

A background image showing a grid of small glass wells, likely a microarray, with a pipette tip positioned over one of the wells, dispensing a small amount of liquid. The lighting is dramatic, with a blue-to-orange gradient.

Multithreaded Multicores, An Update from Sun

Marc Tremblay, Ph.D.

**Sr. VP & Fellow, Chief Architect - Systems
Sun Microsystems, Inc.**

Classic Performance

$$\begin{aligned} \text{CPU Time} = & \text{Instruction Count} * \\ & \text{Cycles Per Instruction} * \\ & \text{Clock Cycle Time} \end{aligned}$$

- With some constraints on overall chip area and power

Performance in 2006

$$\begin{aligned}
 \text{CPU Time} = & \sum_{\text{core}=0}^N \sum_{\text{thread}=0}^T \text{Instruction Count} * \\
 & \text{Cycles Per Instruction} * \\
 & \text{Clock Cycle Time} \\
 & - \text{Overlap}
 \end{aligned}$$

- Overlap is key – temporal and spatial
- Constructive and destructive components
- With tremendous power and area constraints

Processes and Threads – An Opportunity

- Overlapping
 - > Computation
 - > Memory accesses
- Time sharing
 - > Interconnect
 - > Memory controllers
 - > I/O
 - > Caches
 - > Cores
 - > Functional units

Observations

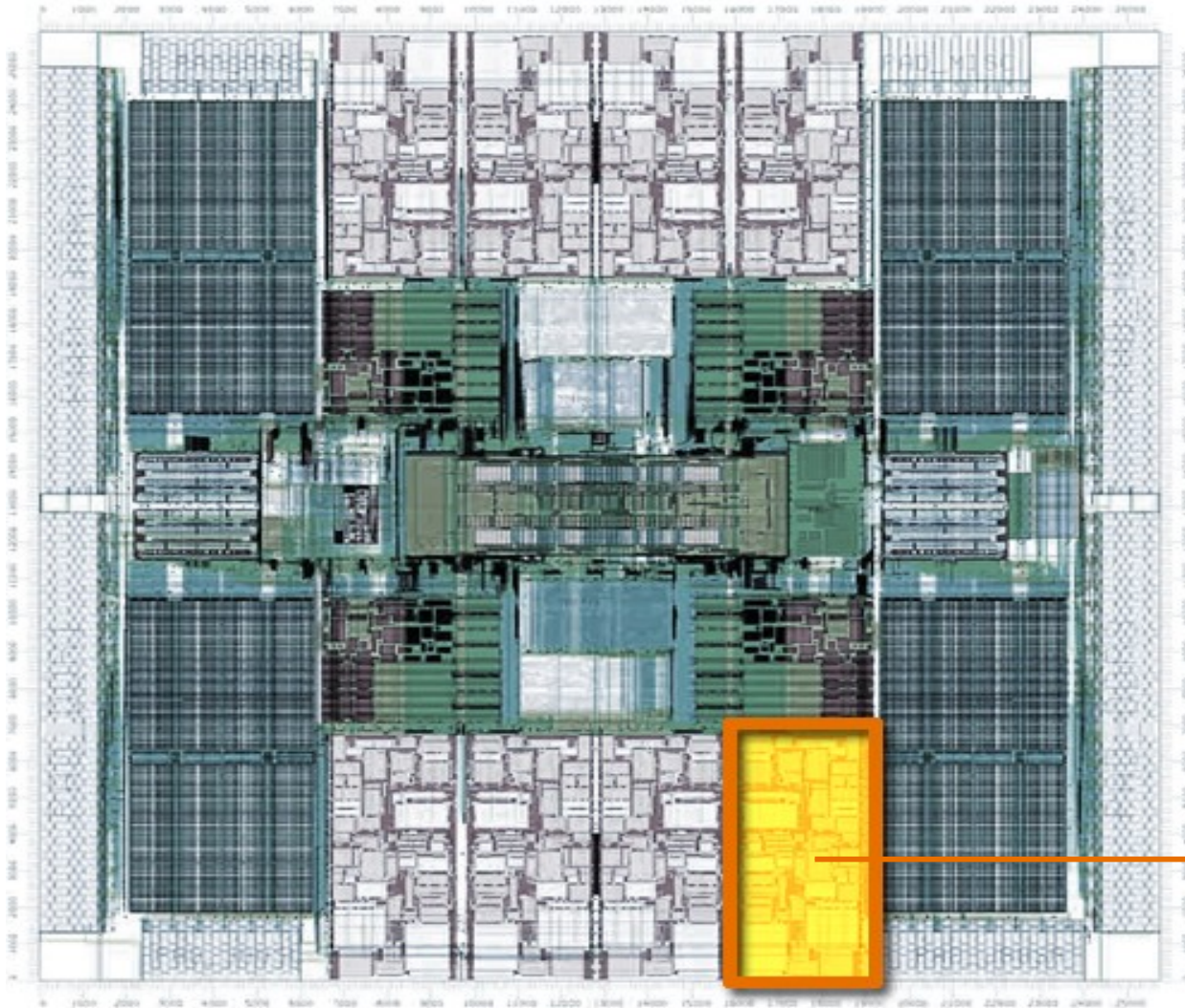
Frequency

- Big lever for cache-resident applications
- Small lever for large commercial applications
 - > High miss rates
- Huge impact on power $P \propto \frac{1}{2} * C * f * V^2$
 - > Equation says proportional:
 - > But designing for high clock rate is much more expensive
- Not super-aggressive but not lazy either
 - > Power efficient circuit family
 - > Small clock domains => low skew overhead
 - > Target traditional RISC cycle time

Area

- Experience so far on three CMTs (at least 8 cores)
 - > Underestimated area significantly
 - > Small increase of 1 mm² on a block replicated 16 times grows the die by at least 16 mm²
- Dedicating area to cores less area for...
 - > On-chip caches
 - > MMUs
- Combined with: many cores, many threads, many states + multi-programming
 - > Higher miss rates

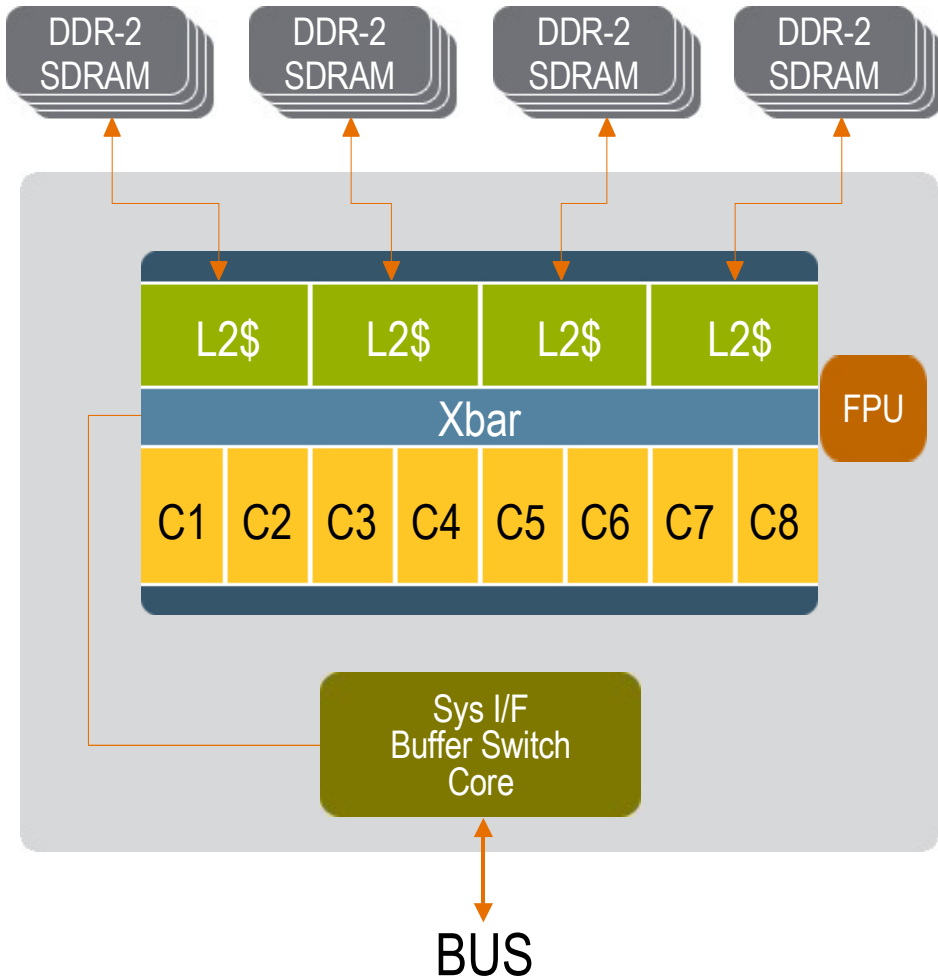
UltraSparc T1 – Die Photo



One
Core

Where Are We Today? UltraSparc T1

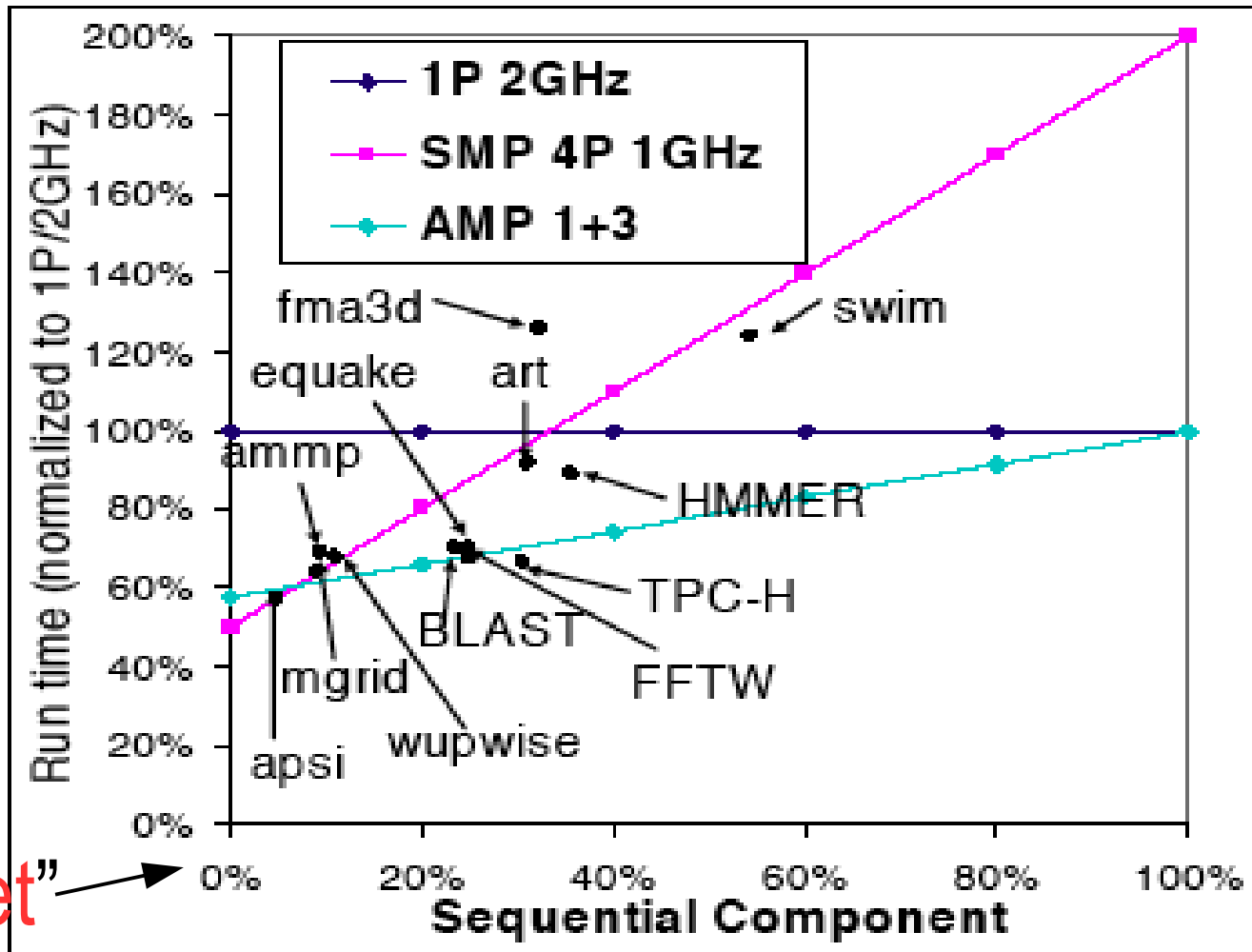
- SPARC V9 implementation
- 8 cores * 4 threads = 32 threads
- All cores connected through a 134.4GB/s crossbar switch
- High-bandwidth 12-way associative 3MB Level-2 cache on chip
- 4 DDR2 channels (23GB/s)
- Power : < 70W !
- ~300M transistors
- 378 sq. mm die



Power of Multithreading

<i>CPI Component</i>	<i>4 Threads Per Core</i>	<i>1 Thread Per Core</i>
Execution	1.000	1.000
Waiting for Pipeline	0.454	0.000
Instruction Cache Miss	1.507	1.176
Data Cache Miss	1.257	0.851
L2 Cache Miss	1.285	0.675
Store Buffer Full	0.315	0.266
Pipeline Latency	1.220	1.238
Miscellaneous	0.173	0.172
Per-thread CPI	7.211	5.378
Per-core CPI	1.802	5.378

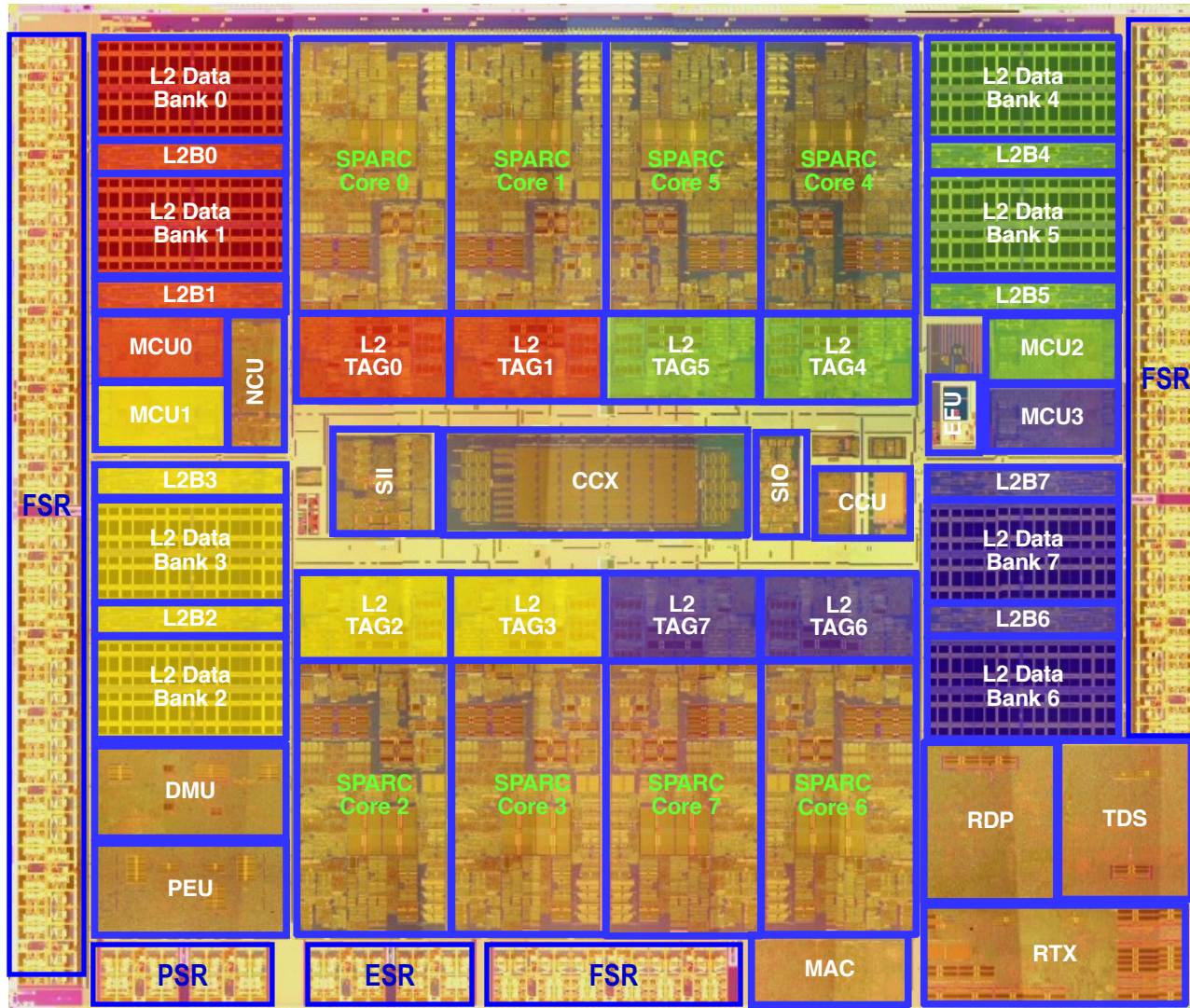
Web 1.0 and Web 2.0 Computing



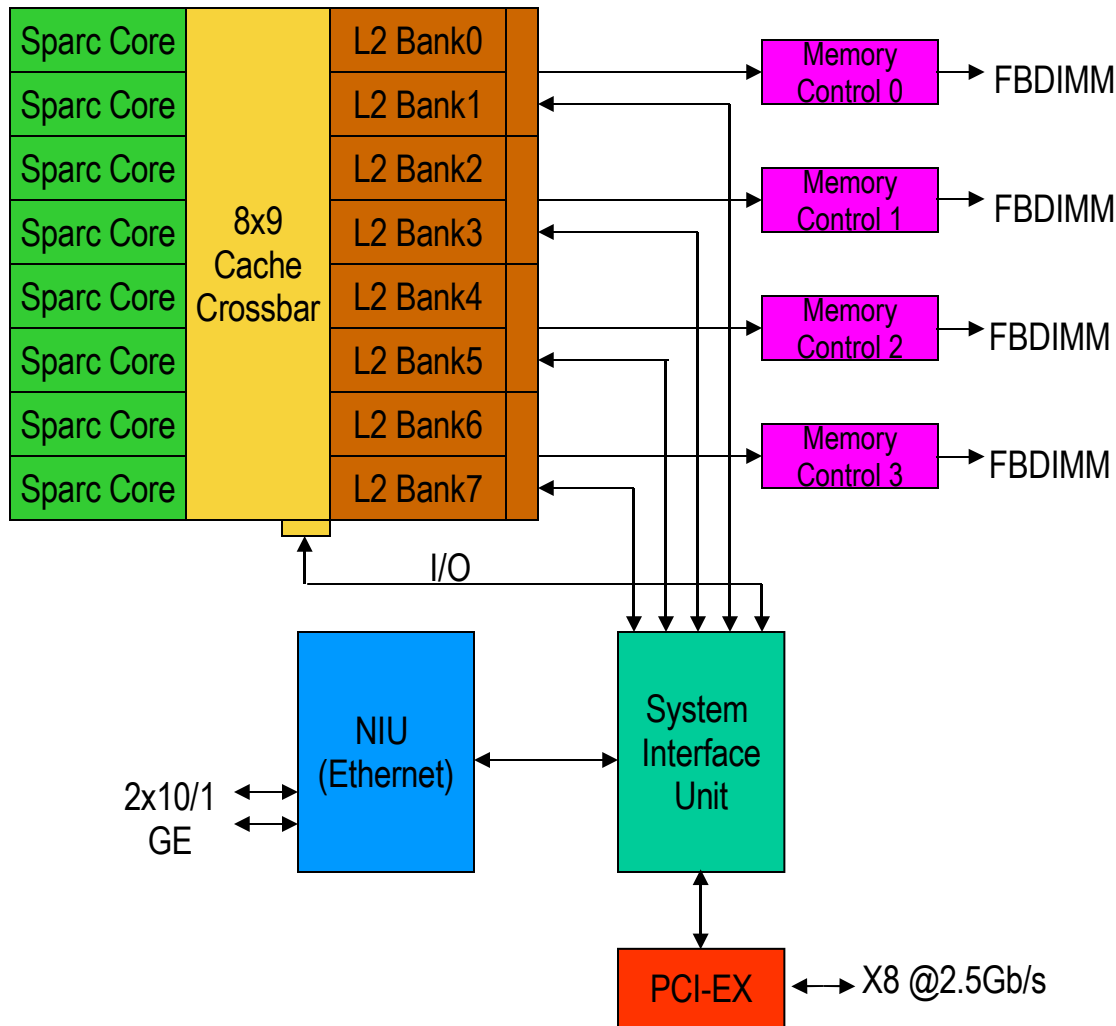
Source: CPI Throttling paper from Intel ISCA 2005

Niagara2 Chip Overview

- 8 Sparc cores, 8 threads each
- Shared 4MB L2, 8-banks, 16-way associative
- Four dual-channel FBDIMM memory controllers
- Two 10/1 Gb Enet ports
- One PCI-Express x8 1.0A port
- 342 mm² die size in 65 nm
- 711 signal I/O, 1831 total

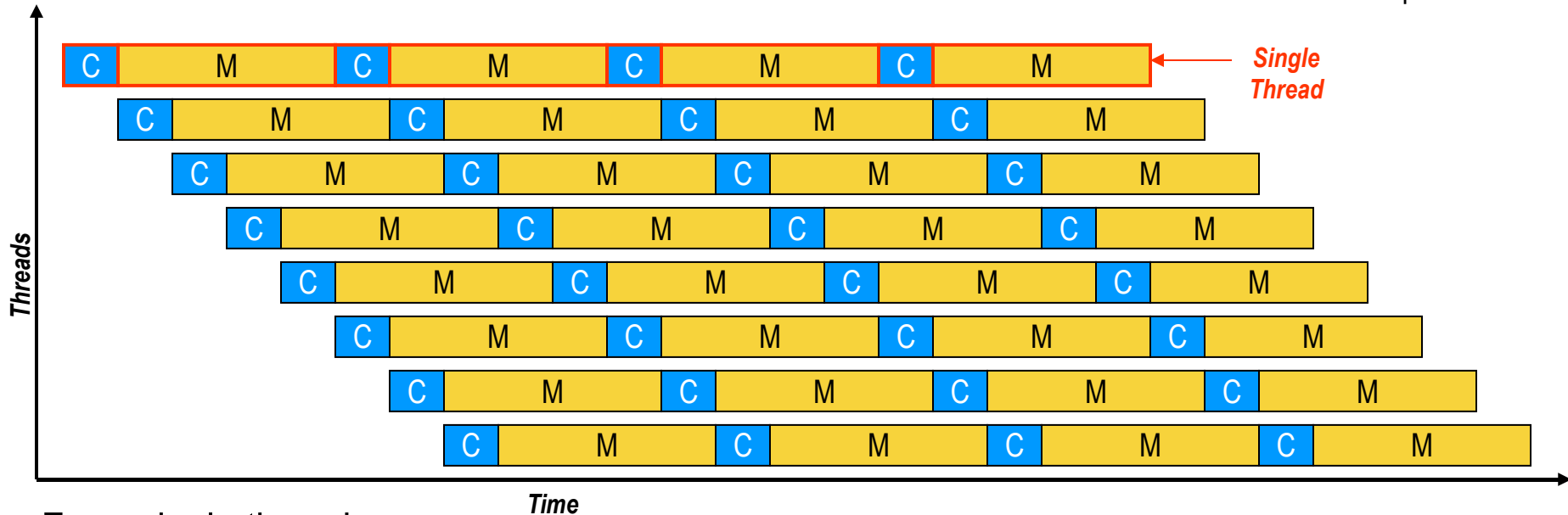


Niagara2 Chip Overview



- Full 8x9 crossbar switch
 - Connects every core to every L2 bank and vice-versa
 - Supports 8 byte writes from a core to a bank
 - Supports 16 byte reads from a bank to core
 - One port for core to read/write IO
- System interface unit connects networking and IO to memory

Throughput Computing

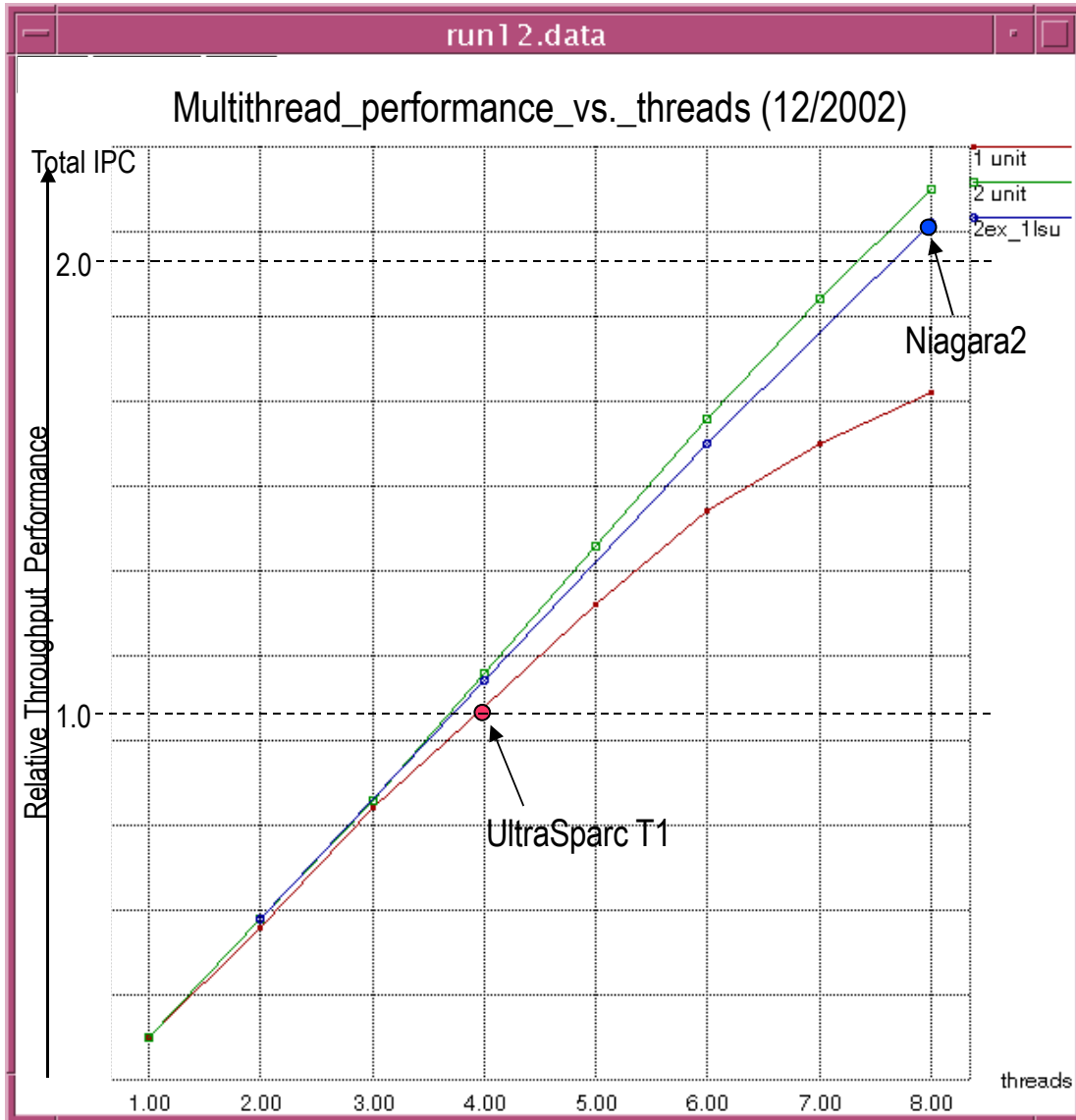


- For a single thread
 - Memory is THE bottleneck to improving performance
 - Commercial server workloads exhibit poor memory locality
 - Only a modest throughput speedup is possible by reducing compute time
 - Conventional single-thread processors optimized for ILP have low utilizations
- With many threads
 - It's possible to find something to execute every cycle
 - Significant throughput speedups are possible
 - Processor utilization is much higher

Engineering Solutions

- Design Problem
 - > Double UltraSparc T1's throughput and throughput/watt
 - > Improve UltraSparc T1's FP single-thread and throughput performance
 - > Minimize required area for these improvements
- Considered doubling number of UltraSparc T1 cores
 - > 16 cores of 4 threads each
 - > Takes too much die area
 - > No area left for improving FP performance

Engineering Solutions



- Probabilistic Modelling
 - > Generate synthetic traces for each thread with an instruction/miss profile that matches TPC-C
 - > Schedule ready threads to run on some number of execution units
 - > End simulation once simulated distributions are close to actual distributions
- Works very well for simple scalar cores running lots of threads on transactional workloads
 - > Within 10 percent of a detailed cycle accurate simulator
 - > Detailed cycle accurate simulator not available at beginning of the project

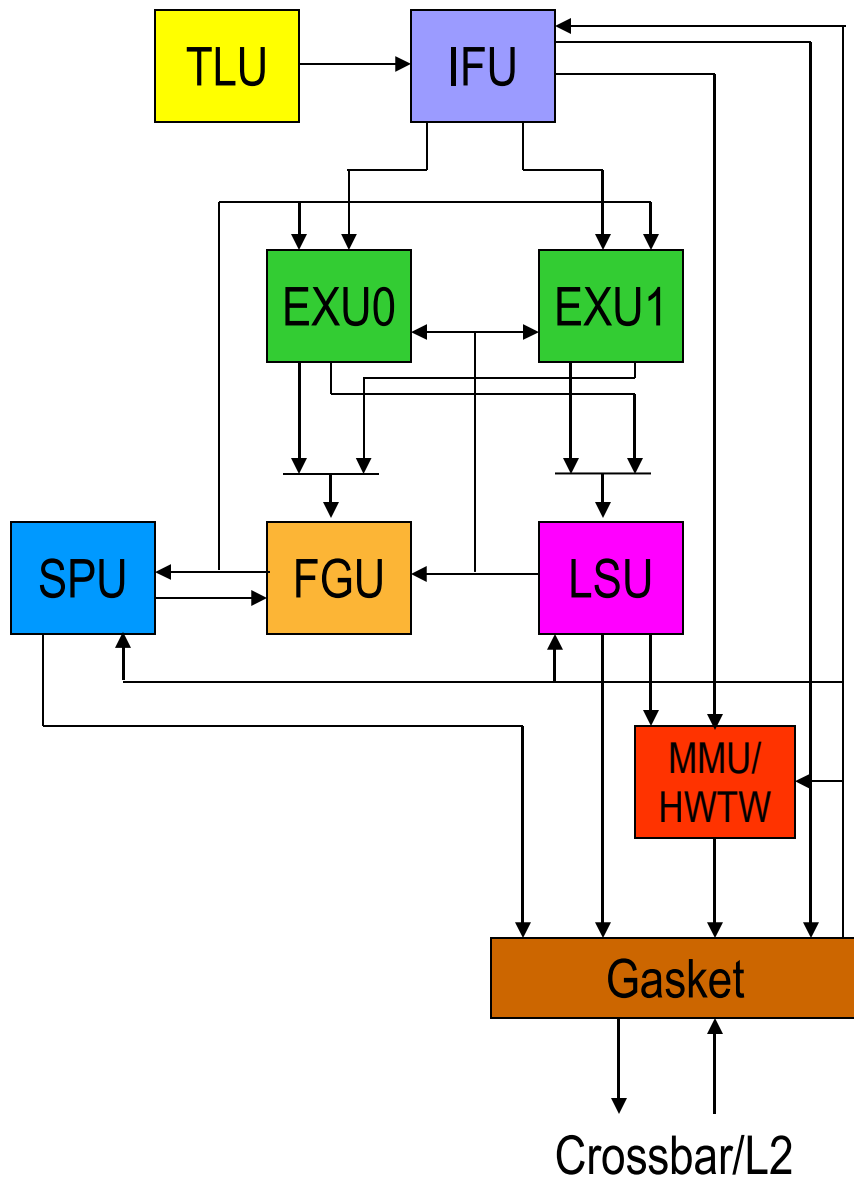
Engineering Solutions

- Decided to increase the number of threads per core and increase execution bandwidth
 - > 8 threads per core x 8 cores = 64 threads total
 - > 2 EXUs per core
 - > More than doubles UltraSparc T1's throughput
 - > Doubling threads is more area efficient than doubling cores
 - > Integrate FGU into core pipeline
 - 6 cycle FP latency
 - Threads running FP are non-blocking
 - > Enhance Niagara2's cryptography
 - Added more ciphers
 - Enhanced existing public key support

Throughput Changes

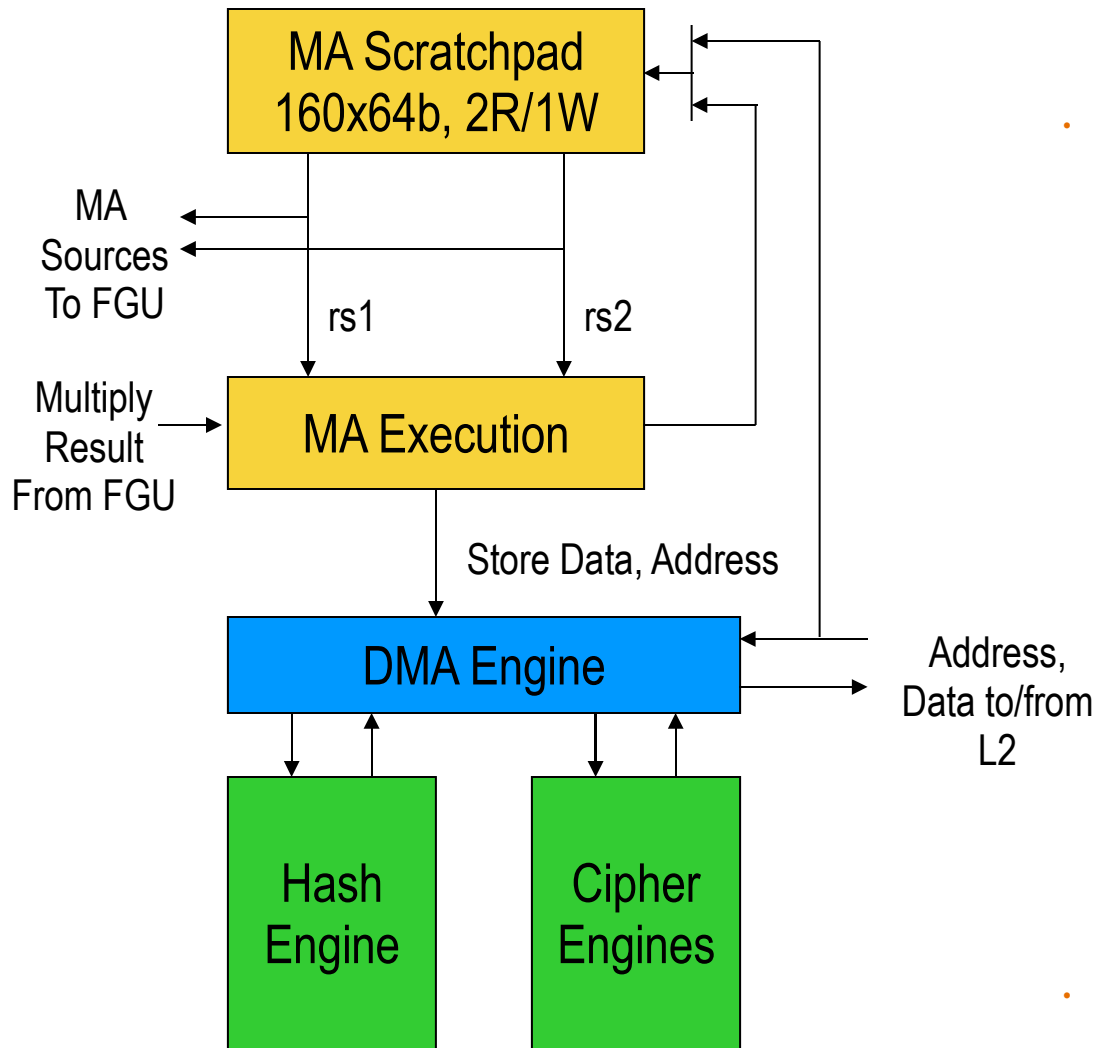
- Niagara2 throughput changes vs. UltraSparc T1
 - > Add instruction buffers after L1 instruction cache for each thread
 - > Add new pipe stage “pick”
 - > Choose 2 threads out of 8 to execute each cycle
 - > Increase execution units from 1 to 2
 - > Increase set associativity of L1 instruction cache to 8
 - > Increase size of fully associative DTLB from 64 to 128 entries
 - > Increase L2 banks from 4 to 8
 - > 15 percent performance loss with only 4 banks and 64 threads
 - > Increase threads from 4 to 8

Sparc Core Block Diagram



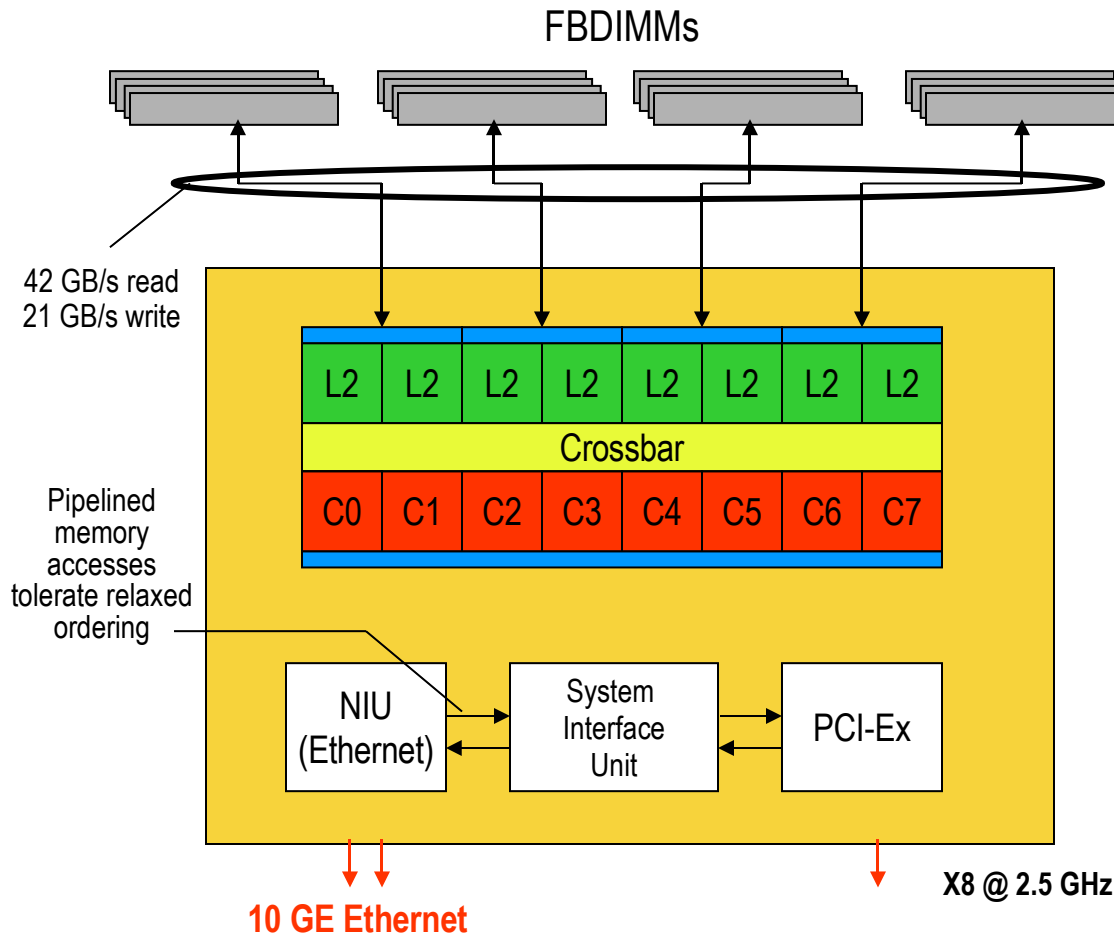
- IFU – Instruction Fetch Unit
 - › 16 KB I\$, 32B lines, 8-way SA
 - › 64-entry fully-associative ITLB
- EXU0/1 – Integer Execution Units
 - › 4 threads share each unit
 - › 8 register windows/thread
 - › 160 IRF entries/thread
- LSU – Load/Store Unit
 - › 8 threads share LSU
 - › 8KB D\$, 16B lines, 4-way SA
 - › 128-entry fully-associative DTLB
- FGU – Floating-Point/Graphics Unit
 - 8 threads share FGU
 - 32 FRF entries/thread
- SPU – Stream Processing Unit
 - › Cryptographic coprocessor
- TLU – Trap Logic Unit
 - › Updates machine state, handles exceptions and interrupts
- MMU – Memory Management Unit
 - › Hardware tablewalk (HWTW)
 - › 8KB, 64KB, 4MB, 256MB pages

Stream Processing Unit



- Cryptographic coprocessor
 - > One per core
 - > Runs in parallel w/core at same frequency
- Two independent sub-units
 - > Modular Arithmetic Unit
 - > RSA, binary and integer polynomial elliptic curve (ECC)
 - > Shares FGU multiplier
 - > Cipher/Hash Unit
 - > RC4, DES/3DES, AES-128/192/256
 - > MD5, SHA-1, SHA-256
 - > Designed to achieve wire-speed on both 10Gb Ethernet ports
 - > Facilitates wire-speed encryption and decryption
- DMA engine shares core's crossbar port

Integrated Networking

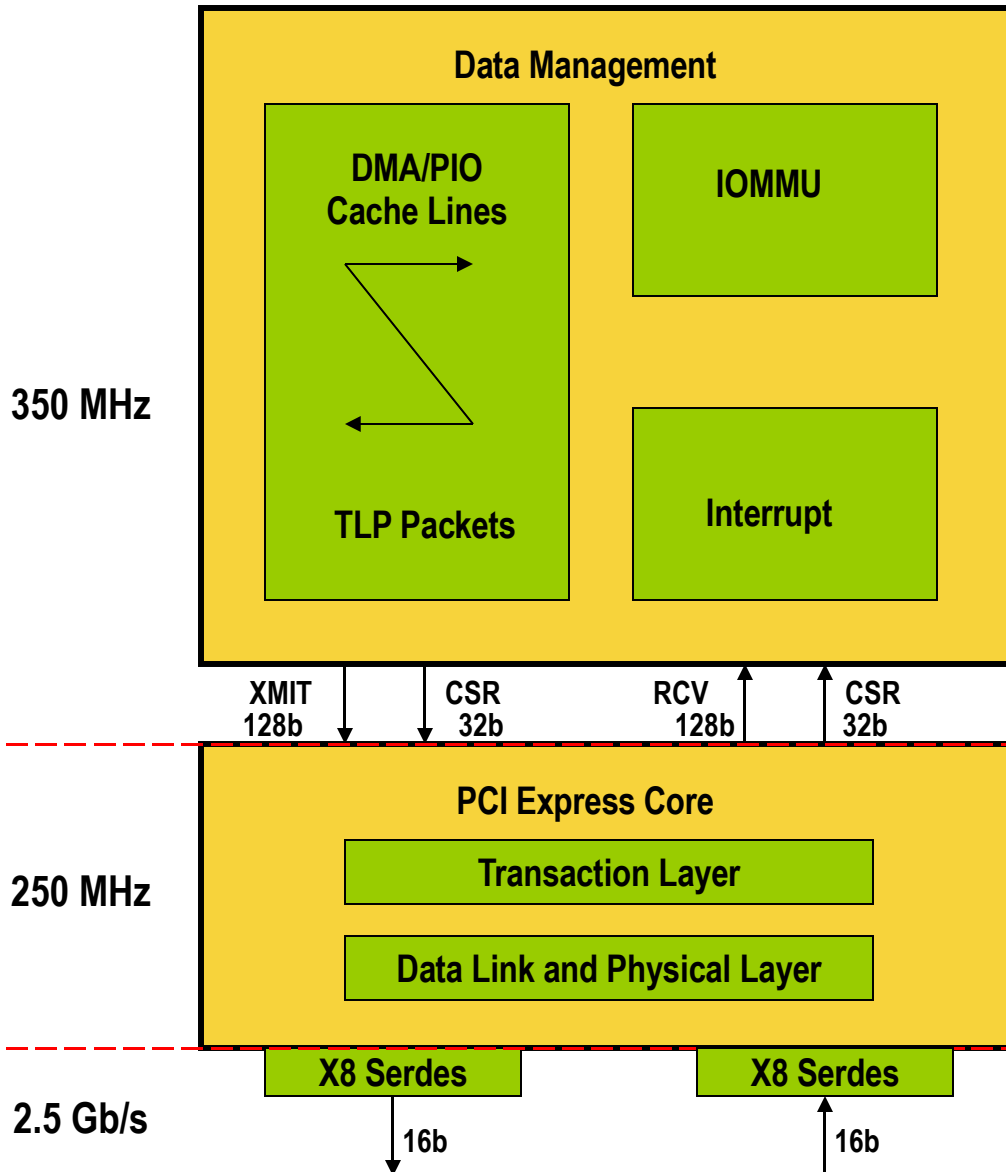


- Integrate networking for better overall performance
 - All network data is sourced from and destined to main memory
 - Integration minimizes impact of memory
 - Get networking closer to memory to reduce latency
 - Able to take full advantage of higher memory bandwidth
 - Eliminates inherent inefficiencies of I/O protocol translation

Networking Features

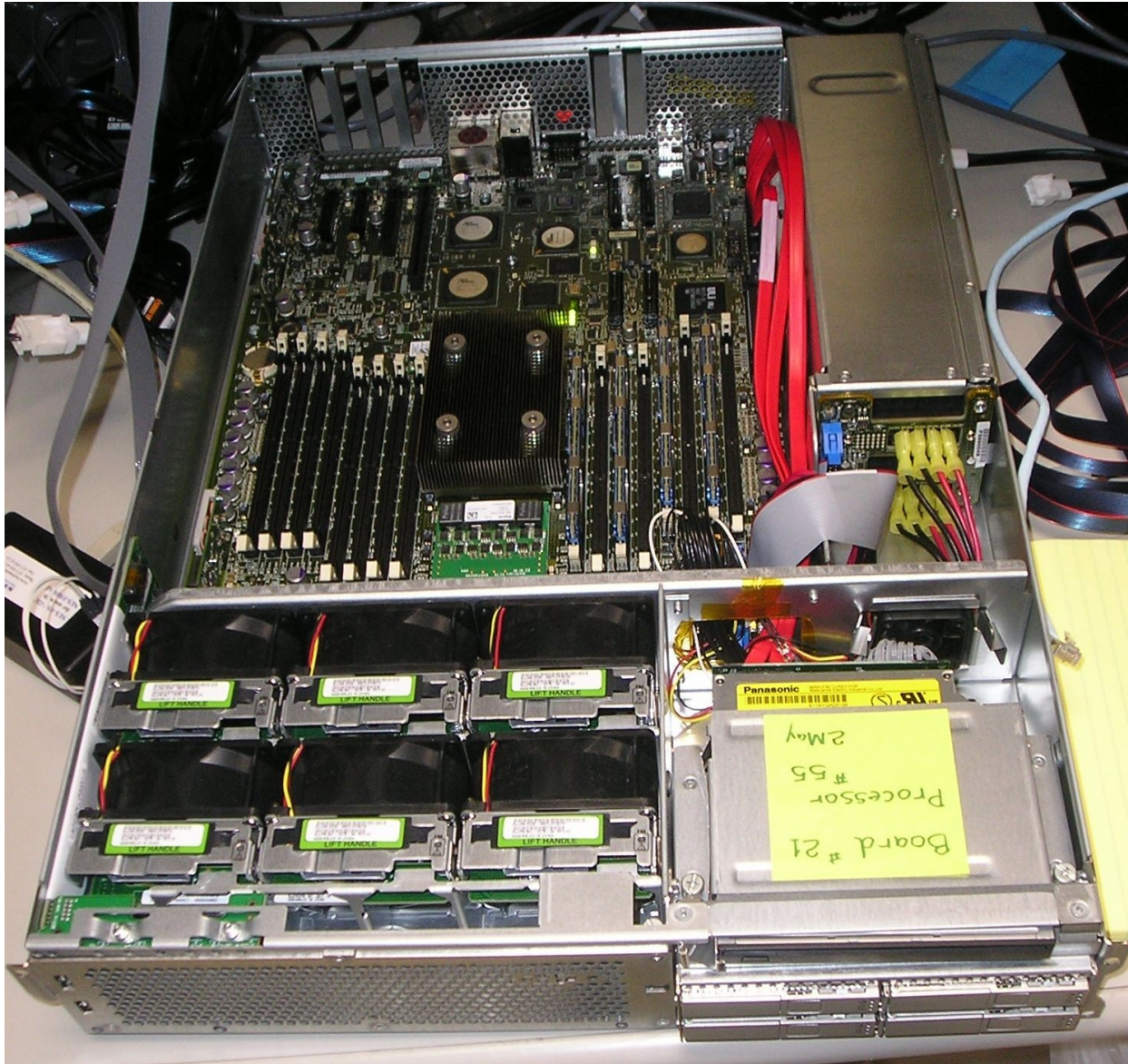
- Line Rate Packet Classification (~30M pkt/s)
 - > Based on Layer 1/2/3/4 of the protocol stack
- Multiple DMA Engines
 - > Matches DMAs to threads
 - > Binding flexibility between DMAs and ports
 - > 16 transmit + 16 receive DMA channels
- Virtualization Support
 - > Supports up to 8 partitions
 - > Interrupts may be bound to different hardware threads
- Dual Ethernet ports
 - > 2 dual-speed MACs (10G/1G) with integrated serdes

PCI-Express



- PCI-Express operates at 2.5 Gb/s per lane per direction
- Point-to-point, dual-simplex chip interconnect
- Transfers are in packets with headers and max data payloads from 128B to 512B
- IOMMU supports I/O virtualization and process device isolation by using PCIE's BDF#
- MSI Support
 - Event queue accumulates MSIs
 - Allows many MSIs to be serviced upon an interrupt
- Total I/O bandwidth is 3-4 GB/s with max payload sizes of 128B to 512B

Niagara2 System Status



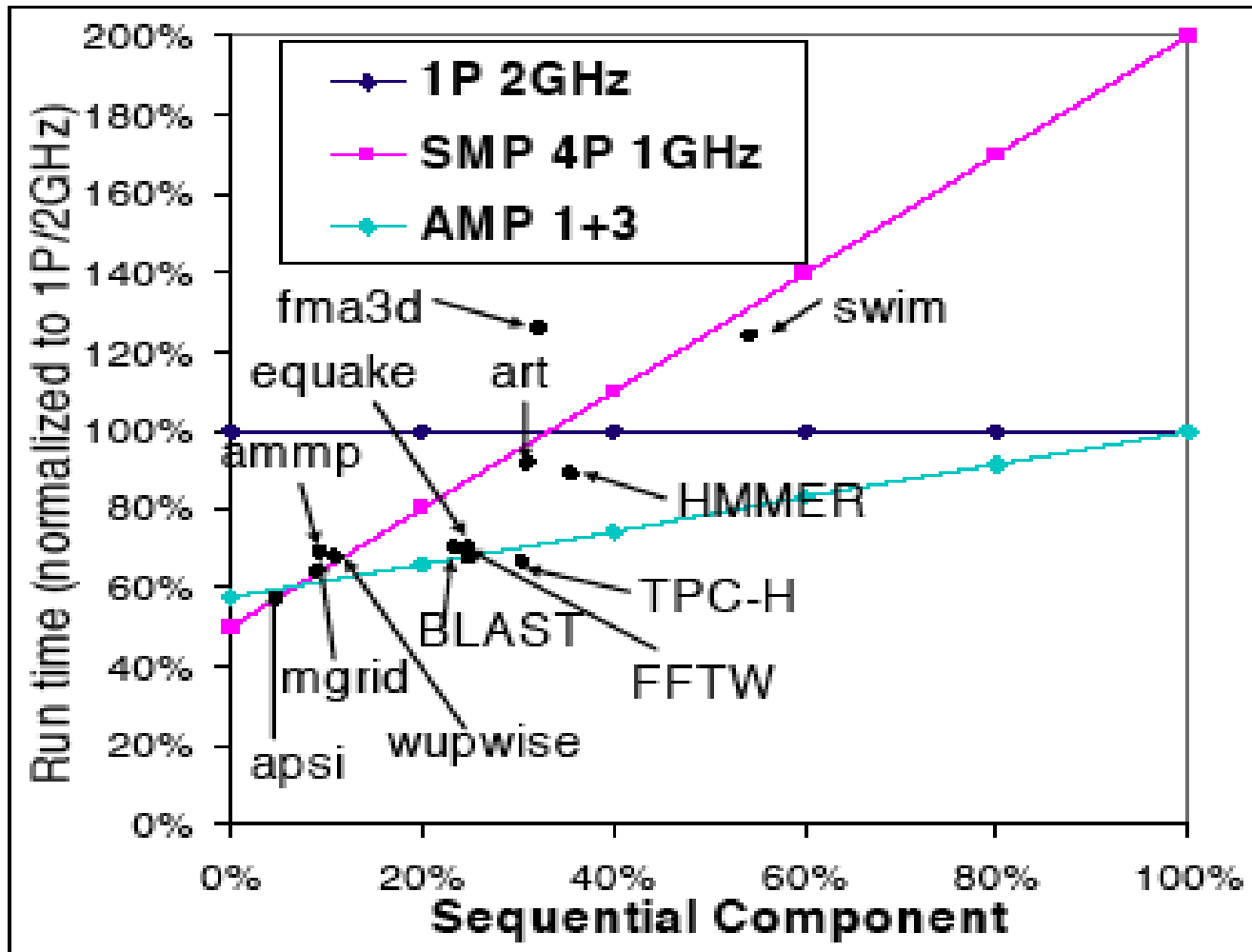
- First silicon arrived at the end of May
- Booted Solaris in 5 days
- Current systems are fully operational
- Expect systems to ship in 2H2007

Summary

- Niagara2 combines all major server functions on one chip
 - > Integrated networking
 - > Integrated PCI-Express
 - > Embedded wire-speed cryptography
- Niagara2 has improved performance vs. UltraSparc T1
 - > Better integer throughput and throughput/watt (>2x)
 - > Improved integer single-thread performance (>1.4x)
 - > Better floating-point throughput (>10x)
 - > Better floating-point single-thread performance (>5x)
- Enables new generation of power-efficient, fully-secure datacenters

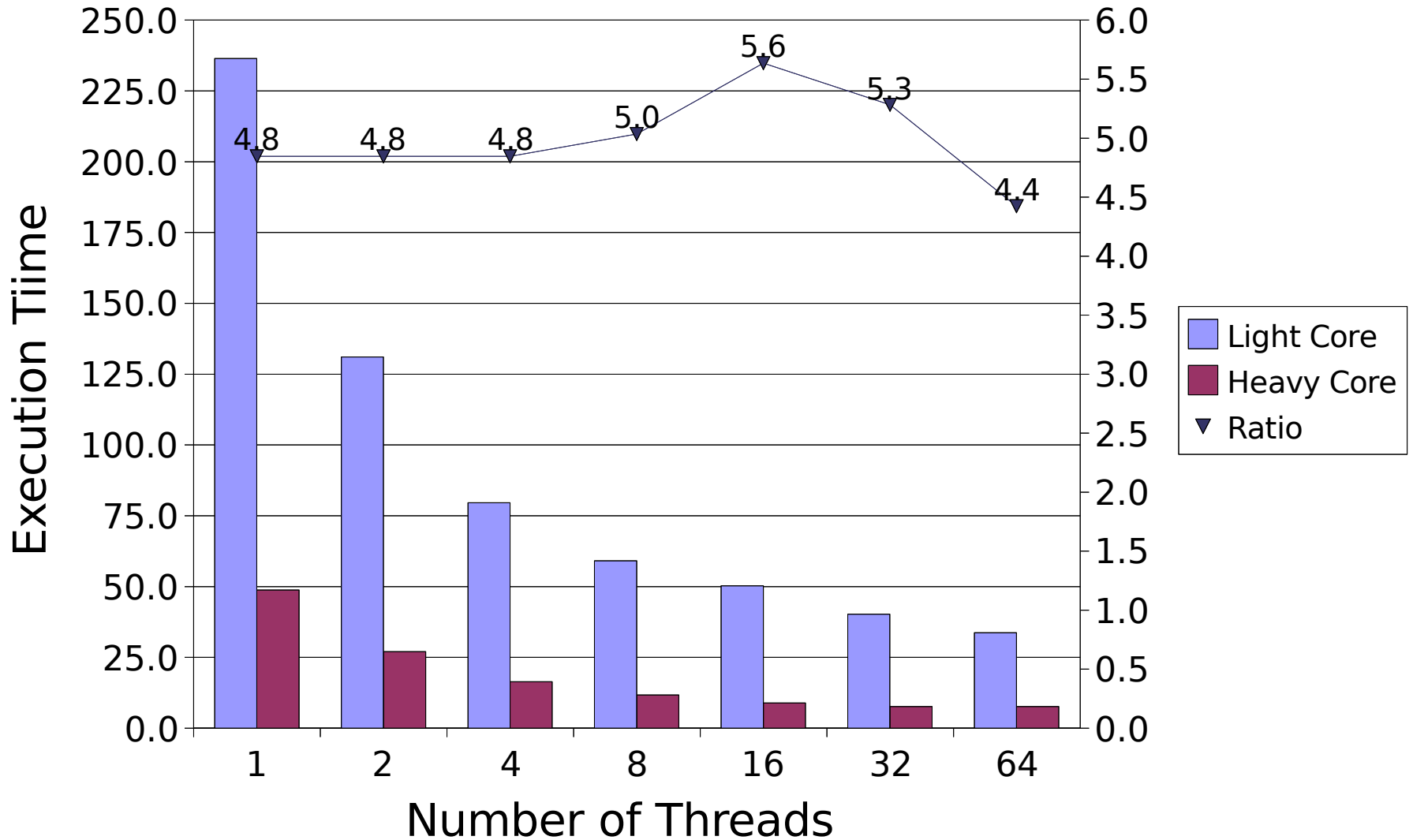
ROCK

A Sample of Target Applications

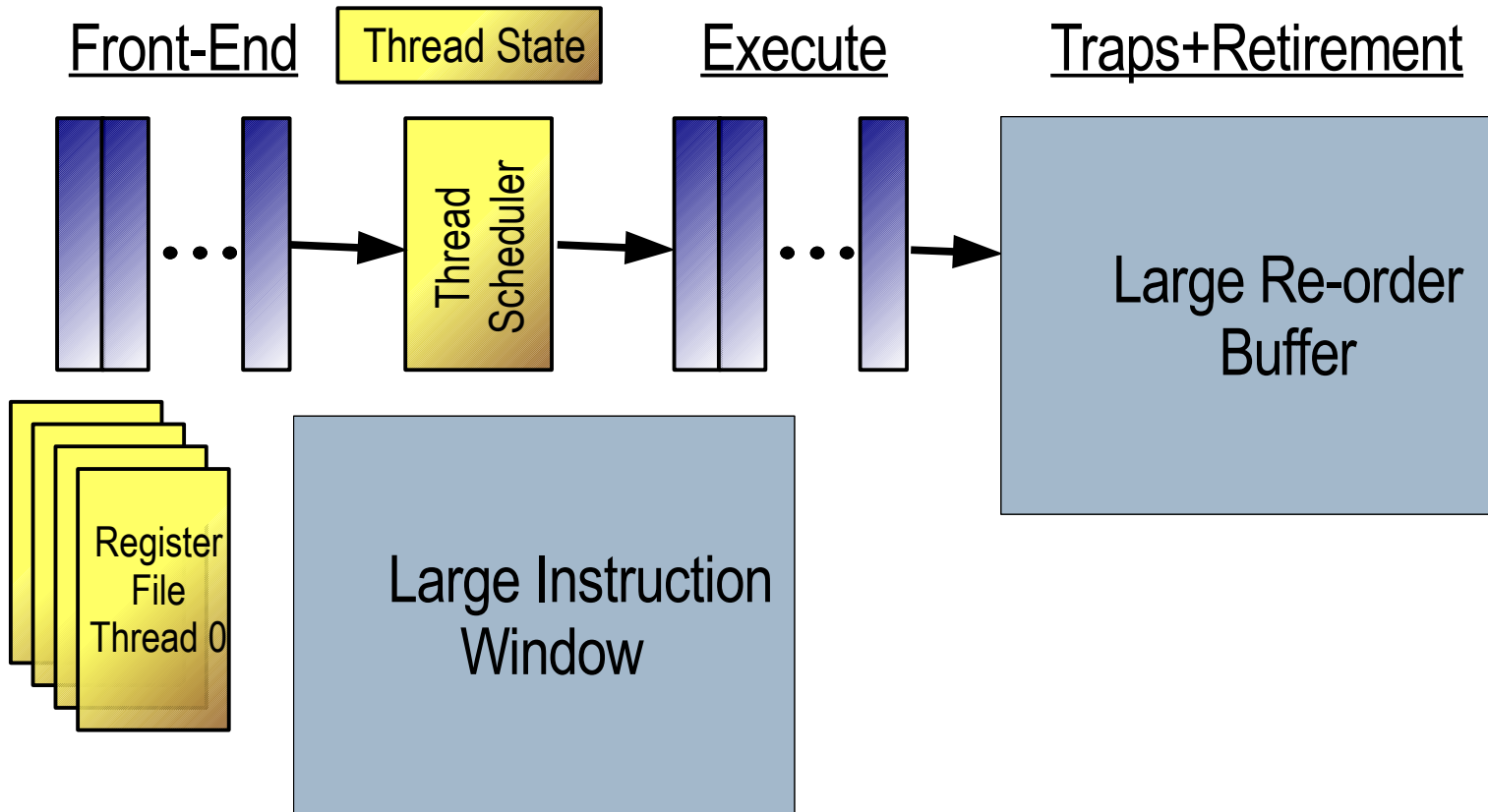


Source: CPI Throttling paper from Intel ISCA 2005

Impact of Amdahl's Law (90%) - Tapered

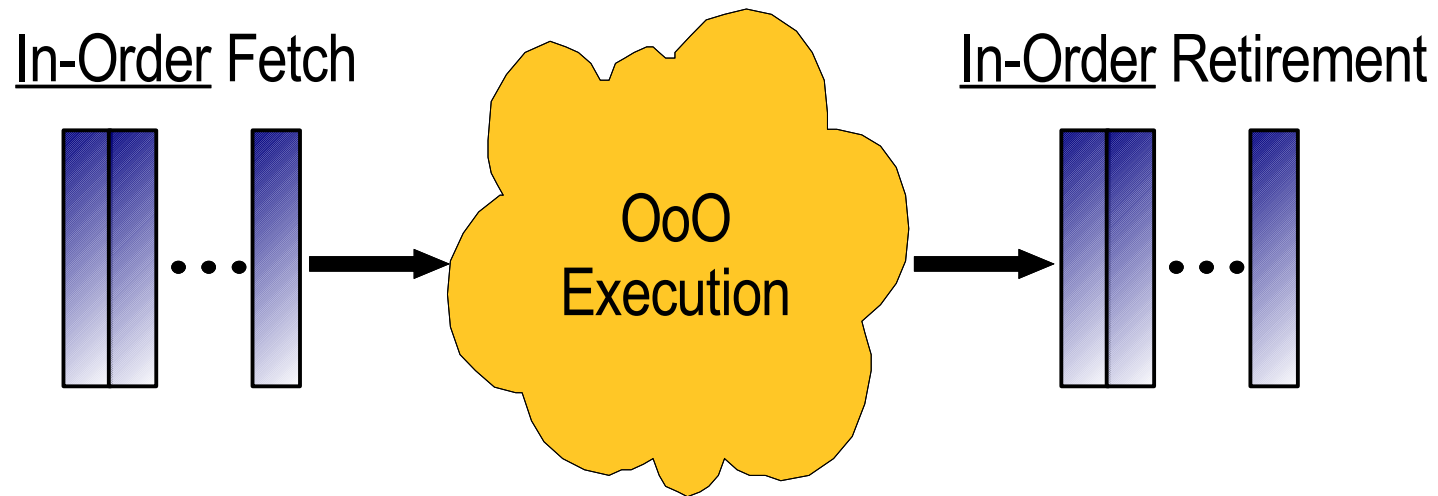


Traditional OoO



- Too large, too power hungry, limits cycle time
- Too small (depth), diminishing return

OoO So Far



Next Decade

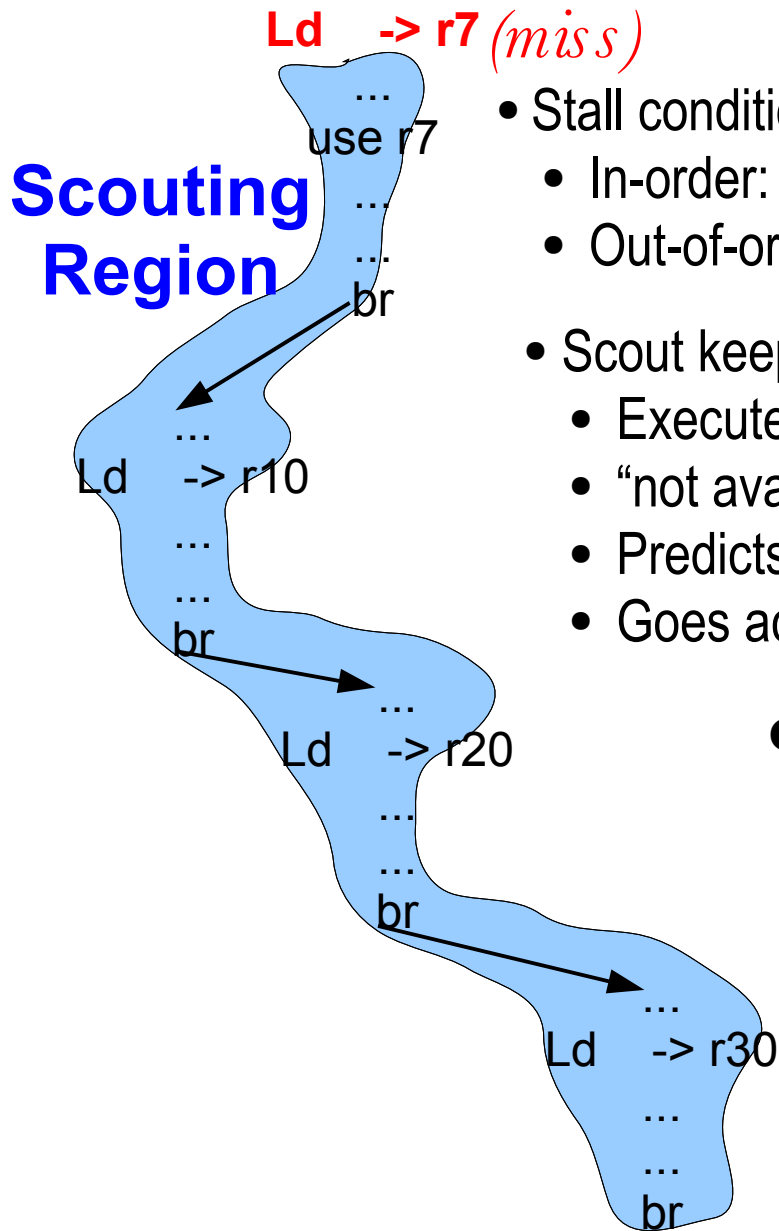
- Checkpoint-based architecture
- Thread-based structures
- Software threads, hardware threads, assist threads, multiple instruction pointers
- Much deeper **SRAM-based** structures
- Much more efficient pipelines
 - > Power
 - > CPI

Key Micro-Architecture Decision

- Threading is king
- Use structures need for threading to speed up single thread performance
- Scouting
- Execute-ahead
- Simultaneous Speculative Threading
- More...

Scouting

Hardware Scout

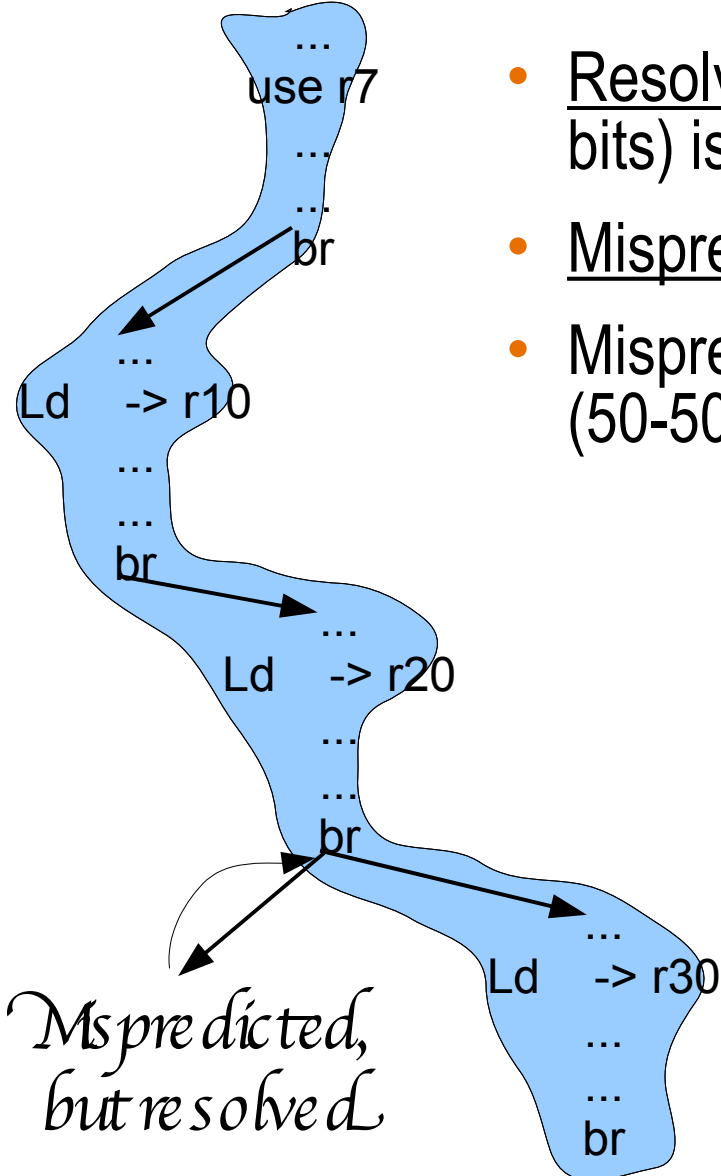


- Stall conditions launch a hardware thread – the scout
 - In-order: data dependencies, others
 - Out-of-order: full buffers (e.g. Instruction window)
- Scout keeps going down the instruction stream
 - Executes independent instructions
 - “not available” is propagated
 - Predicts branches
 - Goes across locks, memory barriers
- Goals for the Scout Thread
 - ▶ Get to next misses
 - ▶ Bring data into caches
 - ▶ Warm-up instruction cache
 - ▶ Warm-up branch predictor














Runtime Resolution

Ld -> r7

- Resolved: the data (register or conditional code bits) is available (not depending on a load miss)
- Mispredicted, but resolvable, are fine
- Mispredicted and unresolvable are not always bad (50-50)



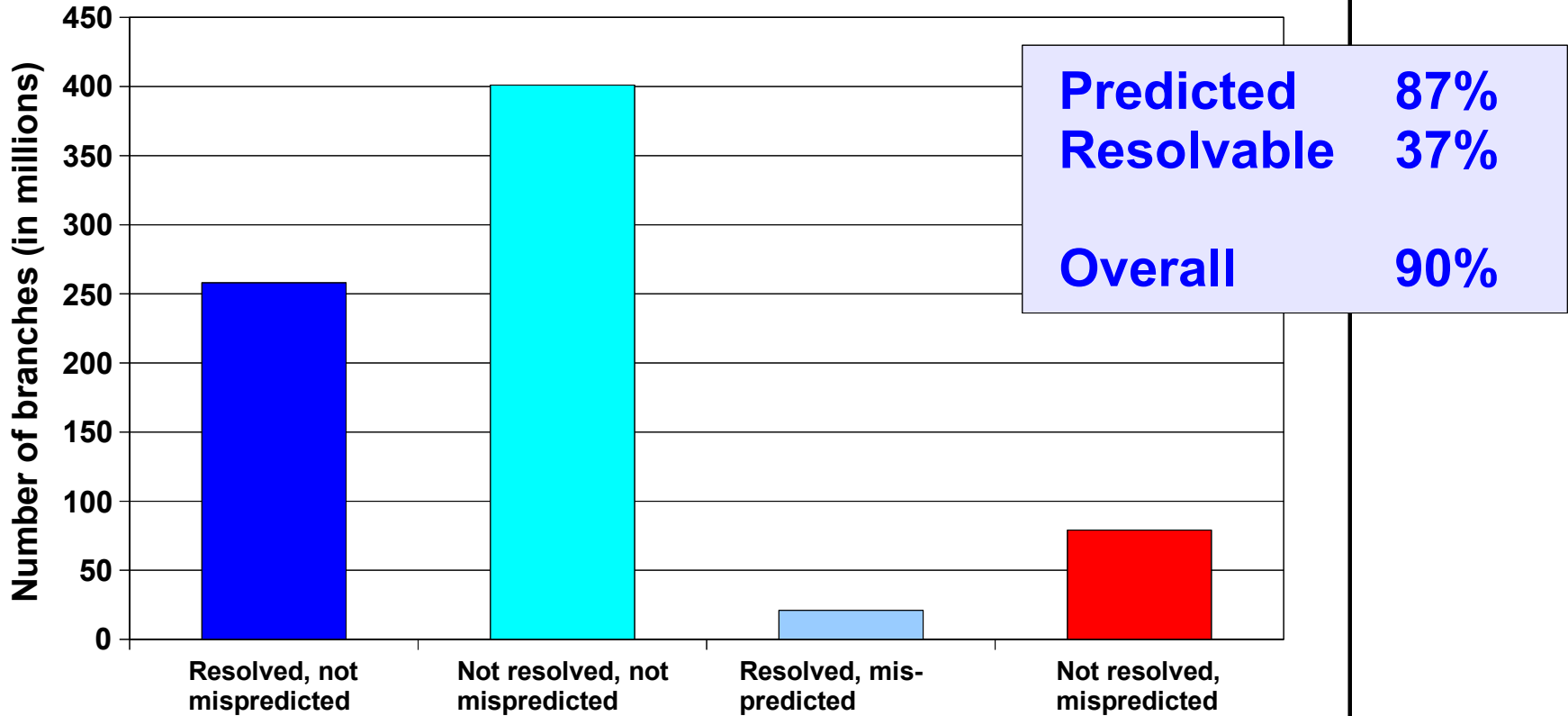
Hardware Scout - Example

	7334 lduw [%i1 + %g2], %g2	MISS	Data Return
	7335 or %o0, 0x3ca, %o1		
	7336 subcc %g2, 0x0, %g0	%g2 NT	
	7337 bne,pn %icc, 0x1001434d8	%icc NT	
	7338 sllx %o1, 12, %g5		
	7339 sllx %o2, 1, %g2	%g2 reused	
	7340 or %g5, 0x4c0, %o0		
	7341 lduw [%l4 + 0x160], %g5	MISS	
	... ~30 instructions		
	7388 or %g3, 0xef, %g2		
	7389 or %g4, 0xda, %g3		
	7390 sllx %g3, 12, %g3		
	...		

Scouting Results

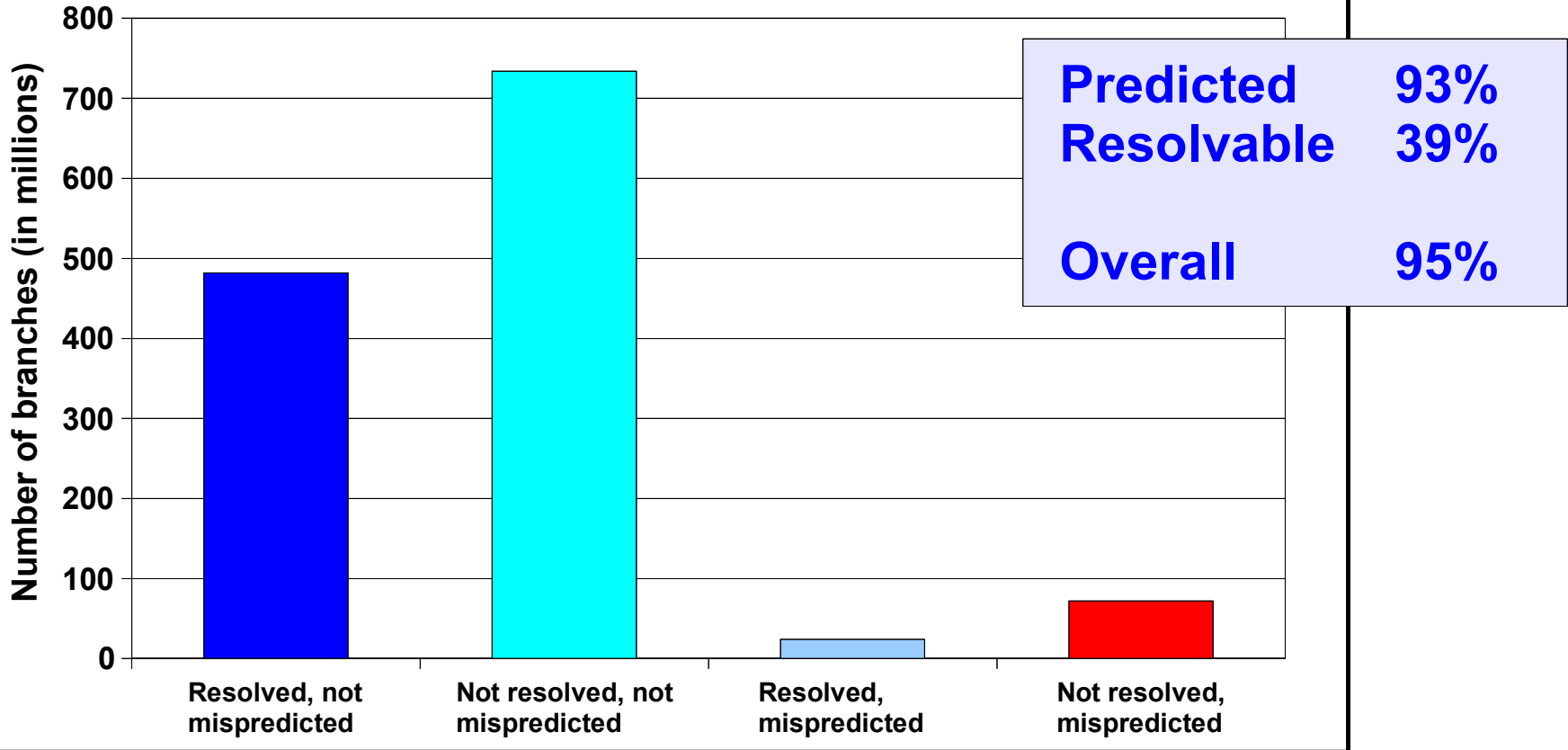
Hardware Scout - Simulation Results/1

Database Workload - Branch Statistics



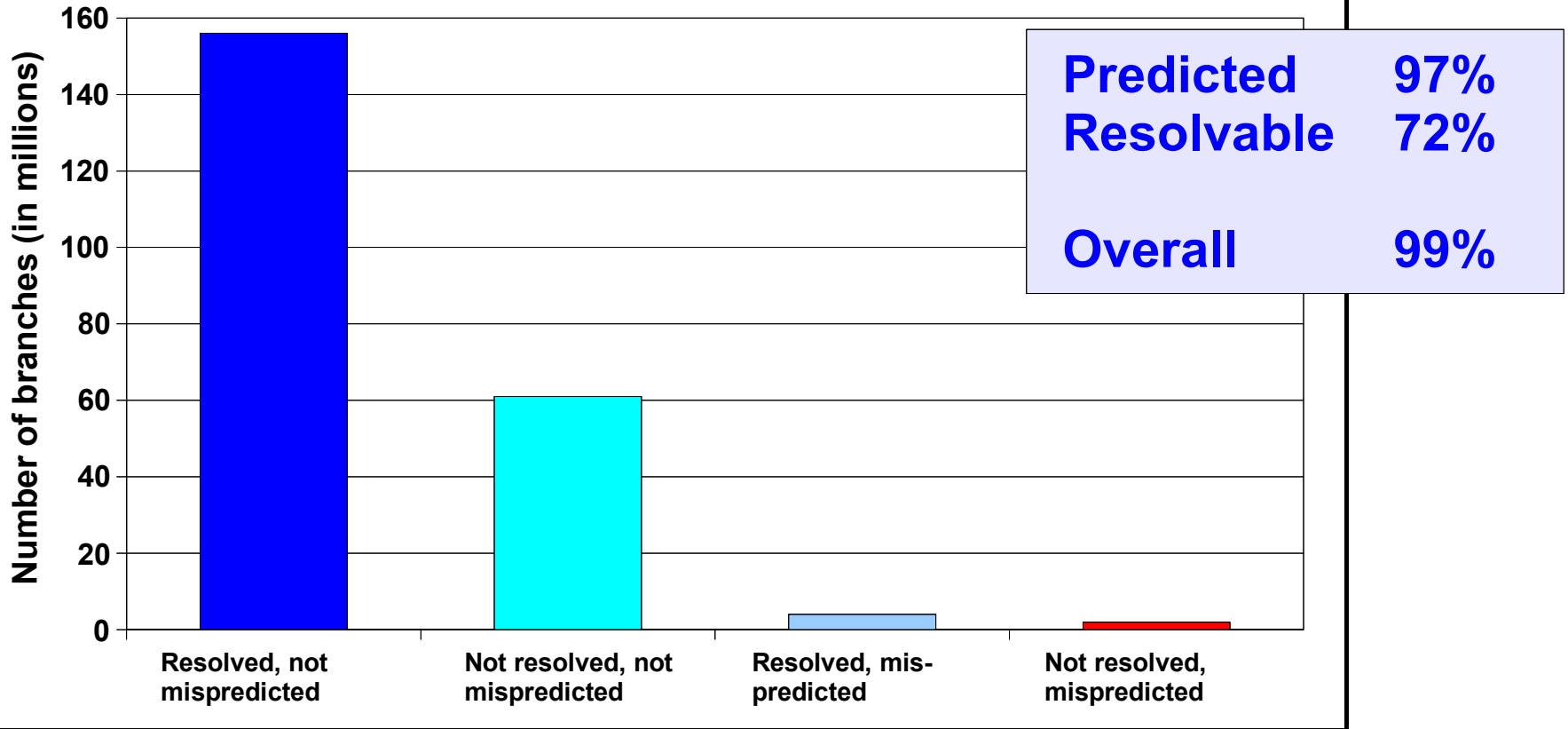
Hardware Scout - Simulation Results/2

SPECint2000 - Branch Statistics



Hardware Scout - Simulation Results/3

SPECfp2000 - Branch Statistics

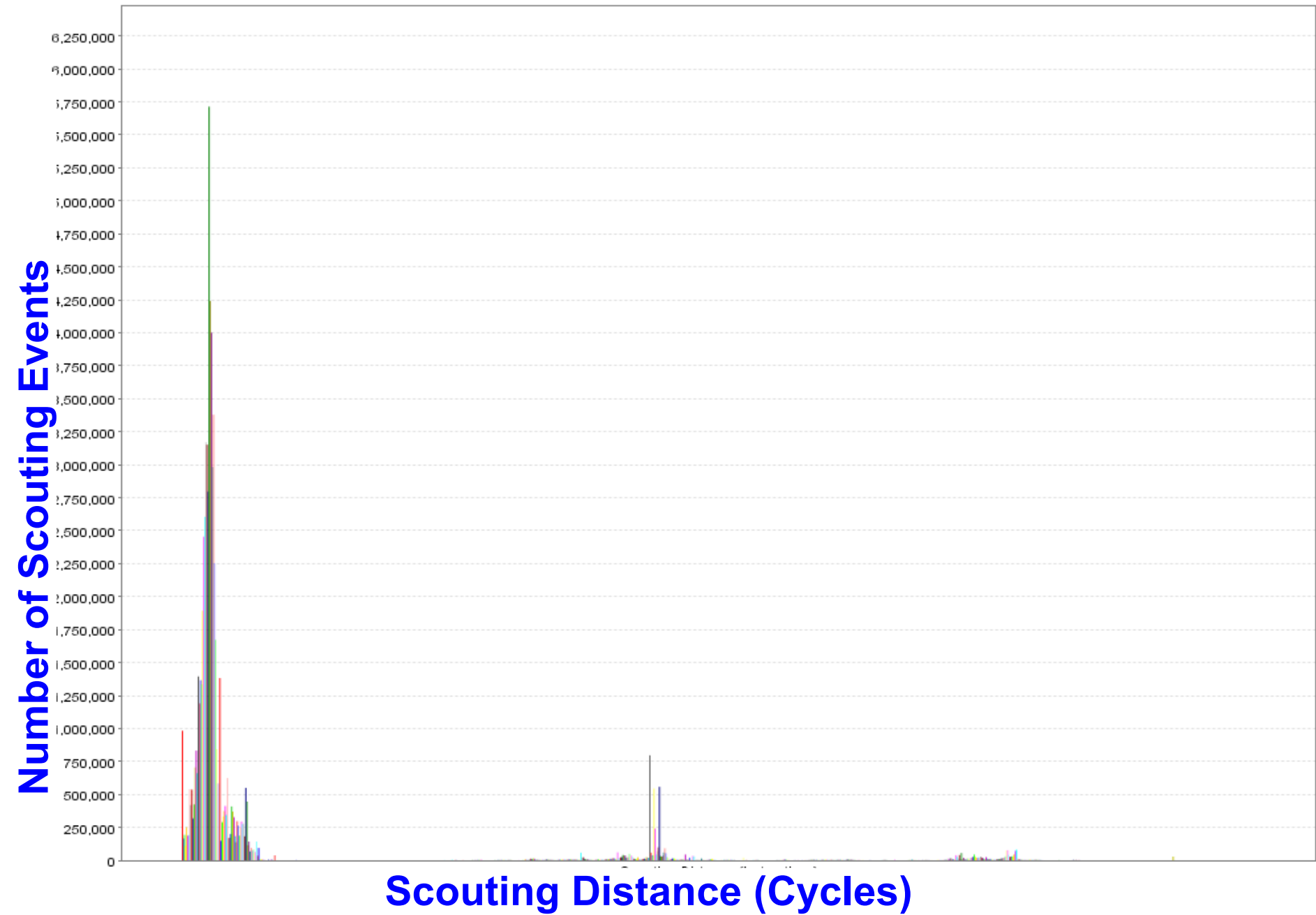


Scouting Distance

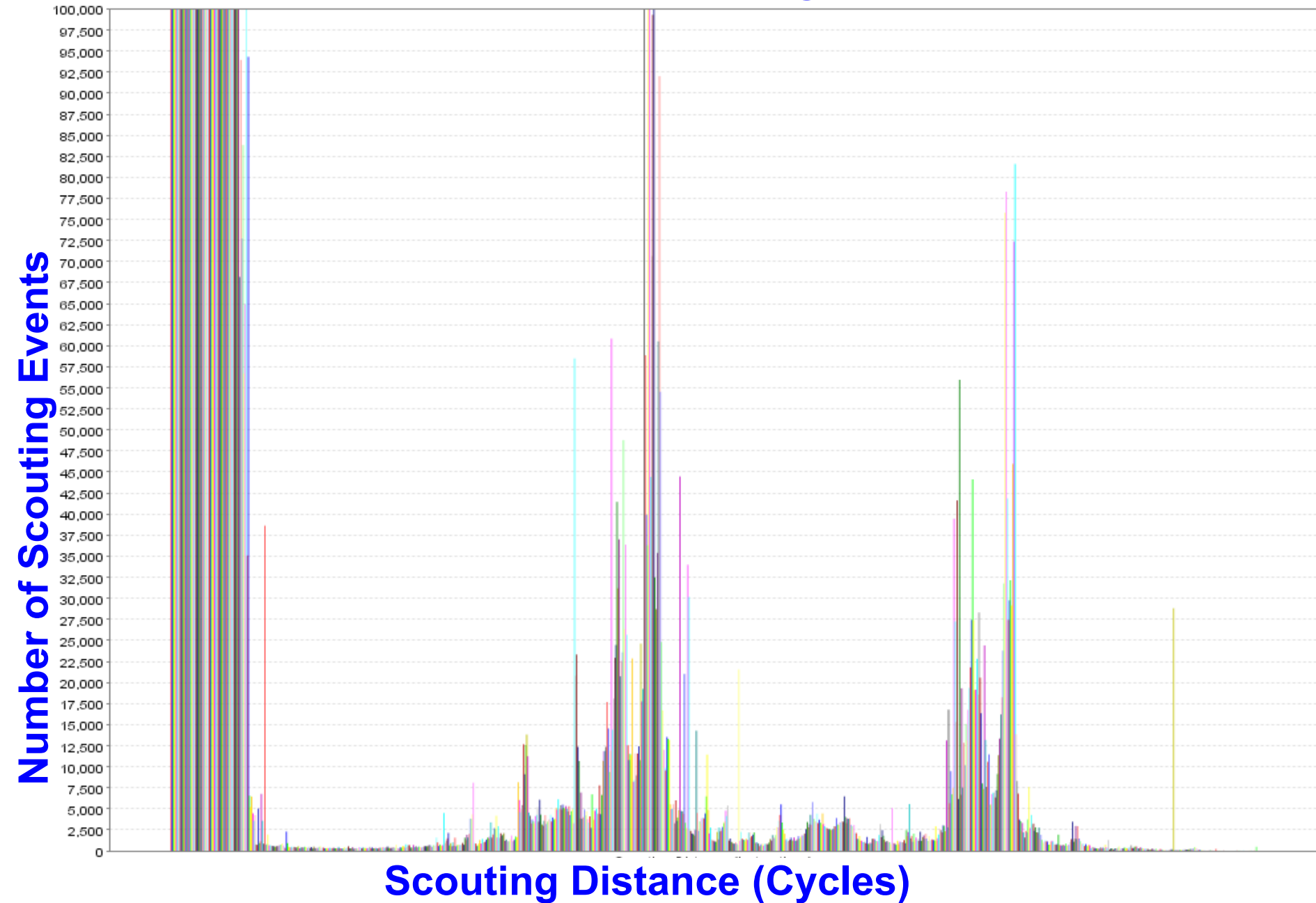
- ❑ ***Control speculation while scouting is very accurate***
- ❑ ***Our hardware scout implementation is faster than normal execution - Rips through instructions faster:***
 - ***Doesn't break groups as severely***
 - ***Doesn't have same resource constraints***
 - ***Even groups dependent instructions***

How Deep Can The Scout Thread Go ?

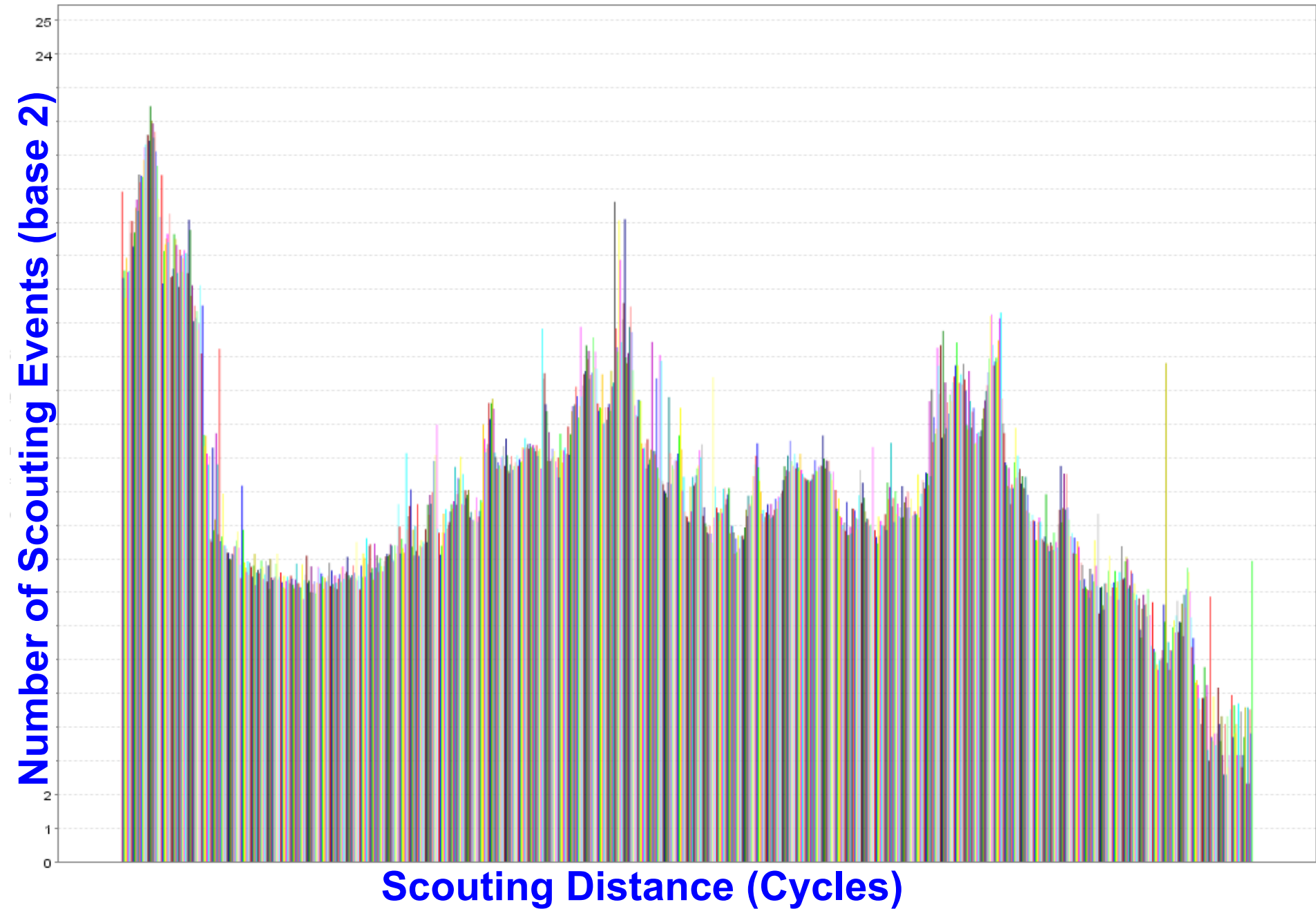
SPECfp2000 - Scouting Distance



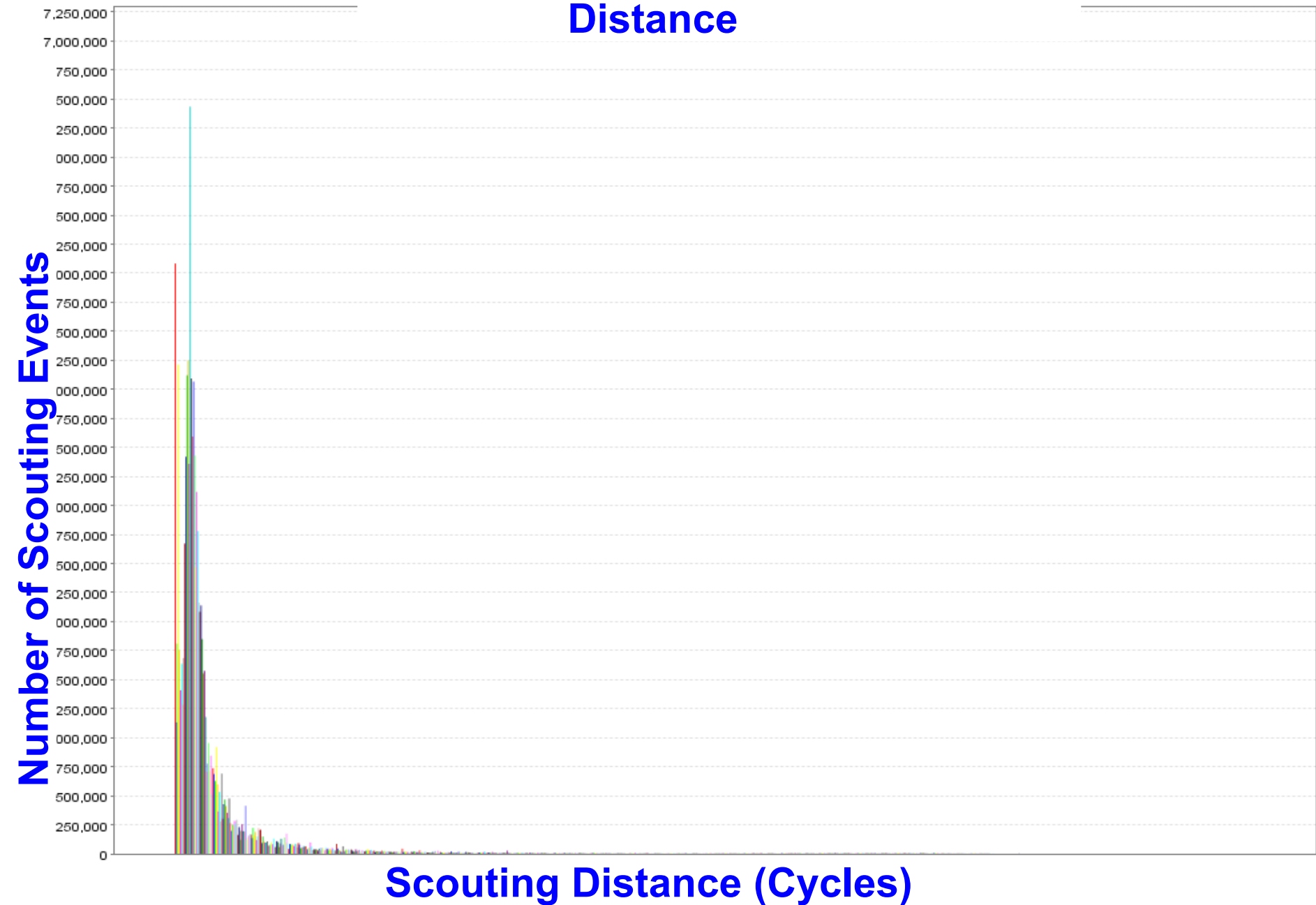
SPECfp2000 - Scouting Distance



SPECfp2000 - Scouting Distance

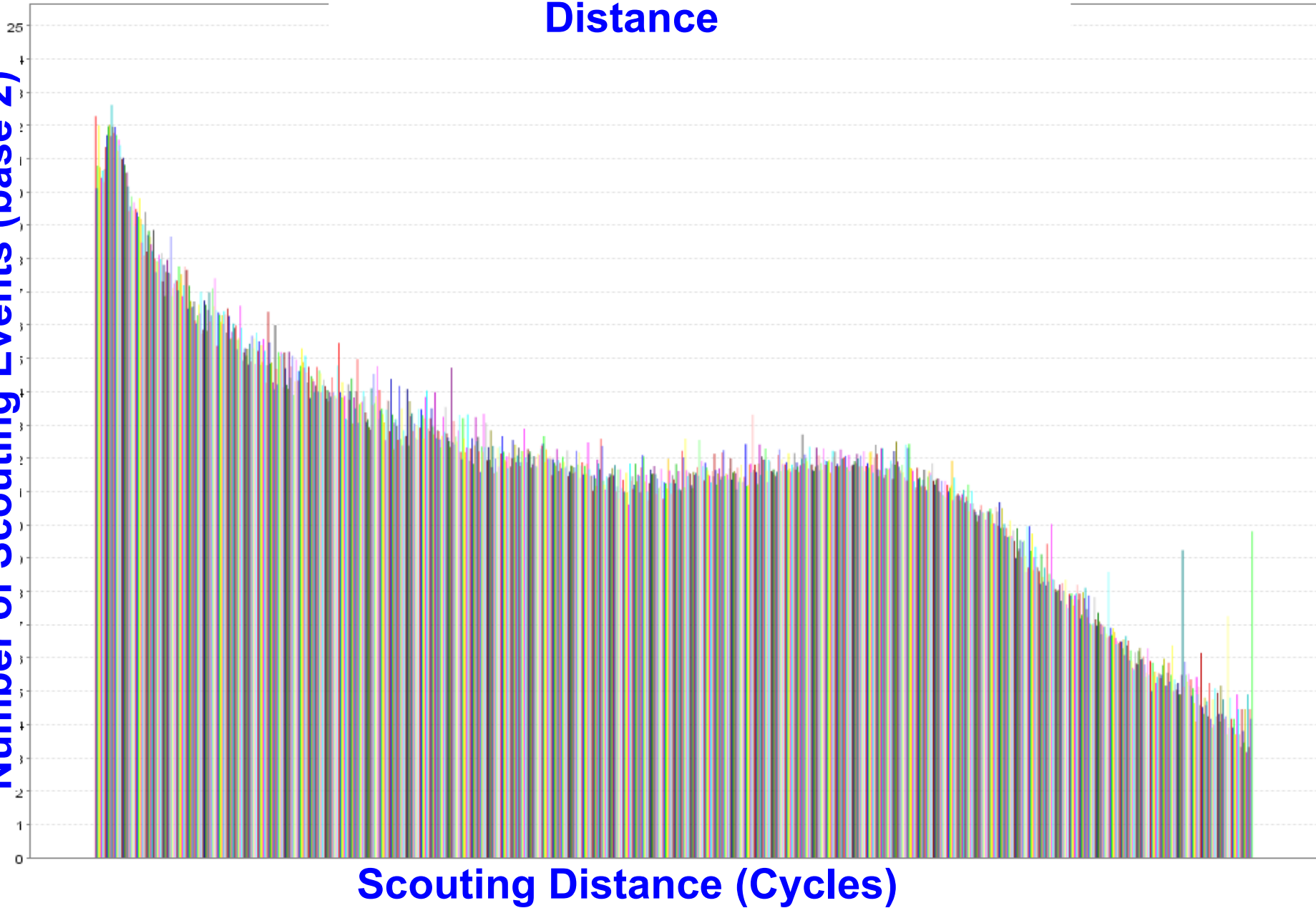


Database Workload - Scouting Distance



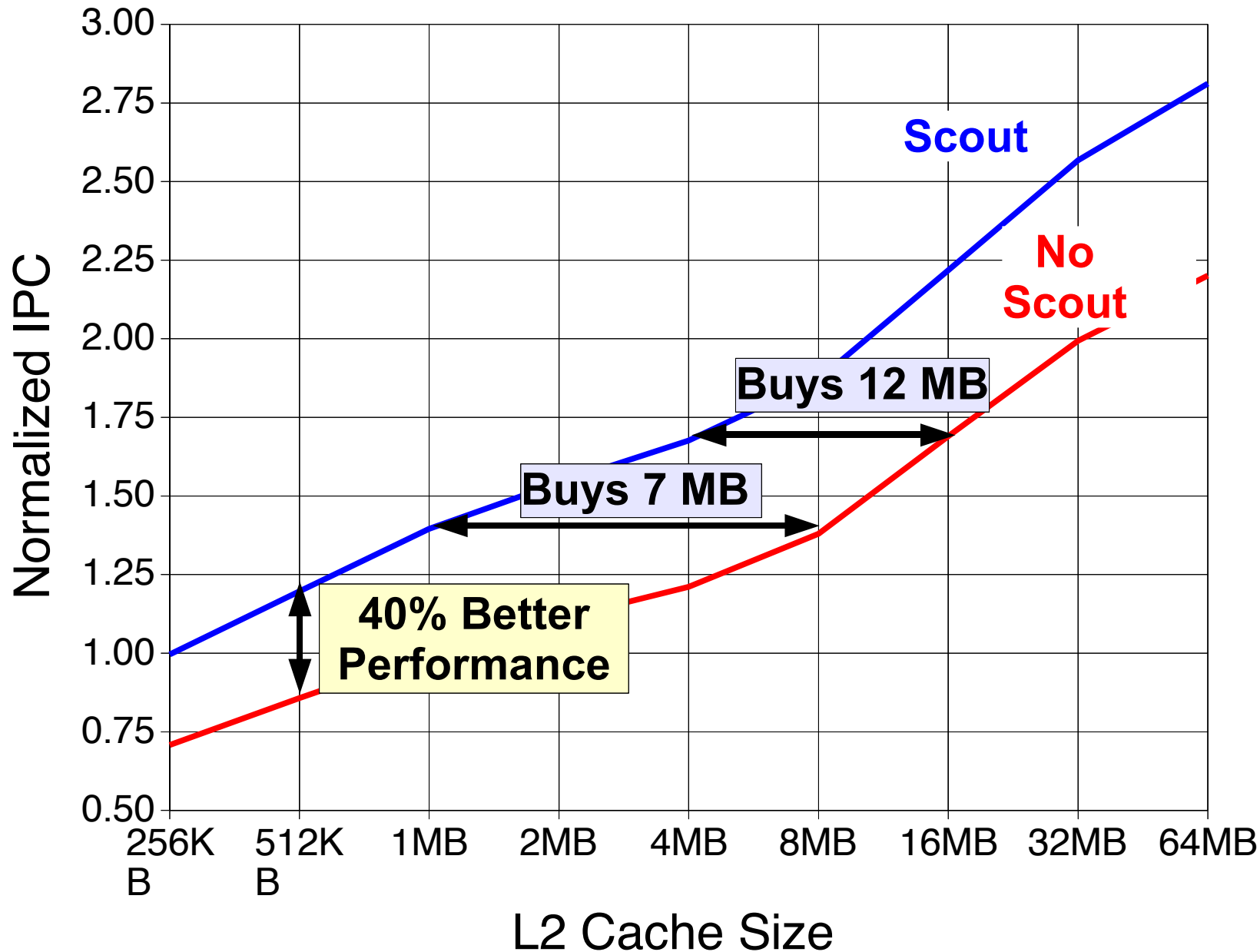
Database Workload - Scouting Distance

Number of Scouting Events (base 2)

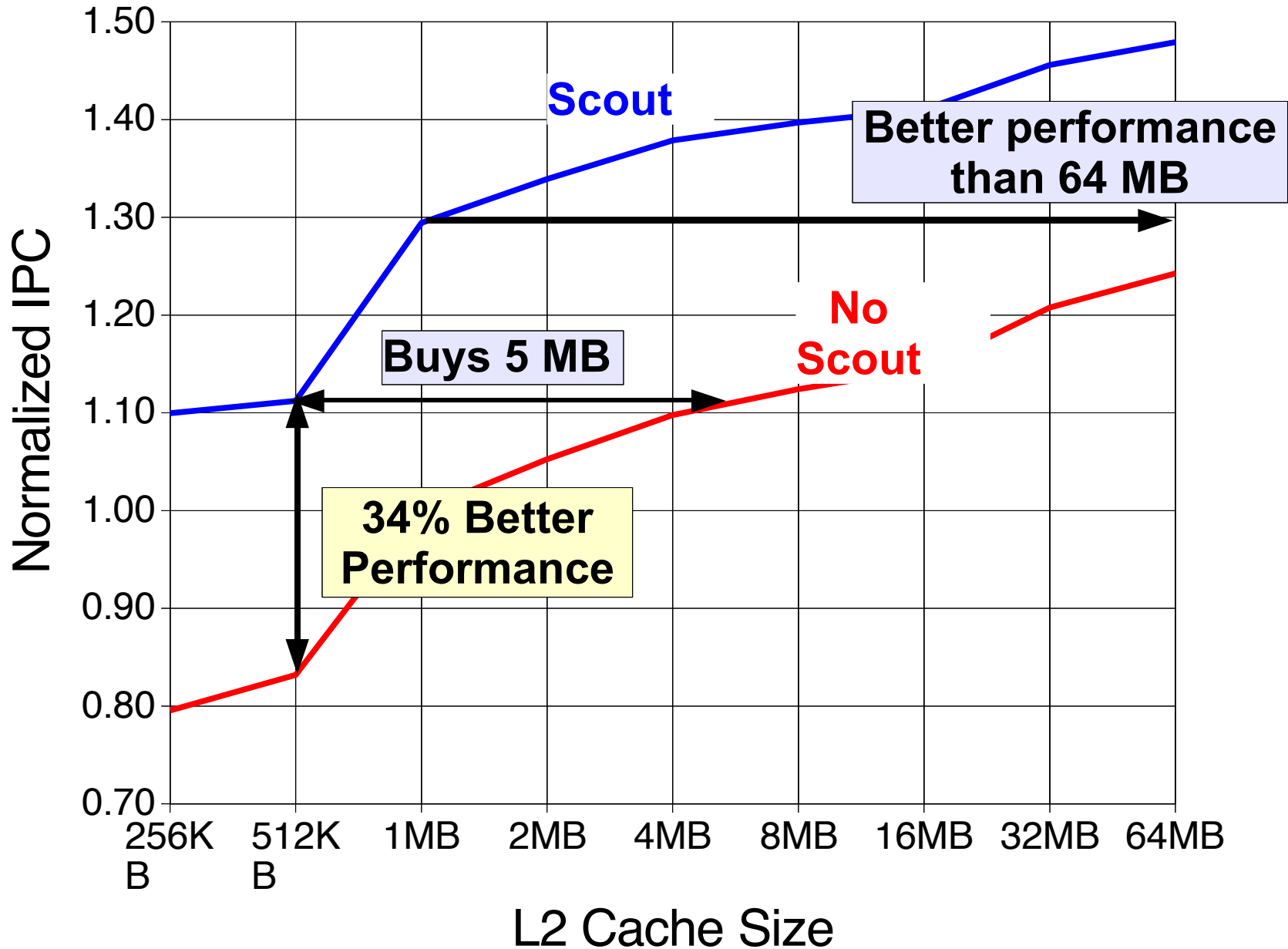


Scouting Distance (Cycles)

Scouting Improvement - Database



Scouting Improvement - SPECfp2000




**10 SLIDES DELETED FOR PUBLIC
DISTRIBUTION**

Summary

- Threads are very powerful
 - > For running threads
 - > For assisting the main thread
- Checkpoint based architecture enables deep speculation
- Scouting takes care of MLP
- But...

Summary (cont)

- More importantly this all results in
 - > A small core
 - > A low power core
- We want 10-20 of these in 65nm
- Need matching
 - > Memory system
 - > Coherent interconnect
 - > I/O
- New world of innovation – enjoy the ride!

An aerial photograph of a rural landscape. In the foreground, there is a field of tall, brown stalks, possibly corn. A small, simple house with a red roof is situated in the middle ground, surrounded by a fence. The background consists of rolling green and yellow fields under a sky with large, white, fluffy clouds.

A Modern High-Performance Processor Pipeline

Marc Tremblay, Ph.D.
marc.tremblay@sun.com