UltraSPARC T1: A 32-threaded CMP for Servers

James Laudon
Distinguished Engineer
Sun Microsystems
james.laudon@sun.com

Outline

- Server design issues
  - > Application demands
  - > System requirements
- Building a better server-oriented CMP
  - > Maximizing thread count
  - > Keeping the threads fed
  - > Keeping the threads cool
- UltraSPARC T1 (Niagara)
  - > Micro-architecture
  - > Performance
  - > Power

Attributes of Commercial Workloads

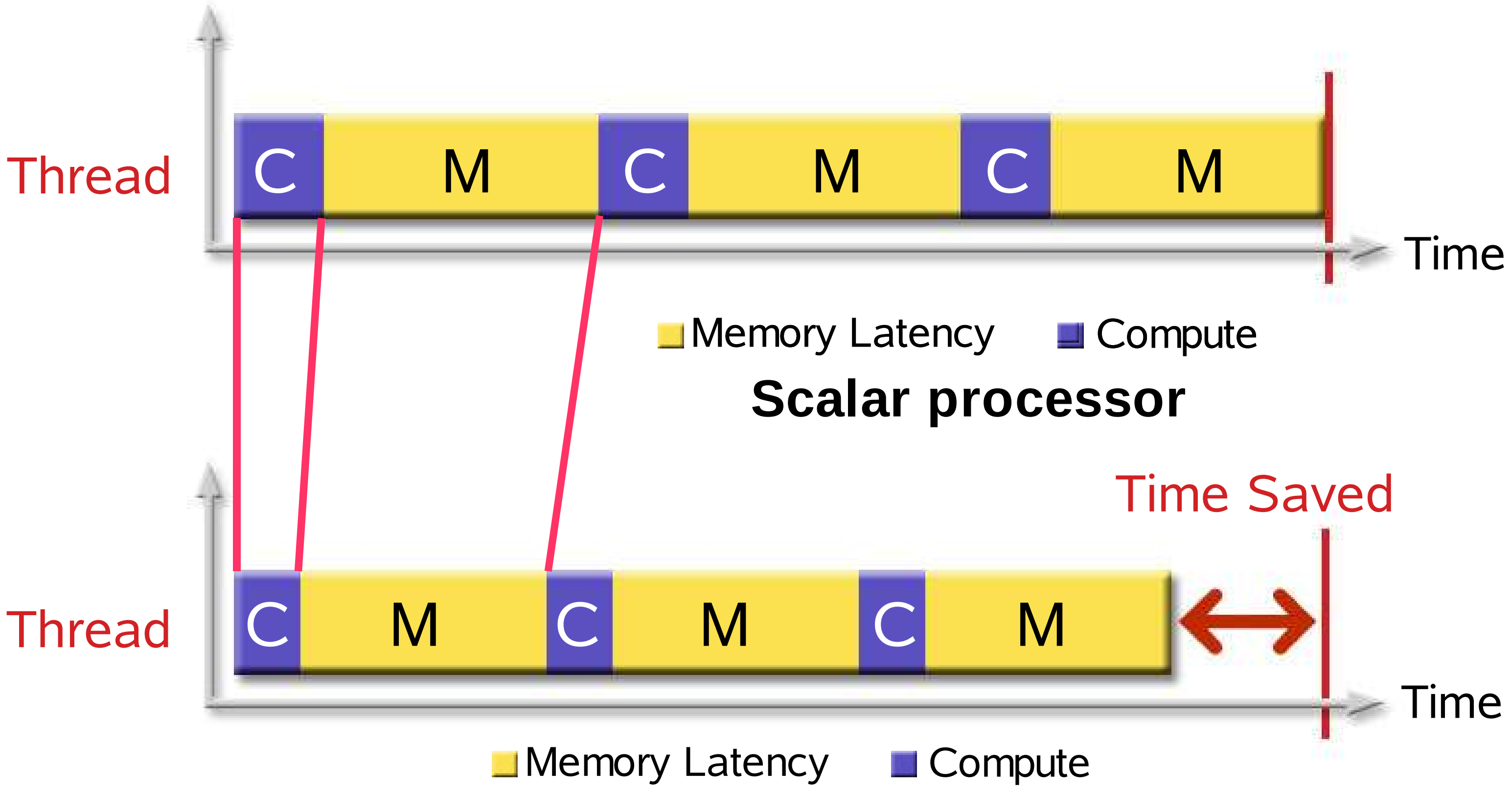
Attribute	Web99	jBOB (JBB)	TPC-C	SAP 2T	SAP 3T DB	TPC-H
Application Category	Web server	Server Java	OLTP	ERP	ERP	DSS
Instruction-level parallelism	low	low	low	med	low	high
Thread-level parallelism	high	high	high	high	high	high
Instruction/Data working set	large	large	large	med	large	large
Data sharing	low	med	high	med	high	med

- Adapted from “A Performance methodology for commercial servers,” S. R. Kunkel et al, IBM J. Res. Develop. vol. 44 no. 6 Nov 2000

Commercial Server Workloads

- SpecWeb05, SpecJappserver04, SpecJBB05, SAP SD, TPC-C, TPC-E, TPC-H
- High degree of thread-level parallelism (TLP)
- Large working sets with poor locality leading to high cache miss rates
- Low instruction-level parallelism (ILP) due to high cache miss rates, load-load dependencies, and difficult to predict branches
- Performance is bottlenecked by stalls on memory accesses
- Superscalar and superpipelining will not help much

ILP Processor on Server Application



Scalar processor

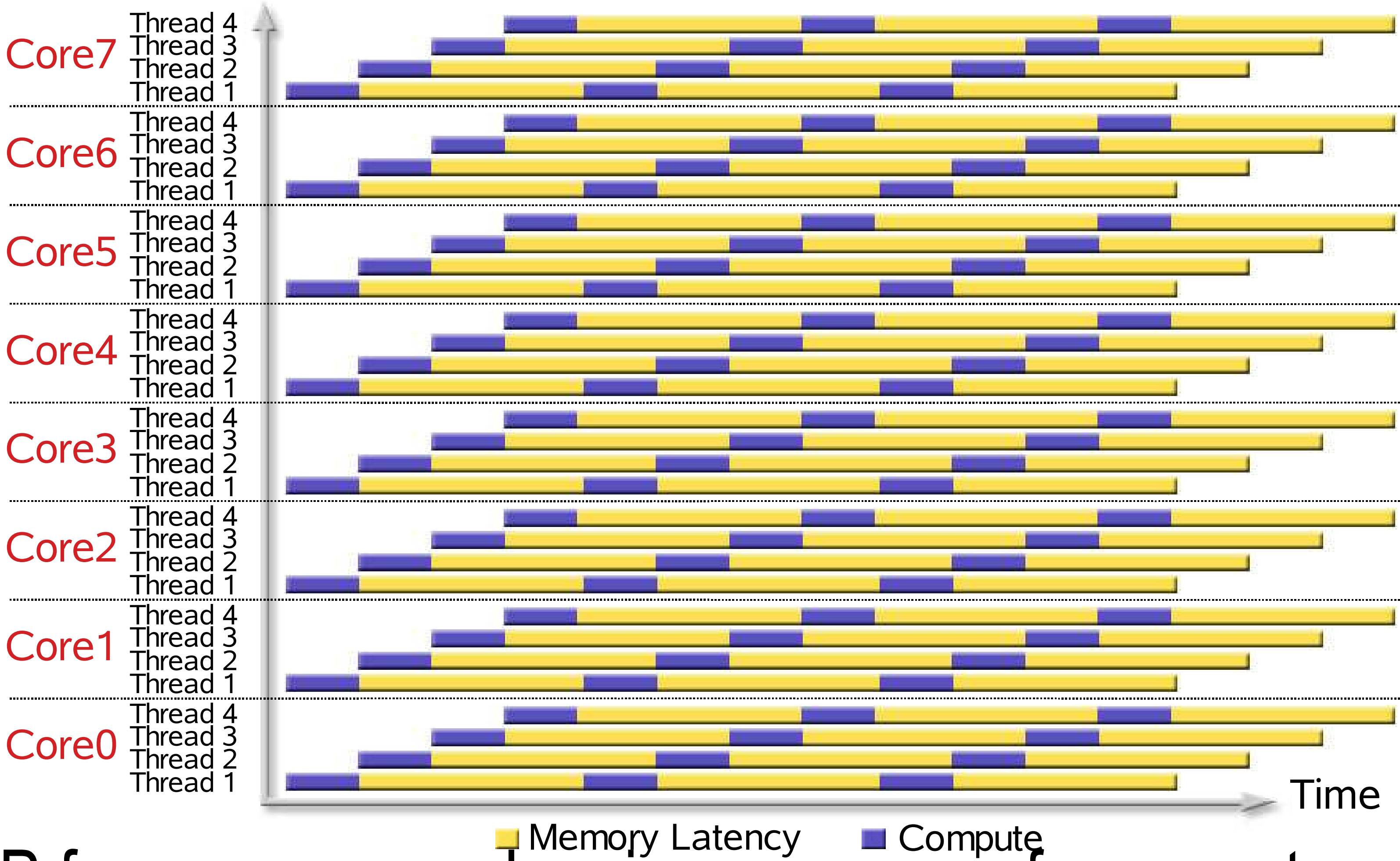
Processor optimized for ILP

ILP reduces the compute time and overlaps computation with L2 cache hits, but memory stall time dominates overall performance

Attacking the Memory Bottleneck

- Exploit the TLP-rich nature of server applications
- Replace each large, superscalar processor with multiple simpler, threaded processors
  - > Increases core count (C)
  - > Increases thread per core count (T)
  - > Greatly increases total thread count (C*T)
- Threads share a large, high-bandwidth L2 cache and memory system
- Overlap the memory stalls of one thread with the computation of other threads

TLP Processor on Server Application



TLP focuses on overlapping memory references to improve throughput; needs sufficient memory bandwidth

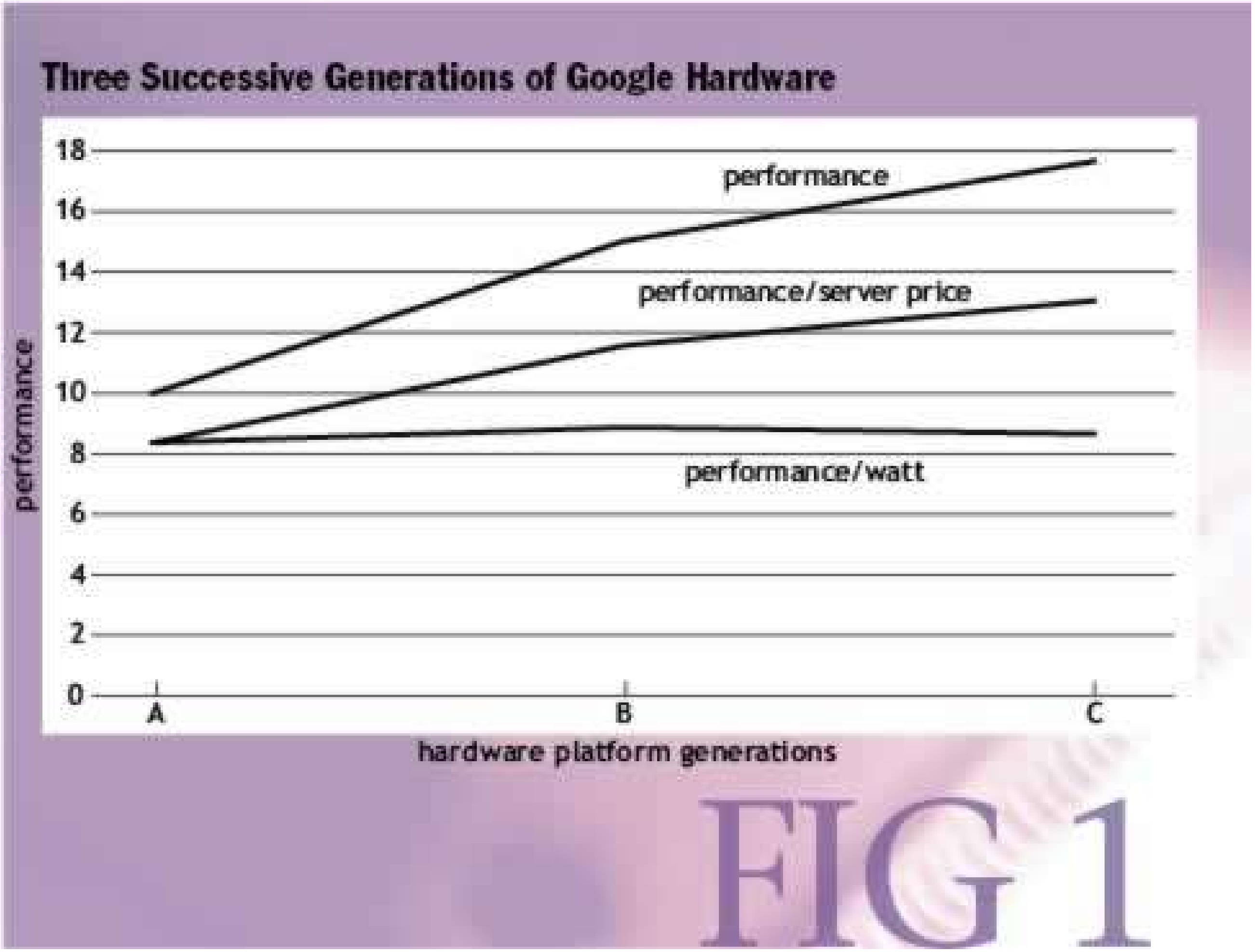
Server System Requirements

- Very large power demands
  - > Often run at high utilization and/or with large amounts of memory
  - > Deployed in dense rack-mounted datacenters
- Power density affects both datacenter construction and ongoing costs
- Current servers consume far more power than state of the art datacenters can provide
  - > 500W per 1U box possible
  - > Over 20 kW/rack, most datacenters at 5 kW/rack
  - > Blades make this even worse...

Server System Requirements

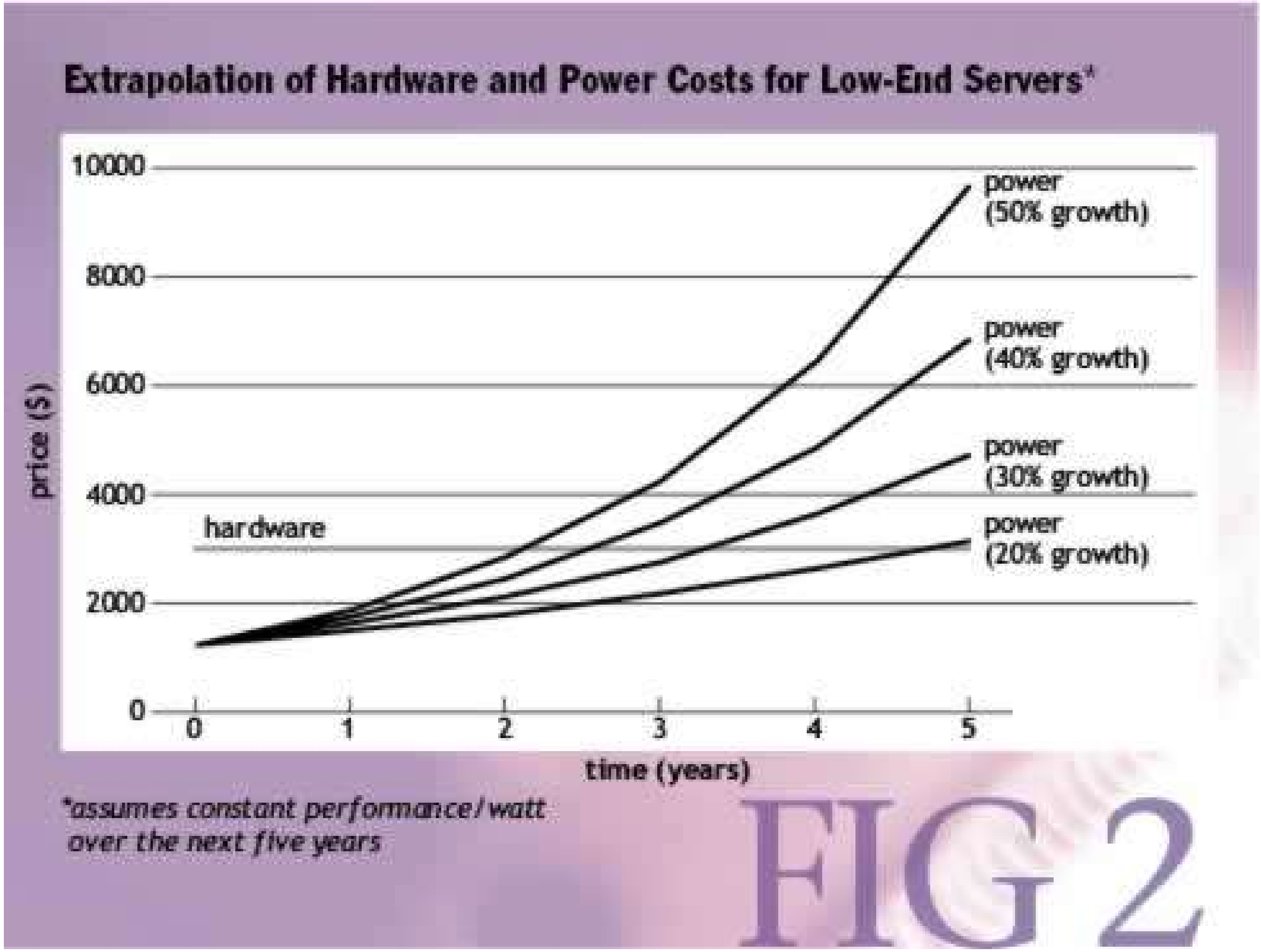
- Processor power is a significant portion of total
  - > Database: 1/3 processor, 1/3 memory, 1/3 disk
  - > Web serving: 2/3 processor, 1/3 memory
- Perf/watt has been flat between processor generations
- Acquisition cost of server hardware is declining
  - > Moore's Law – more performance at same cost or same performance at lower cost
- Total cost of ownership (TCO) will be dominated by power within five years
- The “Power Wall”

Performance/Watt Trends



Source: L. Barroso, *The Price of Performance*, ACM Queue vol 3 no 7

Impact of Flat Perf/Watt on TCO



Source: L. Barroso, *The Price of Performance*, ACM Queue vol 3 no 7

Implications of the “Power Wall”

- With TCO dominated by power usage, the metric that matters is performance/Watt
- Performance/Watt has been mostly flat for several generations of ILP-focused designs
  - Should have been improving as a result of voltage scaling ($fCV^2 + TI_{LC}V$)
  - C, T, $I_{LC,}$ and f increases have offset voltage decreases
- TLP-focused processors reduce f and C/T (per-processor) and can greatly improve performance/Watt for server workloads

Outline

- Server design issues
  - > Application demands
  - > System requirements
- Building a better server-oriented CMP
  - > Maximizing thread count
  - > Keeping the threads fed
  - > Keeping the threads cool
- UltraSPARC T1 (Niagara)
  - > Micro-architecture
  - > Performance
  - > Power

Building a TLP-focused processor

- Maximizing the total number of threads
  - Simple cores
  - Sharing at many levels
- Keeping the threads fed
  - Bandwidth!
  - Increased associativity
- Keeping the threads cool
  - Performance/watt as a design goal
  - Reasonable frequency
  - Mechanisms for controlling the power envelope

Maximizing the thread count

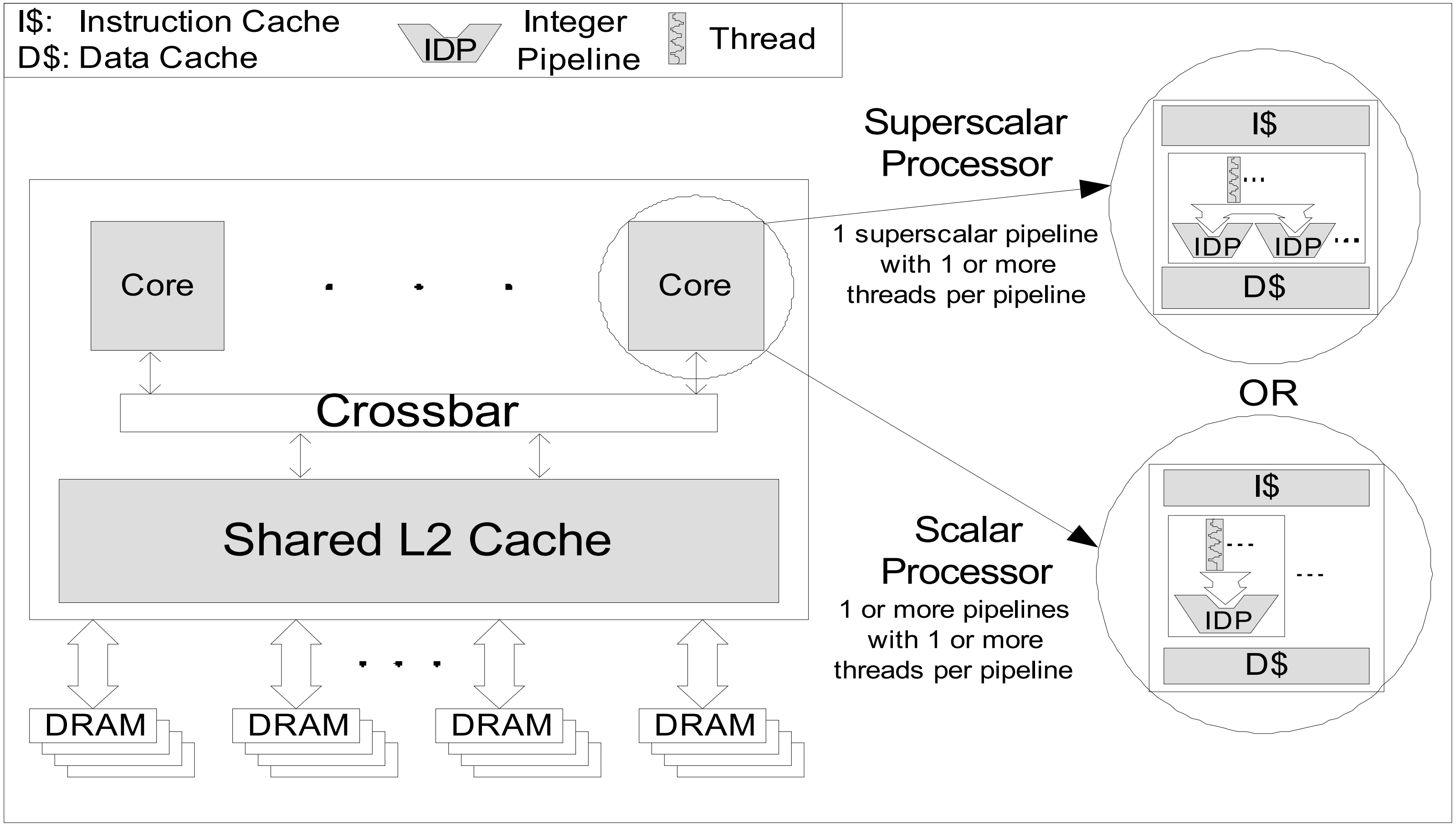
- Tradeoff exists between large number of simple cores and small number of complex cores
  - > Complex cores focus on ILP for higher single thread performance
  - > ILP scarce in commercial workloads
  - > Simple cores can deliver more TLP
- Need to trade off area devoted to processor cores, L2 and L3 caches, and system-on-a-chip
- Balance performance and power in all subsystems: processor, caches, memory and I/O

Maximizing CMP Throughput with Mediocre[1] Cores

- J. Davis, J. Laudon, K. Olukotun PACT '05 paper
- Examined several UltraSPARC II, III, IV, and T1 designs, accounting for differing technologies
- Constructed an area model based on this exploration
- Assumed a fixed-area large die (400 mm^2), and accounted for pads, pins, and routing overhead
- Looked at performance for a broad swath of scalar and in-order superscalar processor core designs

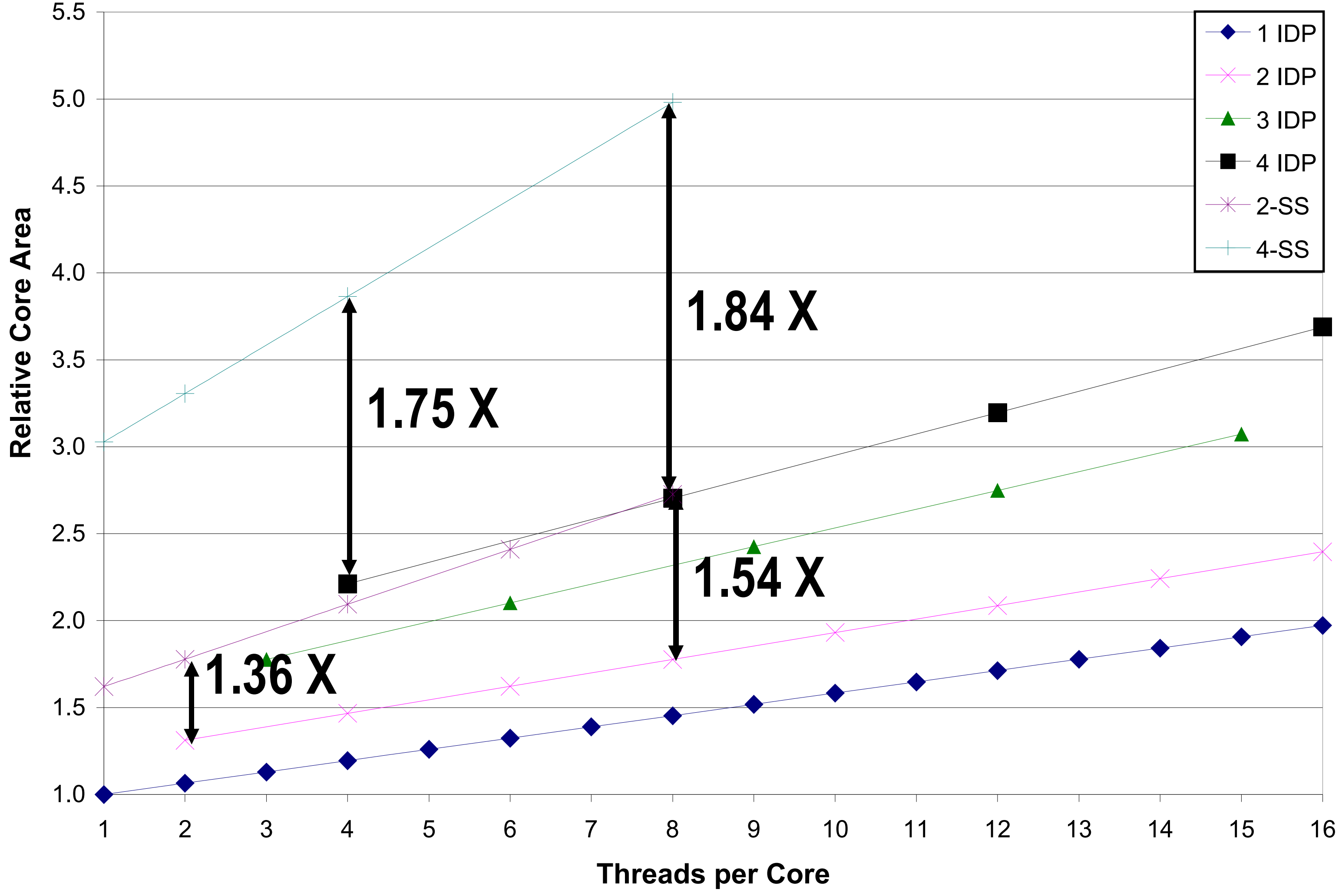
[1] Mediocre: *adj.* ordinary; of moderate quality, value, ability, or performance

CMP Design Space

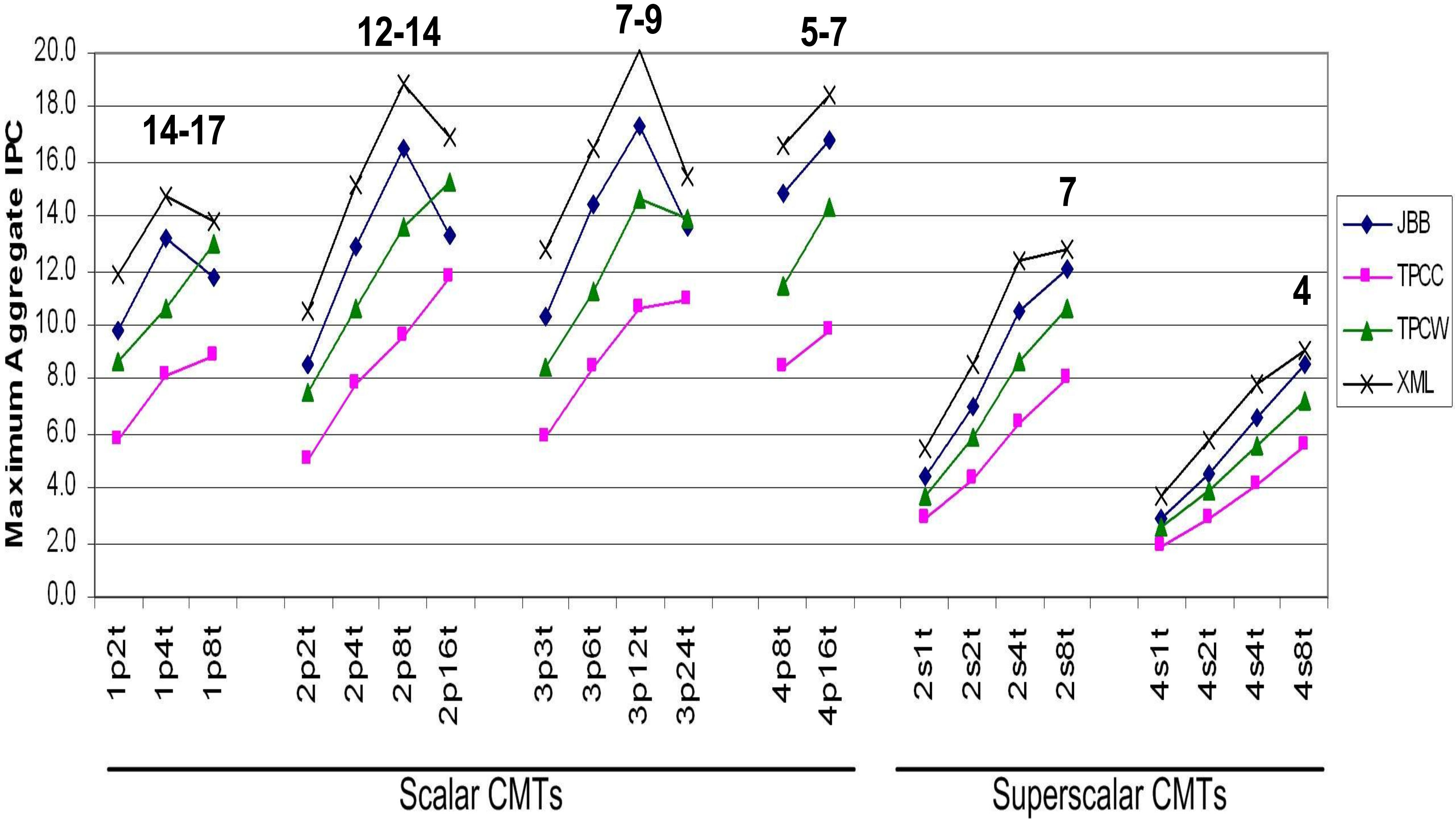


- Large simulation space: 13k runs/benchmark/technology (pruned)
- Fixed die size: number of cores in CMP depends on the core size

Scalar vs. Superscalar Core Area

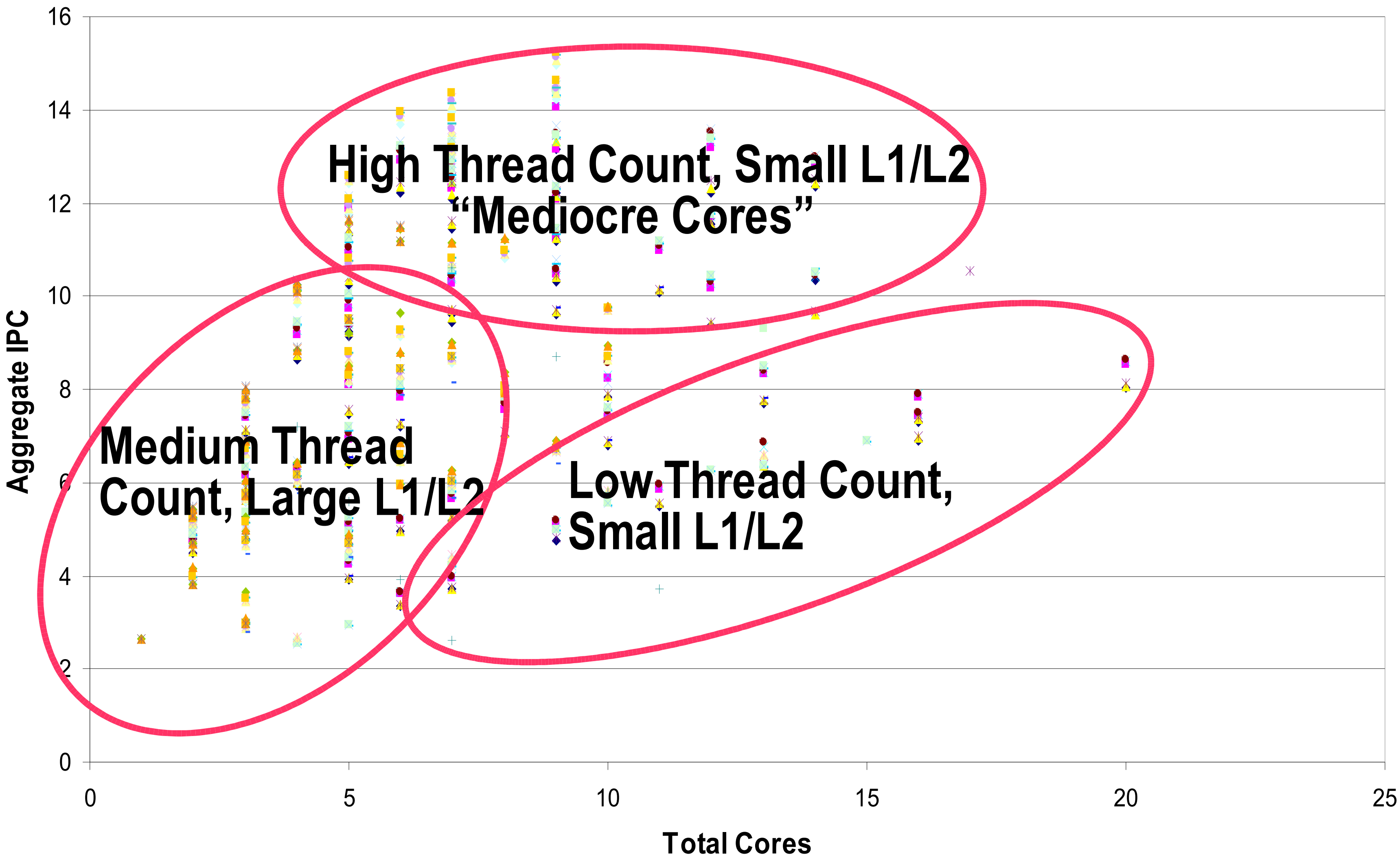


Trading complexity, cores and caches



Source: J. Davis, J. Laudon, K. Olukotun, *Maximizing CMP Throughput with Medicore Cores*, PACT '05

The Scalar CMP Design Space

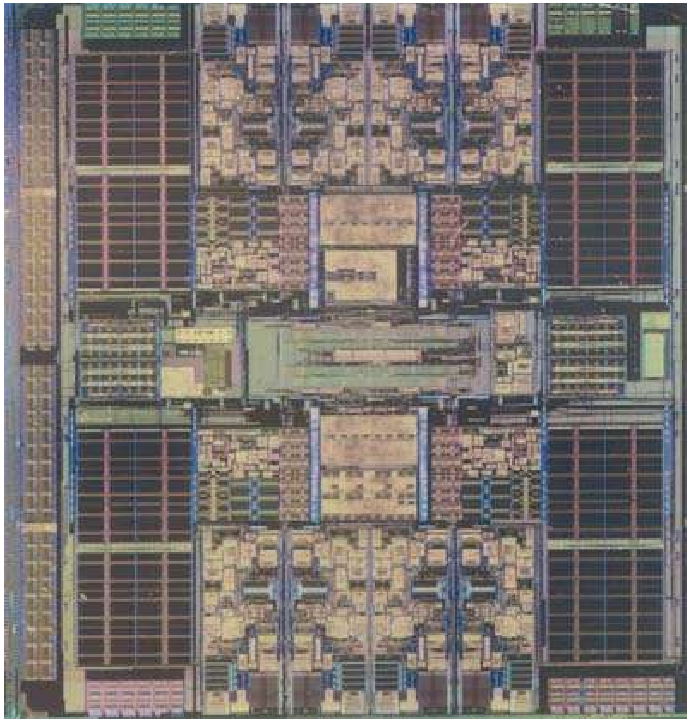


Limitations of Simple Cores

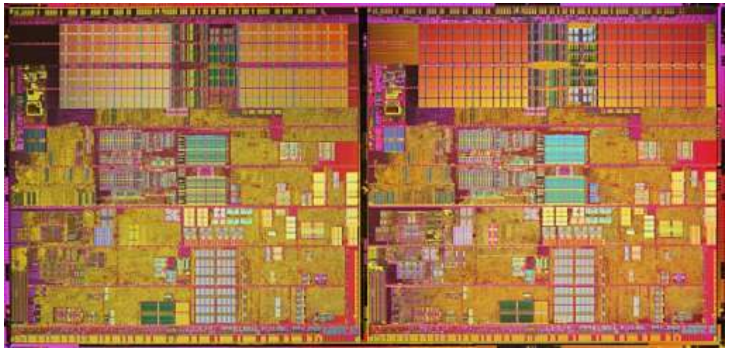
- Lower SPEC CPU2000 ratio performance
  - > Not representative of most single-thread code
  - > Abstraction increases frequency of branching and indirection
  - > Most applications wait on network, disk, memory; rarely execution units
- Large number of threads per chip
  - > 32 for UltraSPARC T1, 100+ threads soon
  - > Is software ready for this many threads?
  - > Many commercial applications scale well
  - > Workload consolidation

Simple core comparison

UltraSPARC T1
379 mm^2



Pentium Extreme Edition
206 mm^2



Comparison Disclaimers

- Different design teams and design environments
- Chips fabricated in 90 nm by TI and Intel
- UltraSPARC T1: designed from ground up as a CMP
- Pentium Extreme Edition: two cores bolted together
- Apples to watermelons comparison, but still interesting

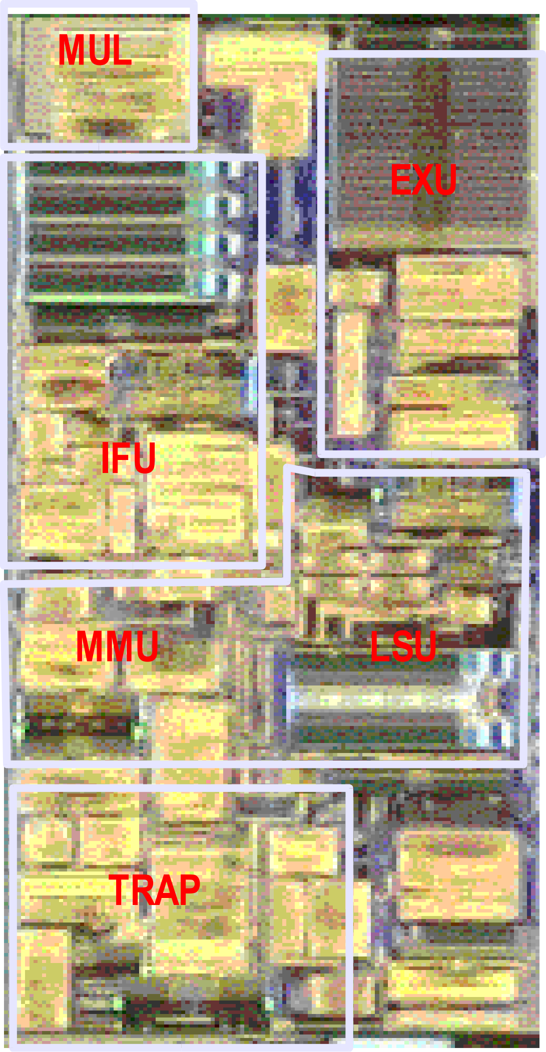
Pentium EE- US T1 Bandwidth Comparison

Feature	***Pentium Extreme Edition***	***UltraSPARC T1***
Clock Speed	3.2 Ghz	1.2 Ghz
Pipeline Depth	31 stages	6 stages
Power	130 W (@ 1.3 V)	72W (@ 1.3V)
Die Size	206 mm^2	379 mm^2
Transistor Count	230 million	279 million
Number of cores	2	8
Number of threads	4	32
L1 caches	12 kuop Instruction/16 kB Data	16 kB Instruction/8 kB Data
Load-to-use latency	1.1 ns	2.5 ns
L2 cache	Two copies of 1 MB, 8-way associative	3 MB, 12-way associative
L2 unloaded latency	7.5 ns	19 ns
L2 bandwidth	~180 GB/s	76.8 GB/s
Memory unloaded latency	80 ns	90 ns
Memory bandwidth	6.4 GB/s	25.6 GB/s

Sharing Saves Area & Ups Utilization

- Hardware threads within a processor core share:
  - > Pipeline and execution units
  - > L1 caches, TLBs and load/store port
- Processor cores within a CMP share:
  - > L2 and L3 caches
  - > Memory and I/O ports
- Increases utilization
  - > Multiple threads fill pipeline and overlap memory stalls with computation
  - > Multiple cores increase load on L2 and L3 caches and memory

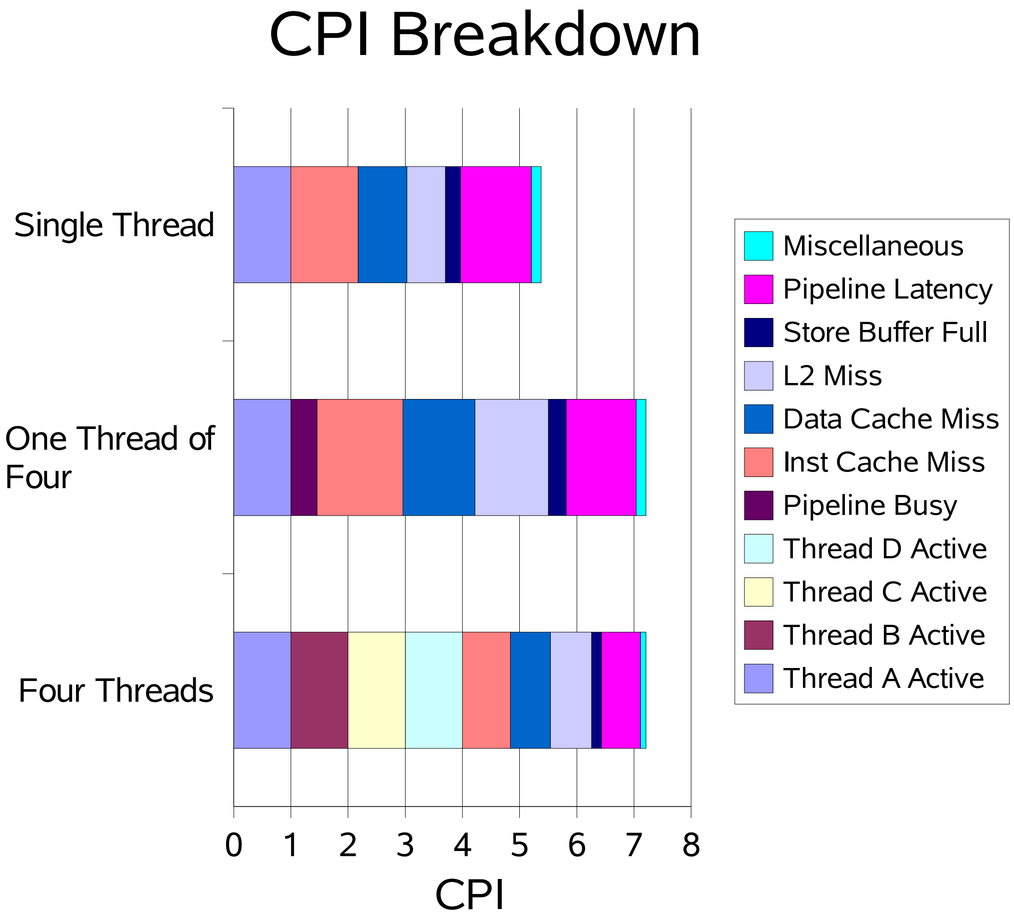
Sharing to save area



- UltraSPARC T1
- Four threads per core
- Multithreading increases:
  - > Register file
  - > Trap unit
  - > Instruction buffers and fetch resources
  - > Store queues and miss buffers
- 20% area increase in core excluding cryptography unit

Sharing to increase utilization

UltraSPARC T1 Database App Utilization



- Application run with both 8 and 32 threads
- With 32 threads, pipeline and memory contention slow each thread by 34%
- However, increased utilization leads to 3x speedup with four threads

Keeping the threads fed

- Dedicated resources for thread memory requests
  - Private store buffers and miss buffers
- Large, banked, and highly-associative L2 cache
  - Multiple banks for sufficient bandwidth
  - Increased size and associativity to hold the working sets of multiple threads
- Direct connection to high-bandwidth memory
  - Fallout from shared L2 will be larger than from a private L2
  - But increase in L2 miss rate will be much smaller than increase in number of threads

Keeping the threads cool

- Sharing of resources increases unit utilization and thus leads to increase in power
- Cores must be power efficient
  - > Minimal speculation – high-payoff only
  - > Moderate pipeline depth and frequency
- Extensive mechanisms for power management
  - > Voltage and frequency control
  - > Clock gating and unit shutdown
  - > Leakage power control
  - > Minimizing cache and memory power

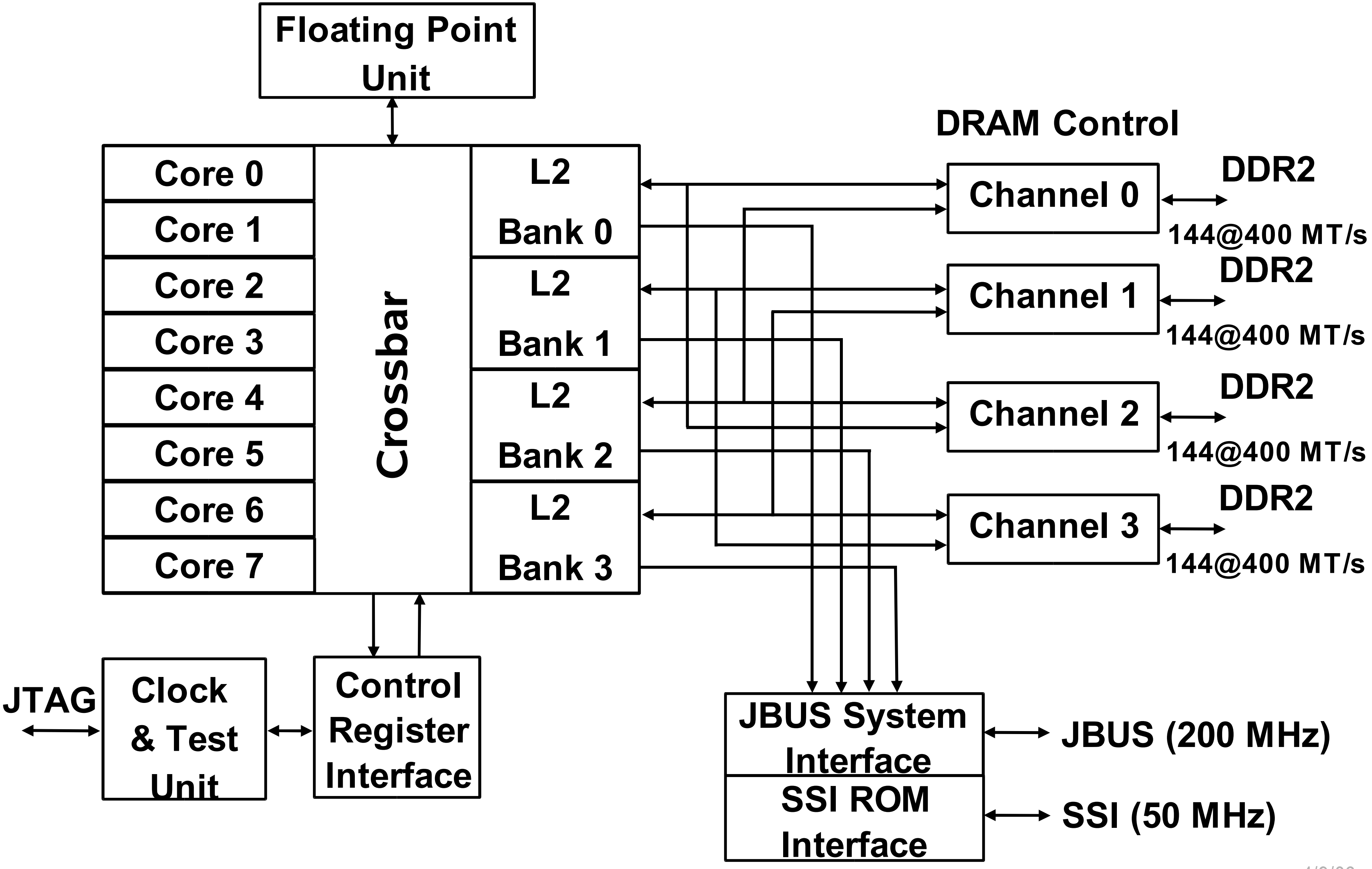
Outline

- Server design issues
  - > Application demands
  - > System requirements
- Building a better server-oriented CMP
  - > Maximizing thread count
  - > Keeping the threads fed
  - > Keeping the threads cool
- UltraSPARC T1 (Niagara)
  - > Micro-architecture
  - > Performance
  - > Power

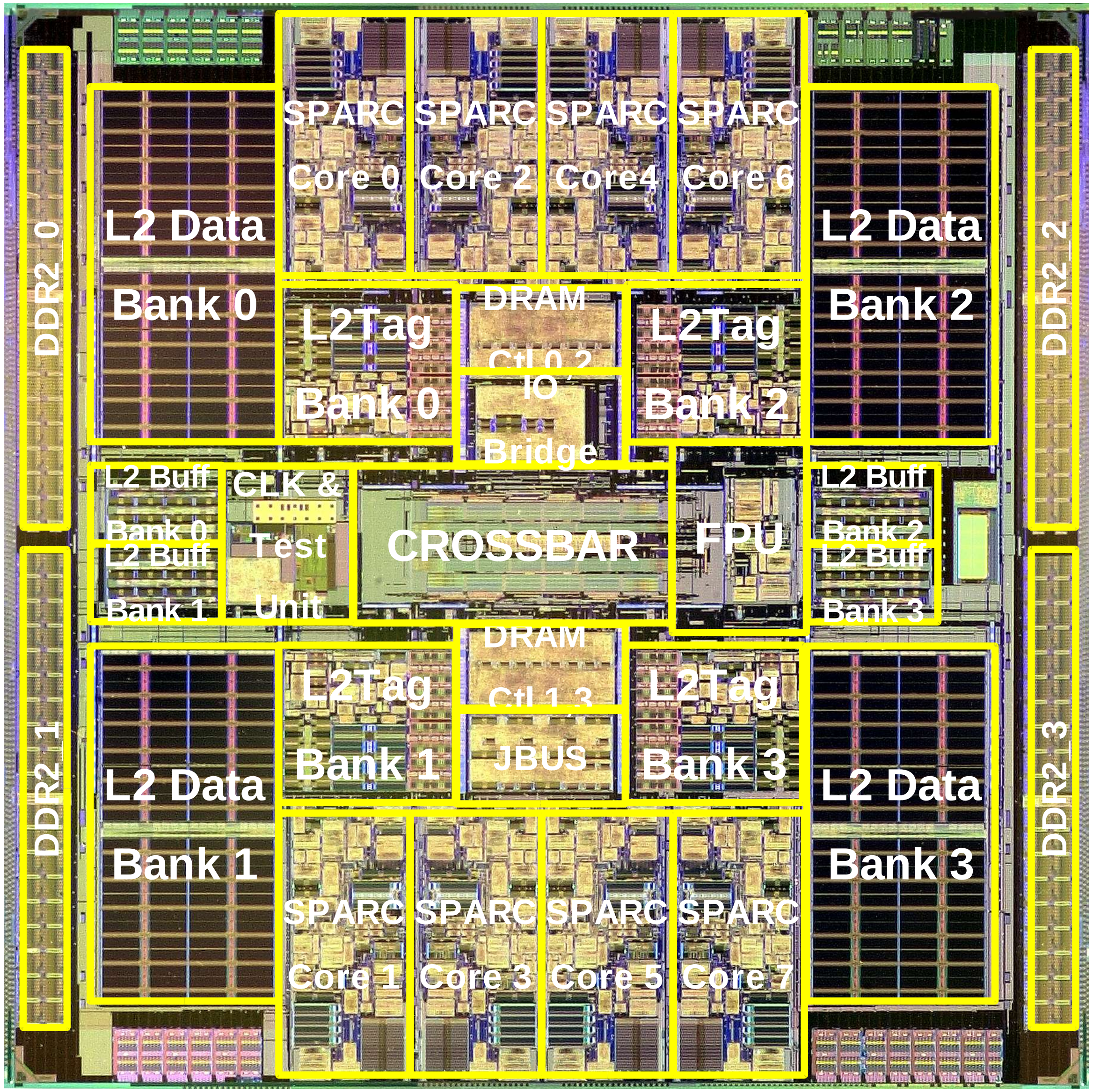
UltraSPARC T1 Overview

- TLP-focused CMP for servers
  - 32 threads to hide memory and pipeline stalls
- Extensive sharing
  - Four threads share each processor core
  - Eight processor cores share a single L2 cache
- High-bandwidth cache and memory subsystem
  - Banked and highly-associative L2 cache
  - Direct connection to DDR II memory
- Performance/Watt as a design metric

UltraSPARC T1 Block Diagram



UltraSPARC T1 Micrograph



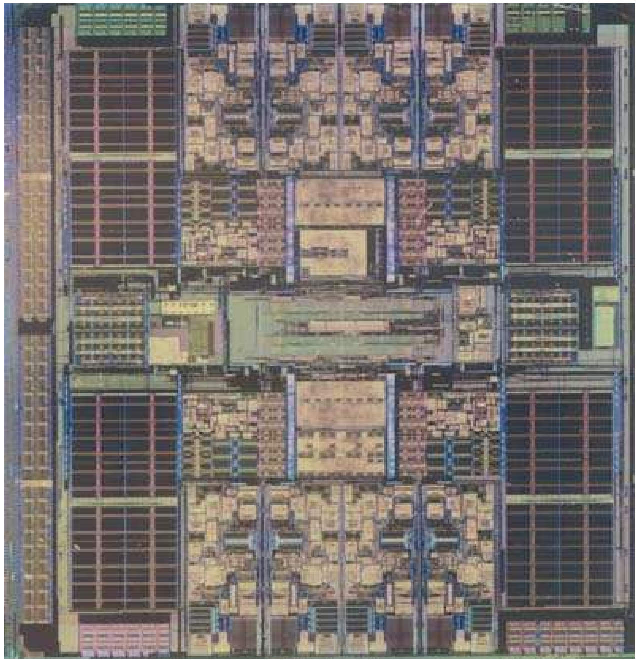
Features:

- 8 64-bit Multithreaded SPARC Cores
- Shared 3 MB, 12-way 64B line writeback L2 Cache
- 16 KB, 4-way 32B line ICache per Core
- 8 KB, 4-way 16B line write-through DCache per Core
- 4 144-bit DDR-2 channels
- 3.2 GB/sec JBUS I/O

Technology:

- TI's 90nm CMOS Process
- 9LM Cu Interconnect
- 63 Watts @ 1.2GHz/1.2V
- Die Size: 379mm^2
- 279M Transistors
- Flip-chip ceramic LGA

UltraSPARC T1 Floorplanning

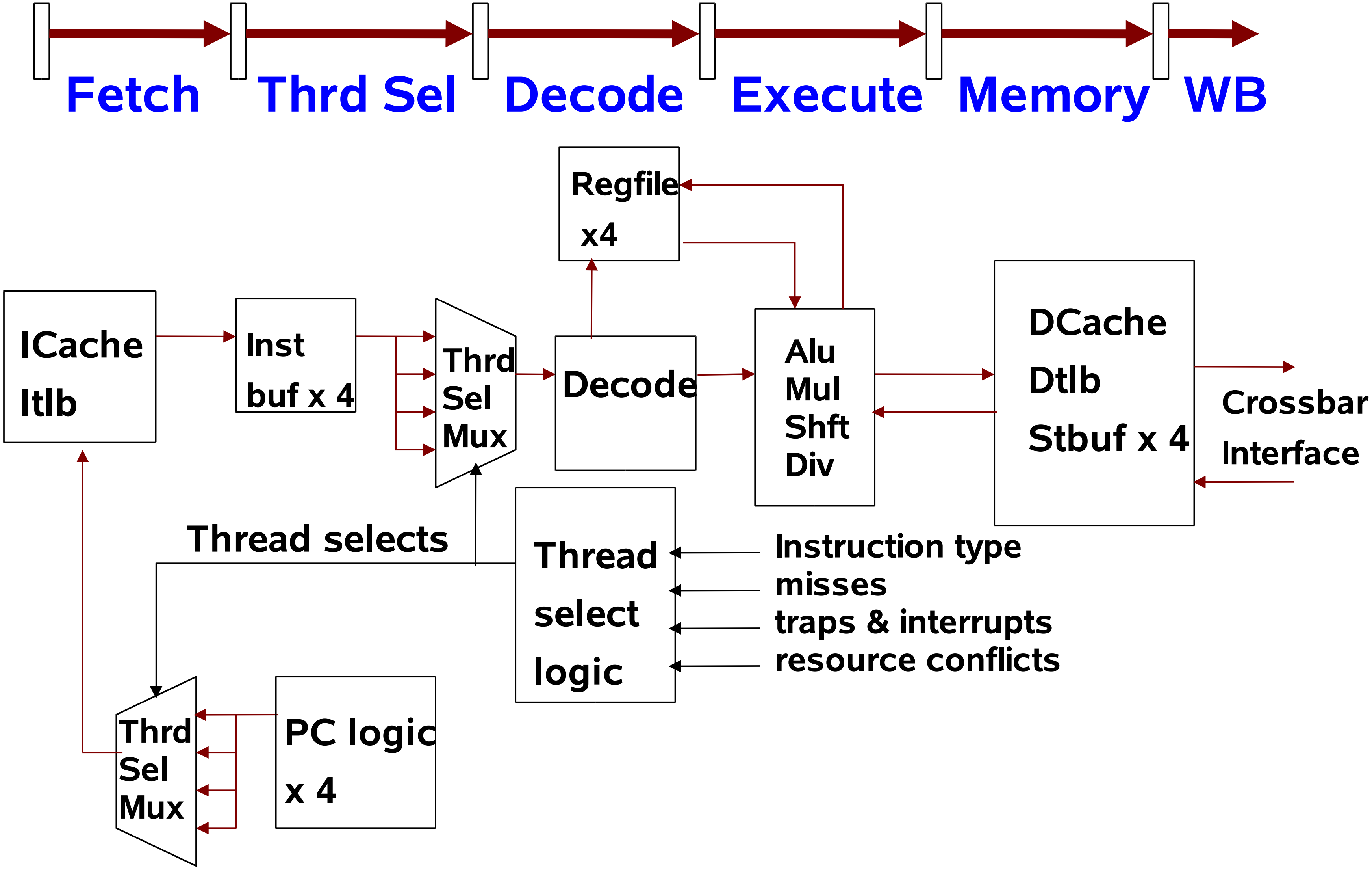


- Modular design for “step and repeat”
- Main issue is that all cores want to be close to all the L2 cache banks
  - > Crossbar and L2 tags located in the center
  - > Processor cores on the top and bottom
  - > L2 data on the left and right
  - > Memory controllers and SOC fill in the holes

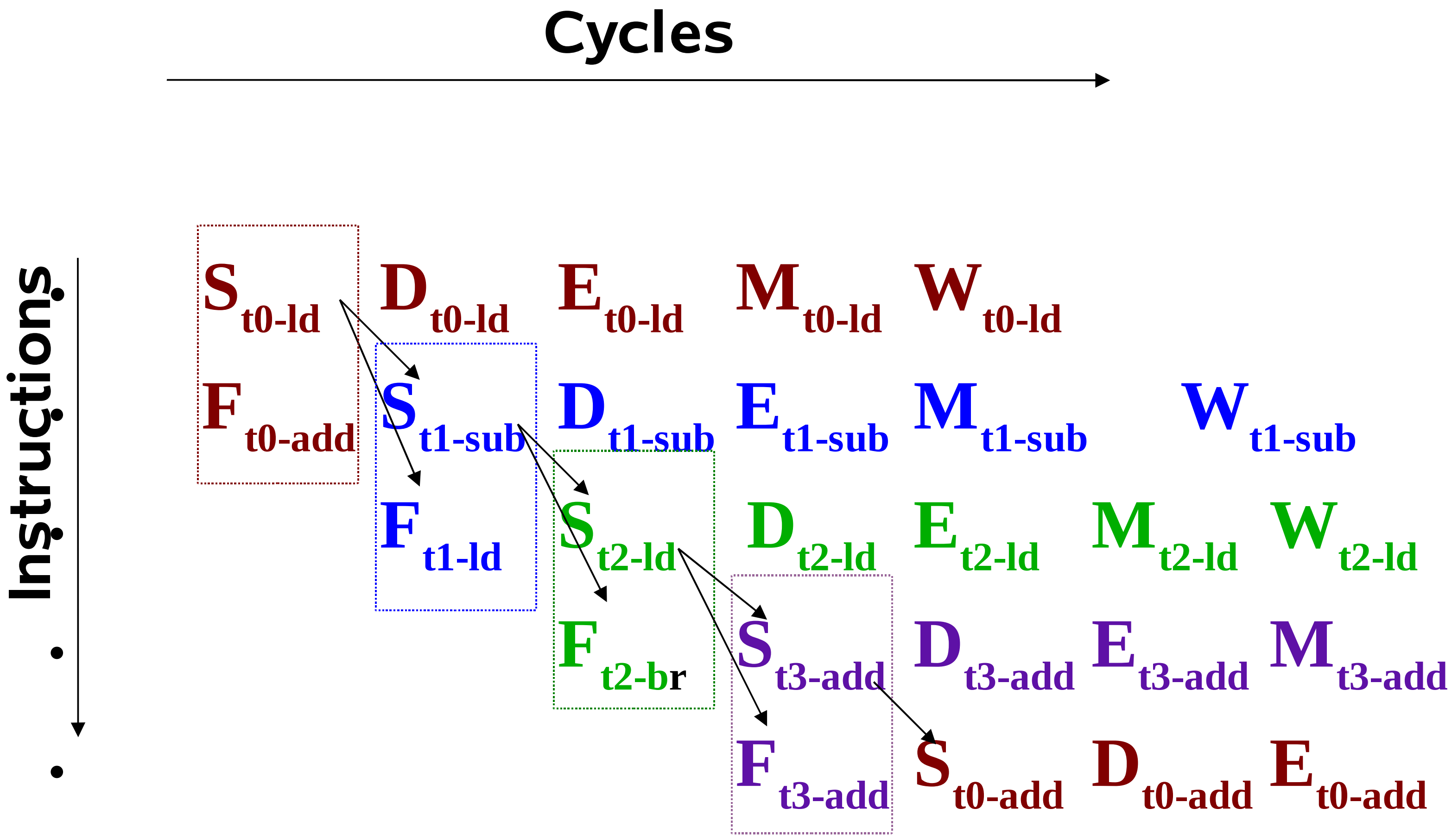
Maximing Thread Count on US-T1

- Power-efficient, simple cores
  - Six stage pipeline, almost no speculation
  - 1.2 GHz operation
  - Four threads per core
    - Shared: pipeline, L1 caches, TLB, L2 interface
    - Dedicated: register and other architectural state, instruction buffers, 8-entry store buffers
  - Pipeline switches between available threads every cycle (interleaved/vertical multithreading)
  - Cryptography acceleration unit per core

UltraSPARC T1 Pipeline

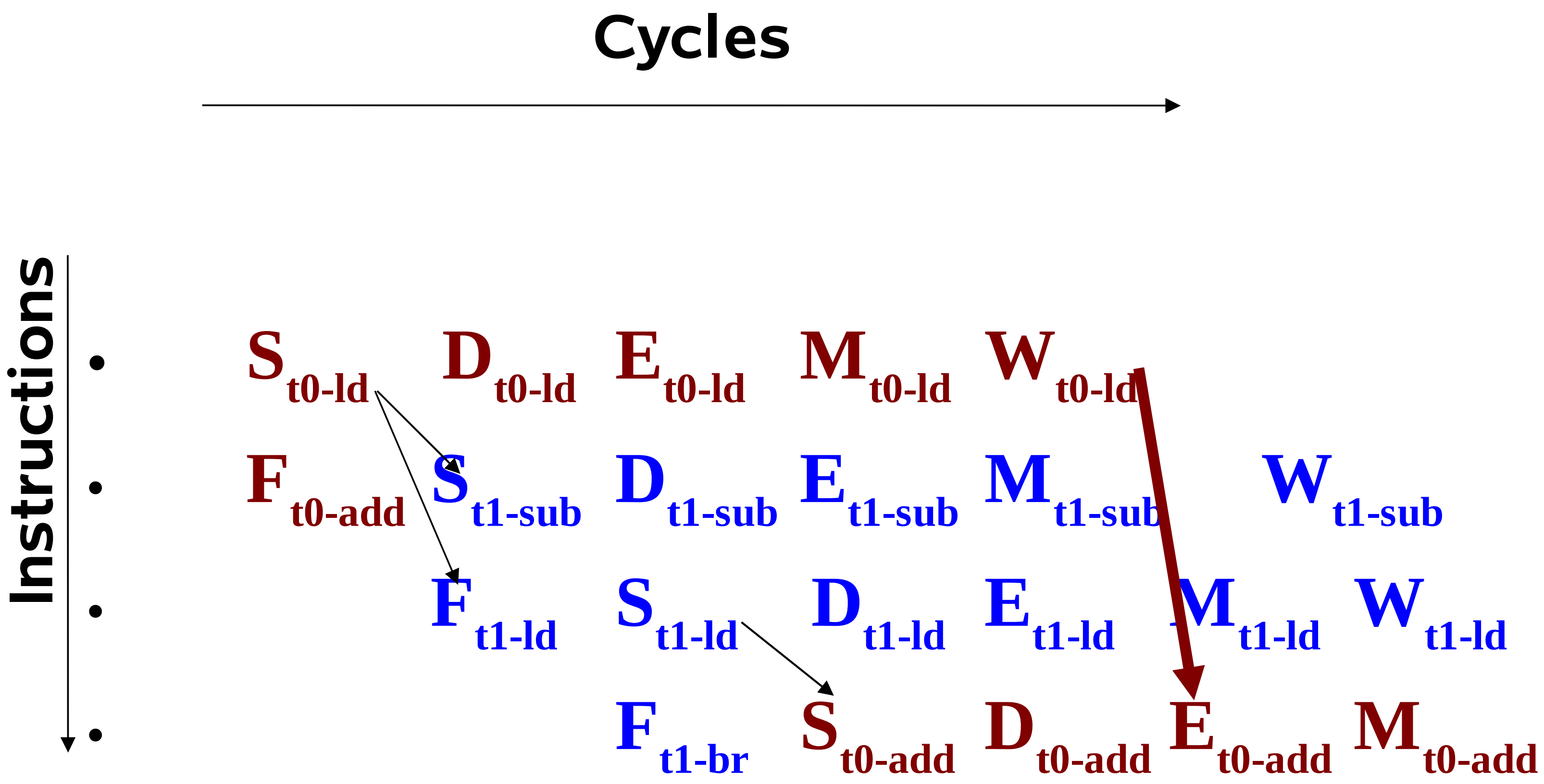


Thread Selection: All Threads Ready



S$_{\text{t0-ld}}$	D$_{\text{t0-ld}}$	E$_{\text{t0-ld}}$	M$_{\text{t0-ld}}$	W$_{\text{t0-ld}}$		
F$_{\text{t0-add}}$	S$_{\text{t1-sub}}$	D$_{\text{t1-sub}}$	E$_{\text{t1-sub}}$	M$_{\text{t1-sub}}$	W$_{\text{t1-sub}}$	
	F$_{\text{t1-ld}}$	S$_{\text{t2-ld}}$	D$_{\text{t2-ld}}$	E$_{\text{t2-ld}}$	M$_{\text{t2-ld}}$	W$_{\text{t2-ld}}$
		F$_{\text{t2-br}}$	S$_{\text{t3-add}}$	D$_{\text{t3-add}}$	E$_{\text{t3-add}}$	M$_{\text{t3-add}}$
			F$_{\text{t3-add}}$	S$_{\text{t0-add}}$	D$_{\text{t0-add}}$	E$_{\text{t0-add}}$

Thread Selection: Two Threads Ready



$S_{t0\text{-}ld}$	$D_{t0\text{-}ld}$	$E_{t0\text{-}ld}$	$M_{t0\text{-}ld}$	$W_{t0\text{-}ld}$		
$F_{t0\text{-}add}$	$S_{t1\text{-}sub}$	$D_{t1\text{-}sub}$	$E_{t1\text{-}sub}$	$M_{t1\text{-}sub}$	$W_{t1\text{-}sub}$	
	$F_{t1\text{-}ld}$	$S_{t1\text{-}ld}$	$D_{t1\text{-}ld}$	$E_{t1\text{-}ld}$	$M_{t1\text{-}ld}$	$W_{t1\text{-}ld}$
		$F_{t1\text{-}br}$	$S_{t0\text{-}add}$	$D_{t0\text{-}add}$	$E_{t0\text{-}add}$	$M_{t0\text{-}add}$

Thread '0' is speculatively switched in before cache hit information is available, in time for the 'load' to bypass data to the 'add'

Feeding the UltraSPARC T1 Threads

- Shared L2 cache
  - > 3 MB, writeback, 12-way associative, 64B lines
  - > 4 banks, interleaved on cache line boundary
  - > Handles multiple outstanding misses per bank
  - > MESI coherence – L2 cache orders all requests
  - > Maintains directory and inclusion of L1 caches
- Direct connection to memory
  - > Four 144-bit wide (128+16) DDR II interfaces
  - > Supports up to 128 GB of memory
  - > 25.6 GB/s memory bandwidth

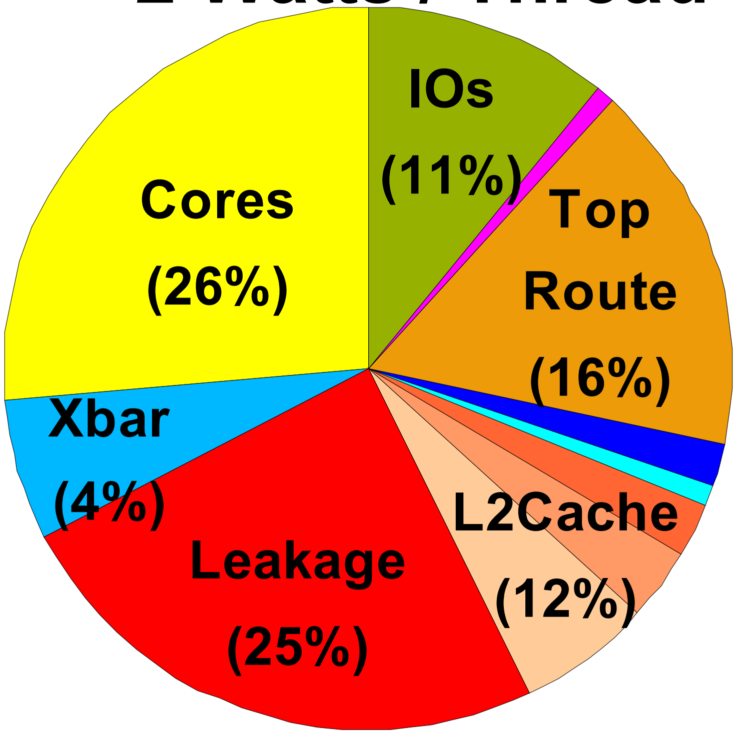
Keeping the US-T1 Threads Cool

- Power efficient cores
  - > 1.2 GHz 6-stage single-issue pipeline
- Features to keep peak power close to average
  - > Ability to suspend issue from any thread
  - > Limit on number of outstanding memory requests
- Extensive clock gating
  - > Coarse-grained (unit shutdown, partial activation)
  - > Fine-grained (selective gating within datapaths)
- Static design for most of chip
- 63 Watts typical power at 1.2V and 1.2 GHz

UltraSPARC T1 Power Breakdown

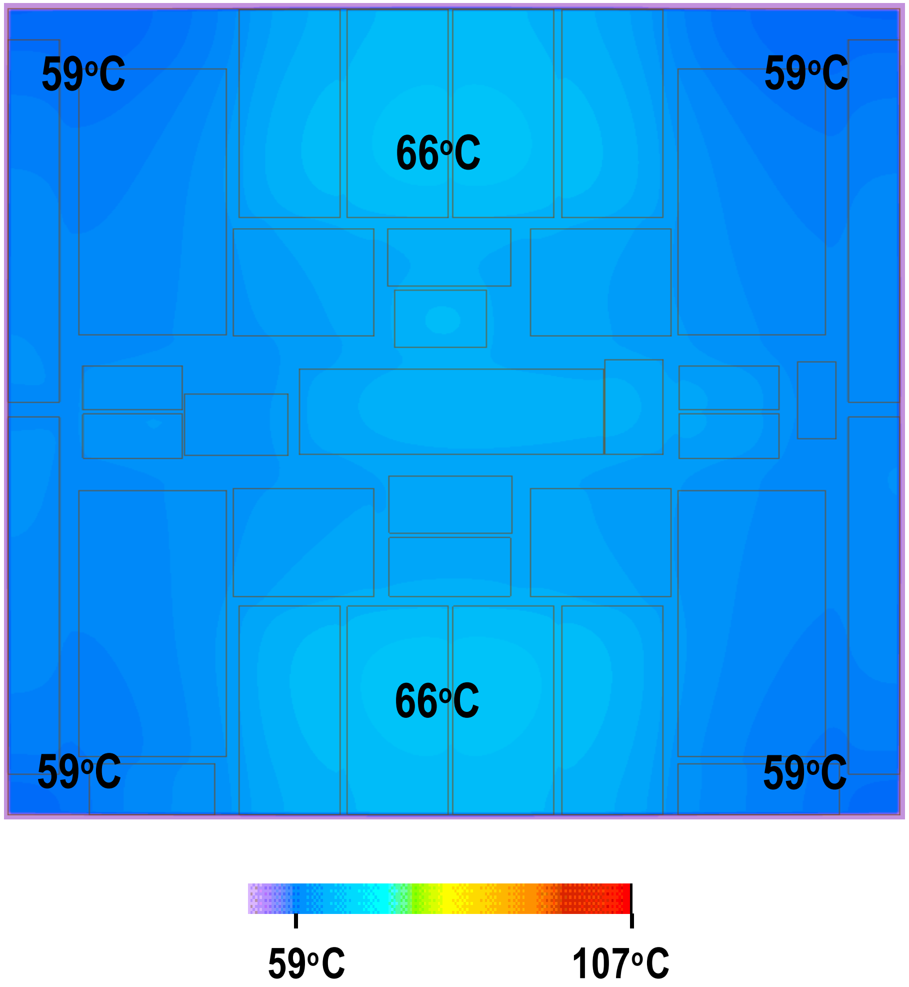
63W @ 1.2Ghz / 1.2V

< 2 Watts / Thread



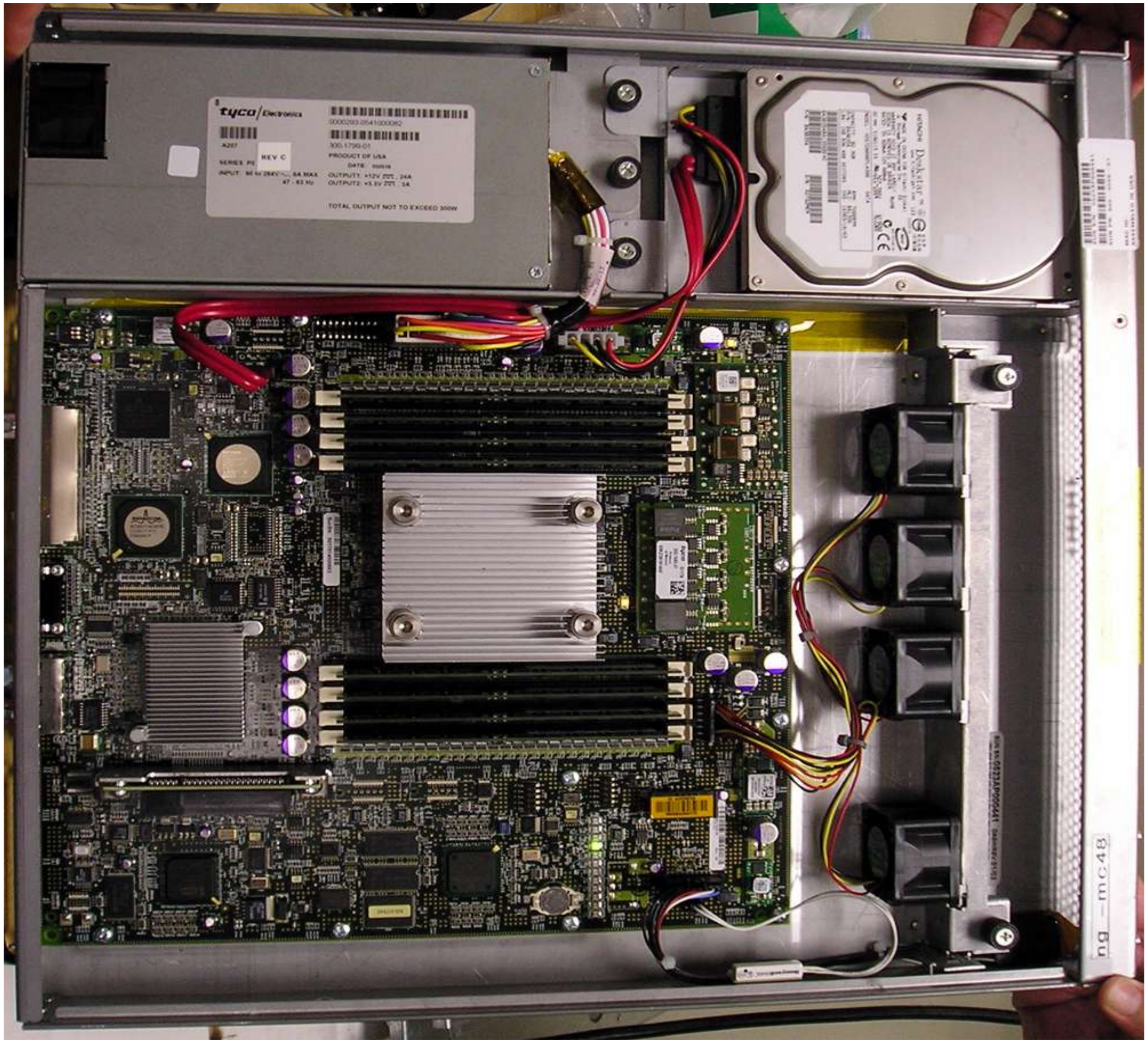
- Fully static design
- Fine granularity clock gating for datapaths (30% flops disabled)
- Lower 1.5 P/N width ratio for library cells
- Interconnect wire classes optimized for power x delay
- SRAM activation control

Advantages of CoolThreads™

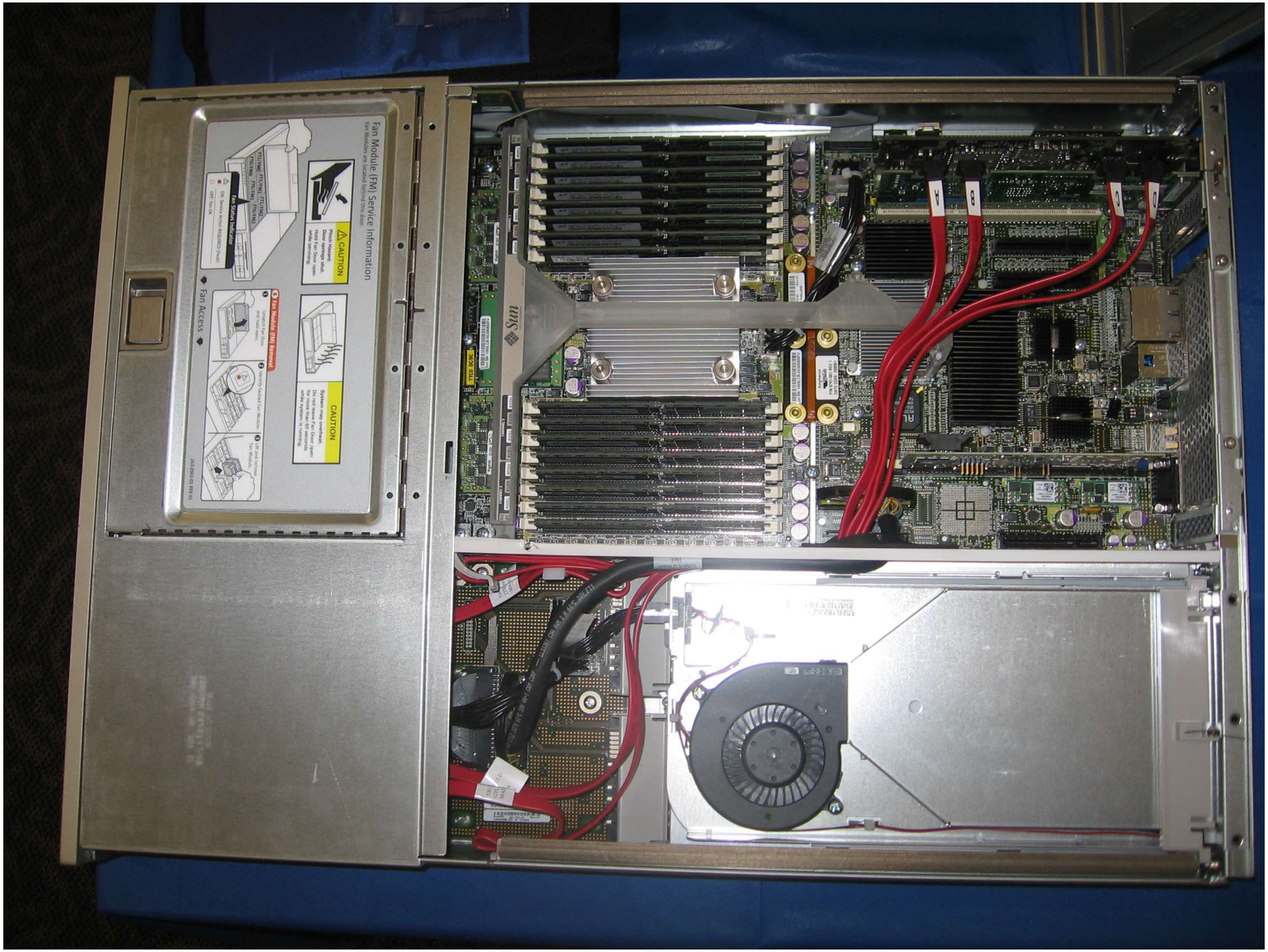


- No need for exotic cooling technologies
- Improved reliability from lower and more uniform junction temperatures
- Improved performance/reliability tradeoff in design

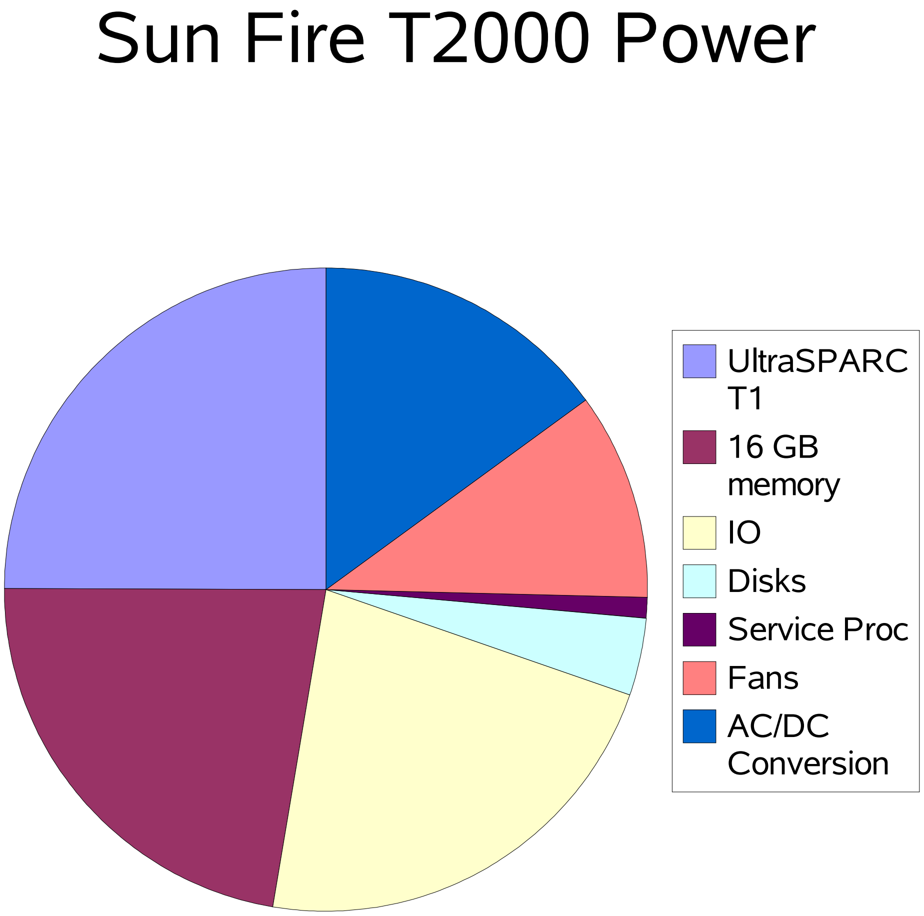
UltraSPARC T1 System (T1000)



UltraSPARC T1 System (T2000)



T2000 Power Breakdown

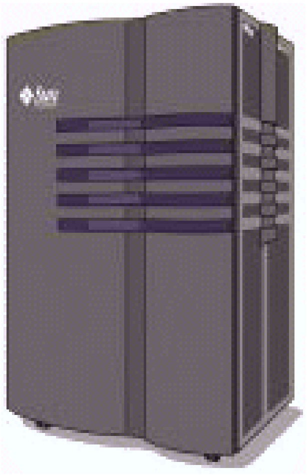


- 271W running SPECJBB 2000
- Power breakdown
  - 25% processor
  - 22% memory
  - 22% I/O
  - 4% disk
  - 1% service processor
  - 10% fans
  - 15% AC/DC conversion

UltraSPARC T1 Performance

Sun Fire T2000		
CPU		UltraSPARC T1
Sockets		1
Height		2U
SpecWeb 2005	Performance	14001
	Power	330 W
	Perf/Watt	42.4
SpecJBB 2005	Performance	63378 BOPS
	Power	298 W
	Perf/Watt	212.7

E10K



1997

32 x US2

77.4 ft³

2000 lbs

13,456 W

52,000 BTUs/hr

T2000

2005

1 x US T1

0.85 ft³

37 lbs

~300 W

1,364 BTUs/hr

Future Trends

- Improved thread performance
  - > Deeper pipelines
  - > More high-payoff speculation
- Increased number of threads per core
- More of the system components will move on-chip
- Continued focus on delivering high performance/Watt and performance/Watt/Volume (SWaP)

Conclusions

- Server TCO will soon be dominated by power
- Server CMPs need to be designed from ground up to improve performance/Watt
    - Simple MT cores => threads ↑=> performance ↑
    - Lower frequency and less speculation => power ↓
    - Must provide enough bandwidth to keep threads fed
- UltraSPARC T1 employs these principles to deliver outstanding performance and performance/Watt on a broad range of commercial workloads

Legal Disclosures

- SPECweb2005 Sun Fire T2000 (8 cores, 1 chip) 14001 SPECweb2005
- SPEC, SPECweb reg tm of Standard Performance Evaluation Corporation
- Sun Fire T2000 results submitted to SPEC Dec 6th 2005
- Sun Fire T2000 server power consumption taken from measurements made during the benchmark run
- SPECjbb2005 Sun Fire T2000 Server (1 chip, 8 cores, 1-way) 63,378 bops
- SPEC, SPECjbb reg tm of Standard Performance Evaluation Corporation
- Sun Fire T2000 results submitted to SPEC Dec 6th 2005
- Sun Fire T2000 server power consumption taken from measurements made during the benchmark run