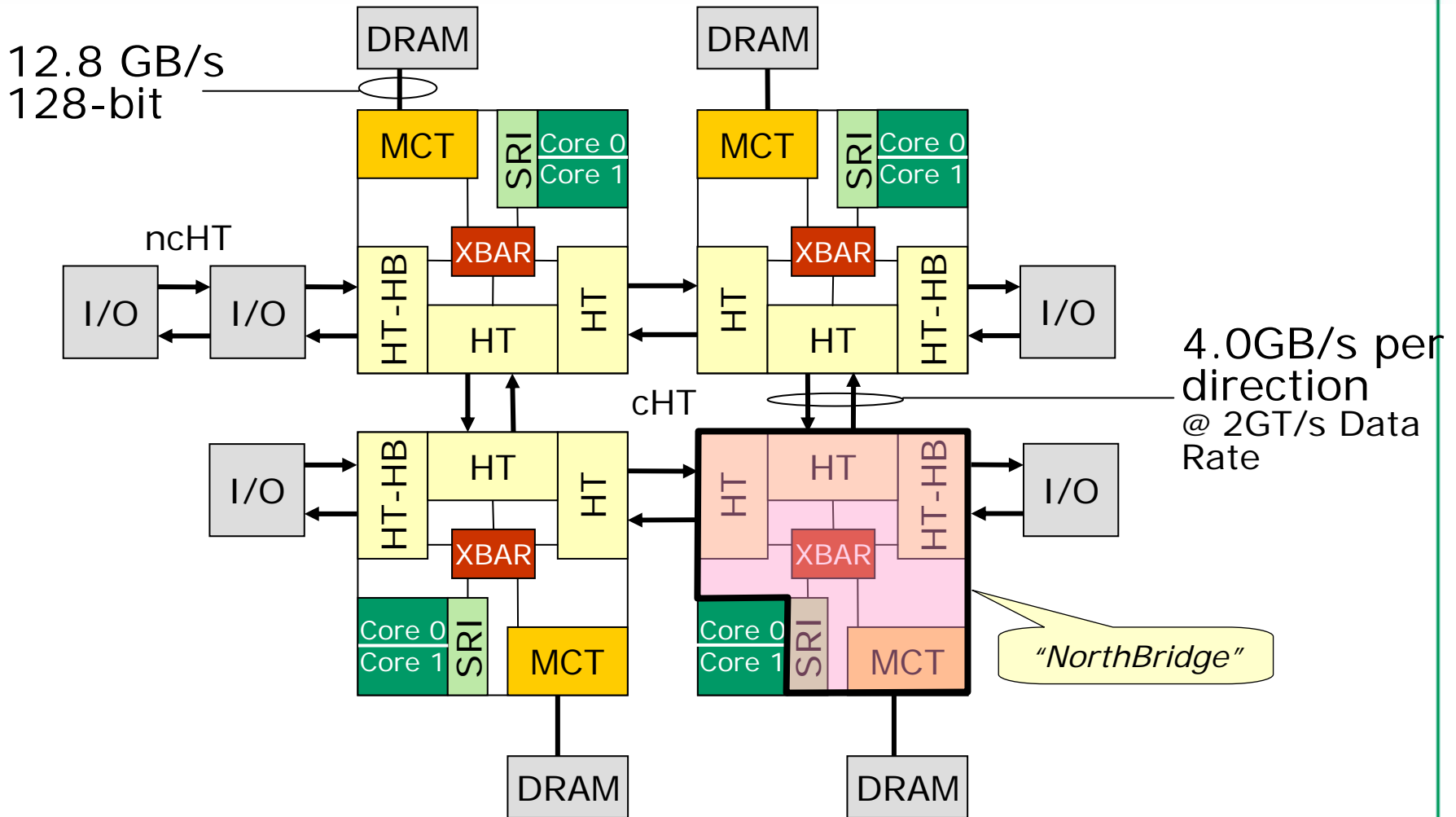
## Correlation against RTL



# System Modeling



# AMD Opteron™ System Overview



Model resource occupancy and latency for Hypertransport, L3 cache, System Request Interface, Memory Controller, etc., message traffic and coherence protocol

Probabilistic traffic generation and miss rates

Abstract CPU model

- $f(\text{ICCPI}, \text{BF and miss rates})$
- Workload parameters Infinite Cache CPI (ICCPI) and Blocking Factors (BF) extracted to fit model to K8 HW measurements

Useful for AMD server performance roadmaps

- Throughput (e.g. tpmC)
- HT bandwidth utilization

## “NB” System Model

Trace-driven multiprocessor model

Doesn't include CPU model (abstracted away)

Includes full Northbridge model in detail

Includes DRAM controller model from timing accurate CPU model

Validated against hardware RTL models

Deterministic MP simulation

- ST-LD ordering across threads preserved
- Enables apples-to-apples comparison of different MP architectures

Useful for clustered MP tradeoff studies

- Queue sizes
- Coherency protocols
- New features

Used to drive Northbridge, Memory system and System design decisions

Trace stimulus comes *directly* from AMD Opteron™ hardware

Focused on Server performance

# Bus Traces

Set of trace files for a *multithreaded* application generated from L2-off bus-traces collected on an AMD Opteron™ MP server.

Each record describes an L1 miss event such as a fetch, load, store or victim

Each record contains metadata specifying how this memory reference is ordered with respect to other memory references in its thread (“intra-thread dependency”) and stores in other threads (“inter-thread dependency”).

Model enforces the same ordering in simulation for consistent comparisons and to expose the effect of memory latency on load-use dependencies.

Loads followed by silent stores (E->M) identified

Synthetic threads added to enable studies of large scale CMP. Code left shared and data made disjoint in synthetic threads.

A 64-thread trace has ~1 Billion L1 miss records across all threads.

# Power Modeling

# Power Modeling

## Motivation

Make power tradeoffs *before* design is complete.

- Evaluate design options before implementation
- Determine features for power efficiency before RTL
- Gatesim based power simulation is (way) too late

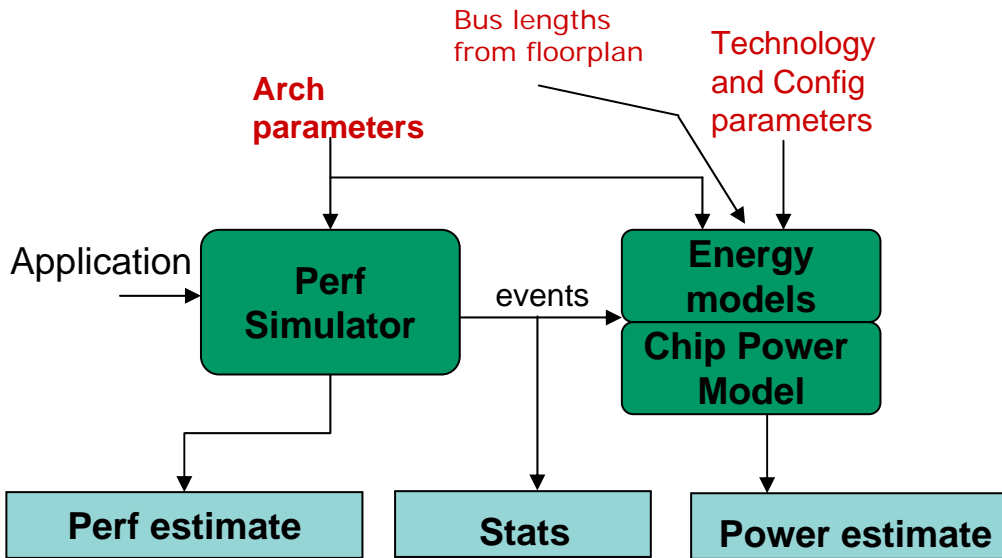
Understand/estimate average power consumption at the benchmark level

- Gatesim power simulation too slow
- Performance Simulator allows many more instructions to be run
- Validate/correlate against actual as design progresses

Measure/track/optimize power throughout the project

Investigate dynamic power management algorithms via performance model

# Power Estimation Overview



Energy models instantiated and configured based on arch params and bus lengths

Chip power model = collection of energy model instantiations

Perf simulator generates events

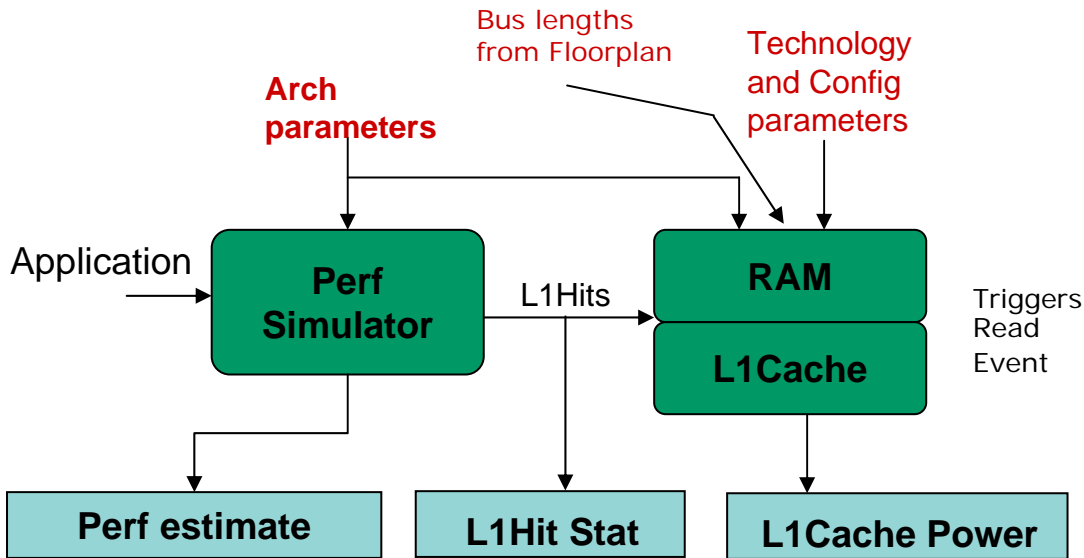
Chip power model returns power for that event

Performance model provides

- microarchitectural energy activity
- time to complete instructions
- Power is equal to energy/time

# Power Estimation Overview

## An Example



Energy model: SRAM

Power model: L1 Cache

Perf event: L1Hits

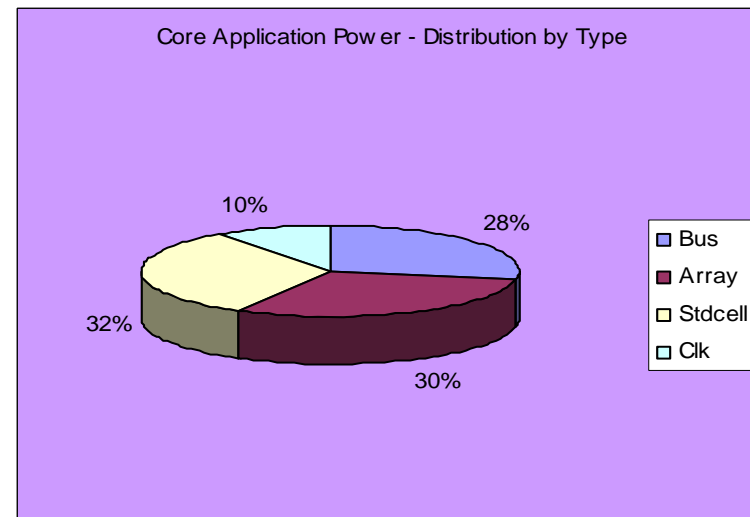
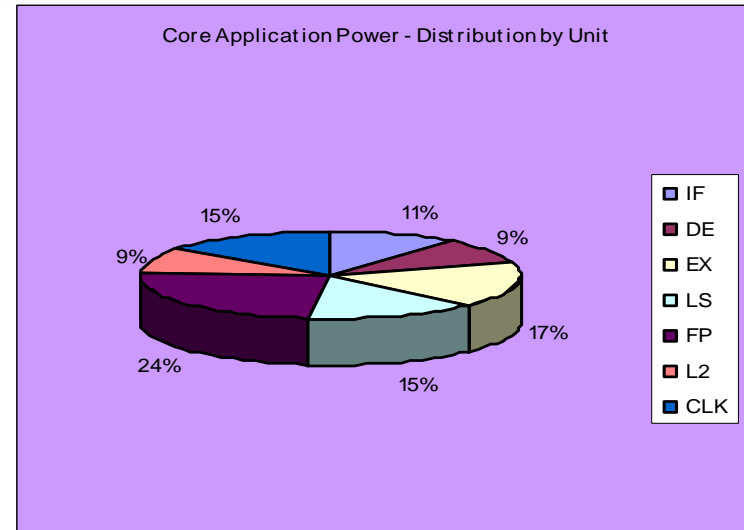
Energy event: Read

Design team provides energy models  
Structures, buses etc in a given technology  
Large amount of work here



# Sample Power Data

- Power analysis can be looked at from different viewpoints
  - Distribution by Unit or Sub-Unit
  - Distribution by Type
  - As a function of time
  - Max power vs Average Power



# Recent Modeling Applications

# Barcelona Core IPC Enhancements: Detailed CPU model application

SSE128 Support

Advanced branch prediction

32B instruction fetch

Sideband Stack Optimizer

Out-of-order load execution

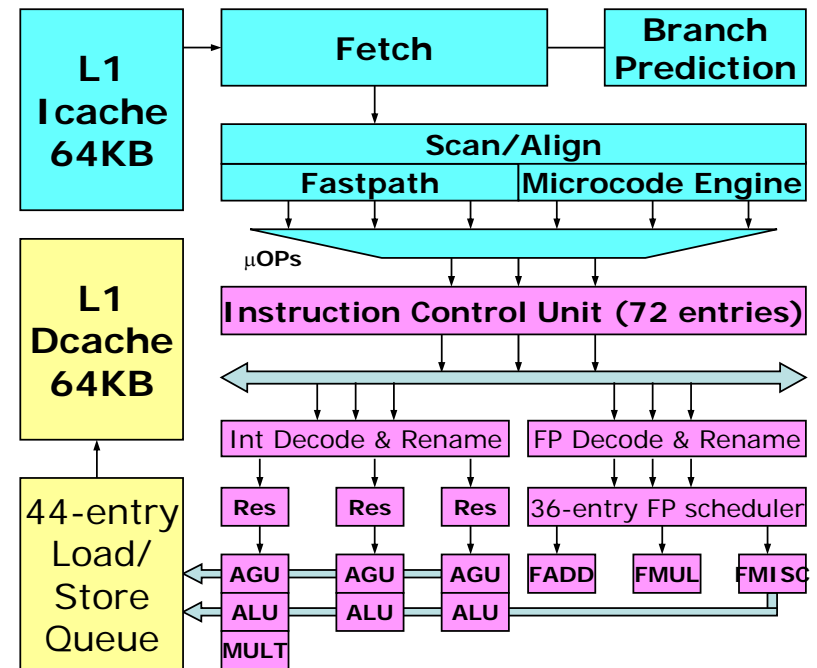
TLB Optimizations

Data-dependent divide latency

Improved Core prefetchers

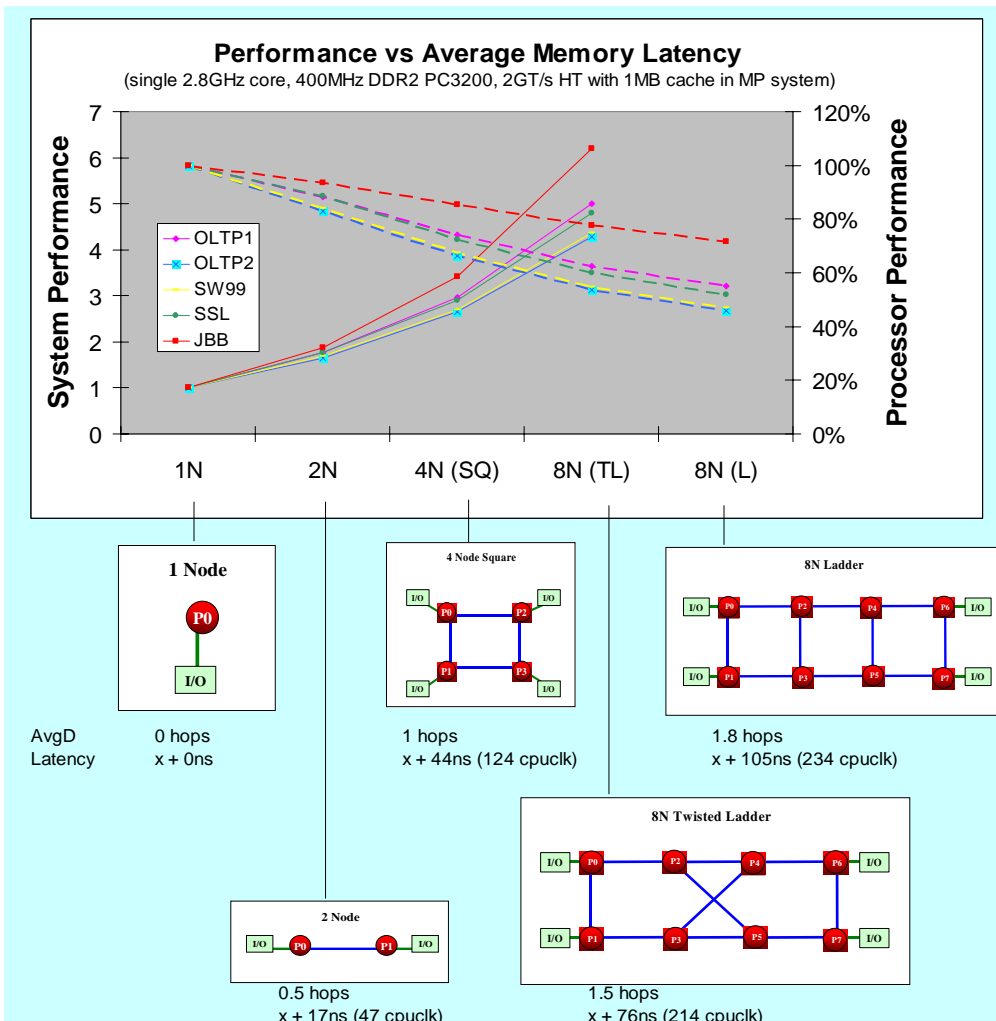
Write bursting

DRAM prefetcher



# Sample System Level Study

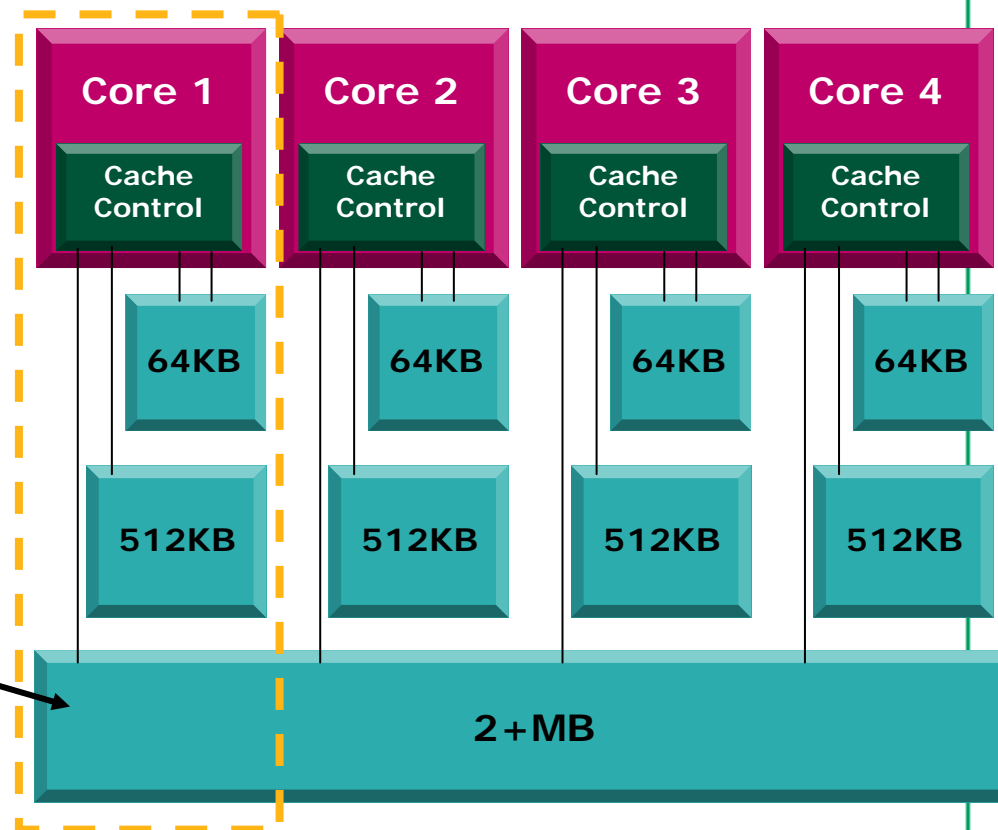
*Memory Latency is the Key to Application Performance!*



# Barcelona L3 Cache Architecture: NB System Model Application

## Shared L3 Cache

- Victim-cache architecture maximizes efficiency of cache hierarchy
- Fills from L3 leave likely shared lines in the L3
- Sharing-aware replacement policy



# Future Directions and Challenges

# Challenges going forward

- CPU + GPU performance modeling
  - Traditionally CPU guys have “abstracted” away (aka ignored) the GPU
  - Traditionally GPU guys have “abstracted” away the CPU
  - Model needs to change going forward
- More MP, all the time!
  - Server, desktop, laptop, palmtop all going MP
- More Virtualization, all the time!
  - A workload/tools challenge
  - What benchmarks and how to run under simulation?
  - Trace or exec-driven?
- Larger systems, more complex, longer workloads
  - More cpu’s, memory, disk, networking, graphics

# Acknowledgments

All this work done by people on the product front line:

- SVDC - Sunnyvale performance team
- ASDC – South Austin performance team
- AMD Performance Labs in Austin
- SimNow team – ASDC & SVDC



## Conclusion

**We've got a lot of work to do!**

**Ask the right questions**

**Apply the right tools**

**Get a reasonable answer ASAP**

**The design can't wait long for perf data**

**Thanks for your attention!**

AMD, the AMD Arrow logo, AMD Opteron, and combinations thereof, AMD Smarter Choice Logo, AMD-V and SimNow are trademarks of Advanced Micro Devices, Inc.

Microsoft, Windows and Windows Vista are registered trademarks of Microsoft Corporation.

BAPCO and SYSmark are registered trademarks of Business Applications Performance Corporation.

3DMark is registered trademark of Futuremark Corporation.

Business Winstone and Content Creation Winstone are registered trademarks of Ziff Davis Media, Inc., in the U.S. and other countries.

Linux is a registered trademark of Linus Torvalds

Other product names and company names used in this publication are for identification purposes only and may be trademarks of their respective companies.