Lecture 8
Thread Level Parallelism (2)

EEC 171 Parallel Architectures

John Owens
UC Davis

Credits

- © John Owens / UC Davis 2007–9.
- Thanks to many sources for slide material: Computer Organization and Design (Patterson & Hennessy) © 2005, Computer Architecture (Hennessy & Patterson) © 2007, Inside the Machine (Jon Stokes) © 2007, © Dan Connors / University of Colorado 2007, © Kathy Yelick / UCB 2007, © Wen-Mei Hwu/David Kirk, University of Illinois 2007, © David Patterson / UCB 2003–7, © John Lazzaro / UCB 2006, © Mary Jane Irwin / Penn State 2005, © John Kubiatowicz / UCB 2002, © Krste Asinovic/Arvind / MIT 2002, © Morgan Kaufmann Publishers 1998.

Outline

- Simultaneous Multithreading
- Google's Architecture
- Sun T1 (Niagara)

Gaming news article

Eve Online sets gaming supercomputer cluster record

Company aiming for 50,000 online players on a single shard
Matt Chapman, vnunet.com, 11 Sep 2006

Massively multiplayer online game Eve Online has set a new record for the largest supercomputer cluster in the games industry.

The Eve Online server cluster manages over 150 million database transactions per day on a 64-bit hardware architecture from IBM.

The database servers use solid state disks instead of traditional hard drives to handle more than 400,000 random I/Os per second.

CCP Games, which created Eve Online, recently set a world record by hosting 30,000 concurrent users on a single server shard. The company now hopes to increase that to at least 50,000 users. ...

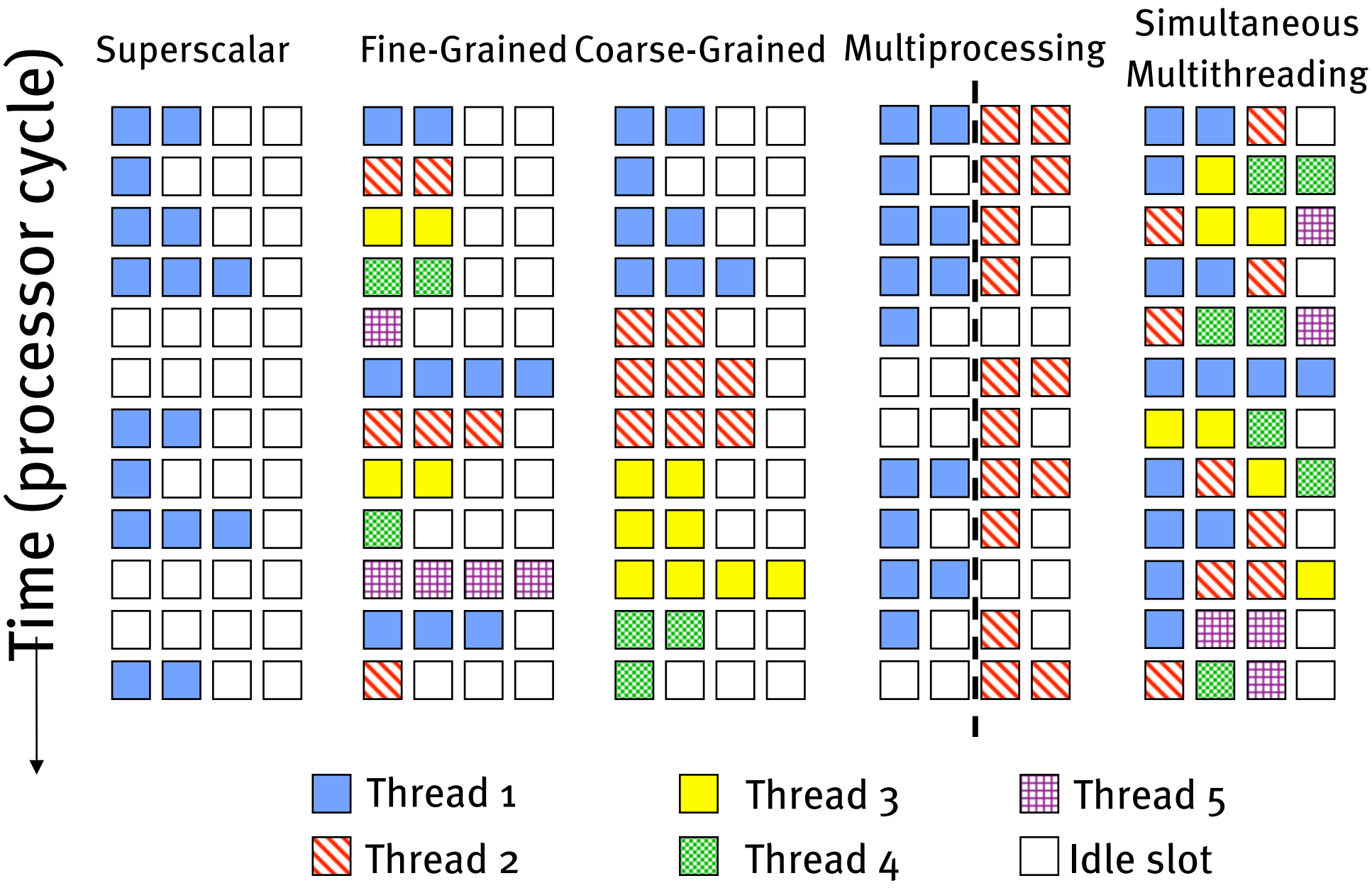
The upgraded server cluster features dual-processor 64-bit AMD Opteron-based IBM BladeCenter LS20 blade servers, as well additional enhancements to the cluster's internet backbone.

http://www.itweek.co.uk/vnunet/news/2163979/eve-online-scores-largest

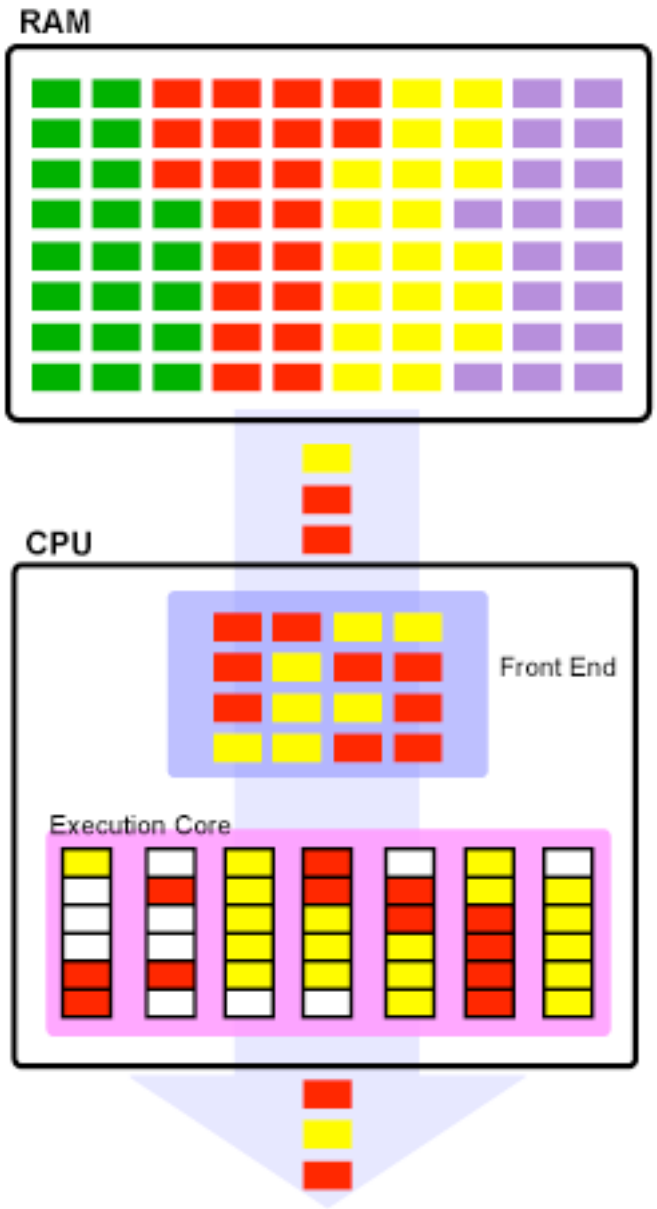
pthreads Example

```
#include ‹stdio.h›
#include ‹stdlib.h›
#include ‹time.h›
#include ‹pthread.h›

static void wait(void) {
  time_t start_time = time(NULL);
  while (time(NULL) == start_time) {
    /* do nothing except chew CPU slices for
up to one second */
  }
}

static void *thread_func(void *vptr_args) {
  for (int i = 0; i ‹ 20; i++) {
    fputs("  b\n", stderr);
    wait();
  }
  return NULL;
}

int main(void) {
  int i;
  pthread_t thread;

  if (pthread_create(&thread, NULL,
thread_func, NULL) != 0) {
    return EXIT_FAILURE;
  }
  for (i = 0; i ‹ 20; i++) {
    fputs("a\n", stdout);
    wait();
  }
  if (pthread_join(thread, NULL) != 0) {
    return EXIT_FAILURE;
  }
  return EXIT_SUCCESS;
}
```

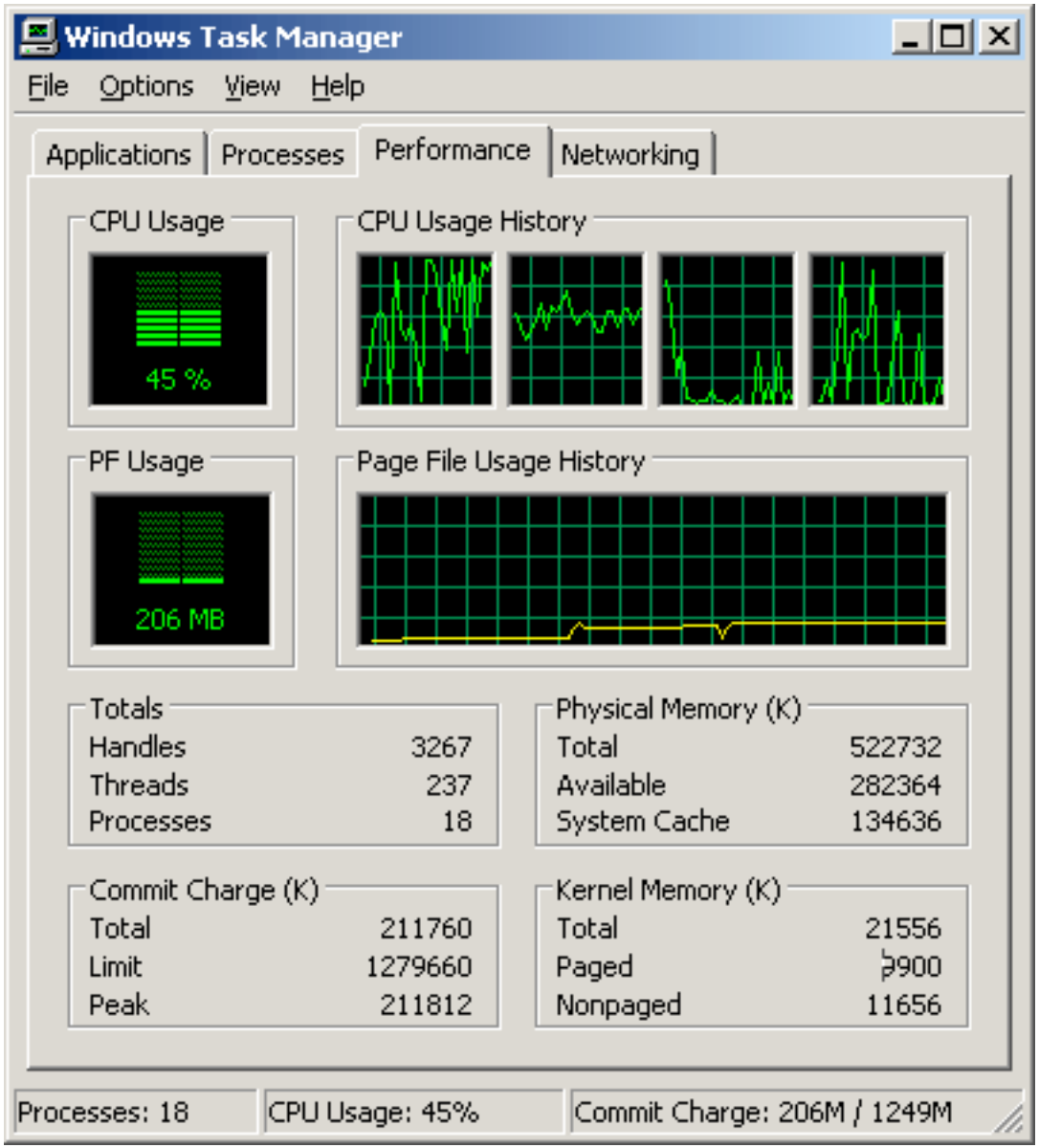
Multithreaded Categories



Simultaneous multithreading (SMT)



“Hyperthreading”



http://www.2cpu.com/Hardware/ht_analysis/images/taskmanager.html

Multithreaded Execution

- When do we switch between threads?
  - Alternate instruction per thread (fine grain)
  - When a thread is stalled, perhaps for a cache miss, another thread can be executed (coarse grain)

Fine-Grained Multithreading

- Switches between threads on each instruction, causing the execution of multiple threads to be interleaved
- Usually done in a round-robin fashion, skipping any stalled threads
- CPU must be able to switch threads every clock
- Advantage is it can hide both short and long stalls, since instructions from other threads executed when one thread stalls
- Disadvantage is it slows down execution of individual threads, since a thread ready to execute without stalls will be delayed by instructions from other threads
- Used on Sun’s Niagara (will see later)

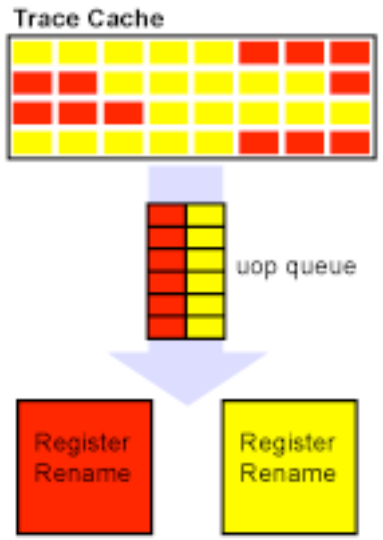
Coarse-Grained Multithreading

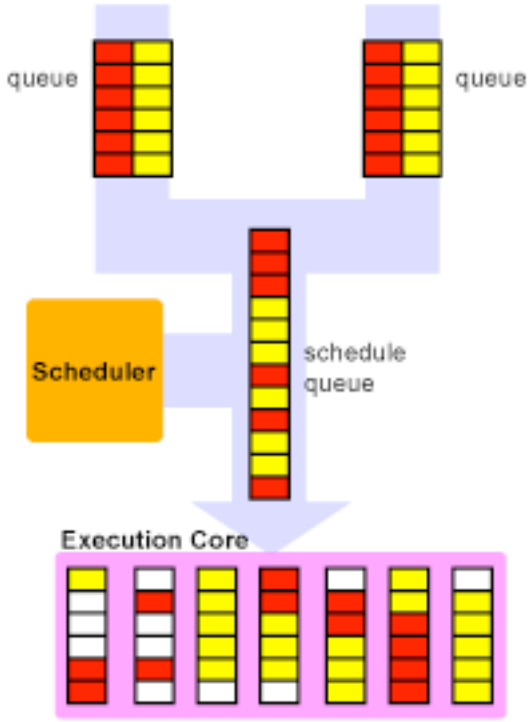
- Switches threads only on costly stalls, such as L2 cache misses
- Advantages
  - Relieves need to have very fast thread-switching
  - Doesn't slow down thread, since instructions from other threads issued only when the thread encounters a costly stall
- Disadvantage is hard to overcome throughput losses from shorter stalls, due to pipeline start-up costs
  - Since CPU issues instructions from 1 thread, when a stall occurs, the pipeline must be emptied or frozen
  - New thread must fill pipeline before instructions can complete
- Because of this start-up overhead, coarse-grained multithreading is better for reducing penalty of high cost stalls, where pipeline refill << stall time
- Used in IBM AS/400

P4Xeon Microarchitecture

- Replicated
  - Register renaming logic
  - Instruction pointer, other architectural registers
  - ITLB
  - Return stack predictor
- Partitioned
  - Reorder buffers
  - Load/store buffers
  - Various queues: scheduling, uop, etc.
- Shared
  - Caches (trace, L1/L2/L3)
  - Microarchitectural registers
  - Execution units
- If configured as single-threaded, all resources go to one thread

Partitioning: Static vs. Dynamic





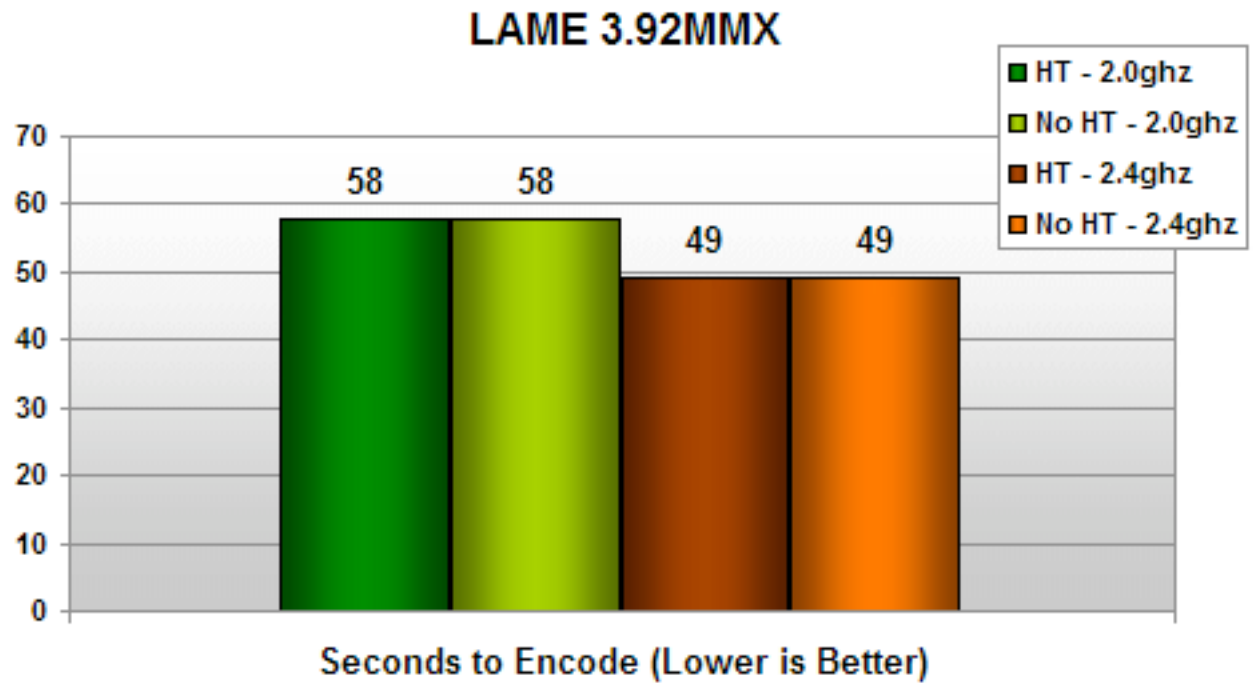
Design Challenges in SMT

- Since SMT makes sense only with fine-grained implementation, impact of fine-grained scheduling on single thread performance?
  - A preferred thread approach sacrifices neither throughput nor single-thread performance?
  - Unfortunately, with a preferred thread, the processor is likely to sacrifice some throughput, when preferred thread stalls
- Larger register file needed to hold multiple contexts
- Not affecting clock cycle time, especially in
  - Instruction issue—more candidate instructions need to be considered
  - Instruction completion—choosing which instructions to commit may be challenging
- Ensuring that cache and TLB conflicts generated by SMT do not degrade performance

Problems with SMT

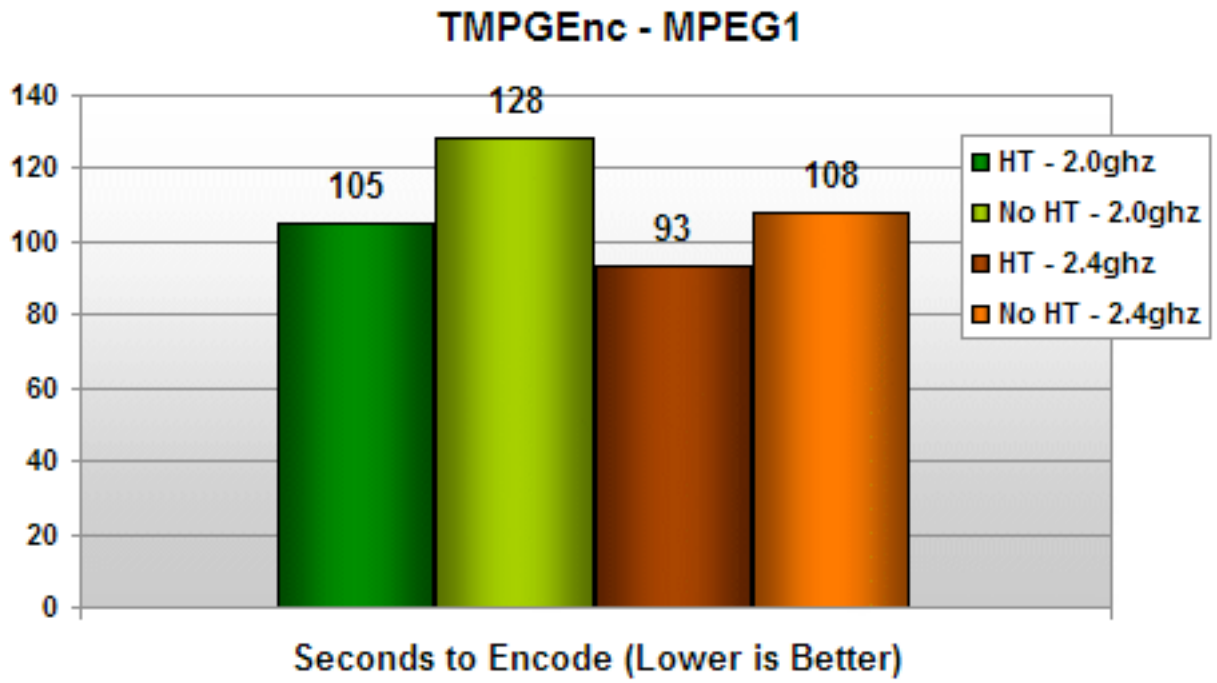
- One thread monopolizes resources
  - Example: One thread ties up FP unit with long-latency instruction, other thread tied up in scheduler
- Cache effects
  - Caches are unaware of SMT—can't make warring threads cooperate
  - If both warring threads access different memory and have cache conflicts, constant swapping

Hyperthreading Neutral!



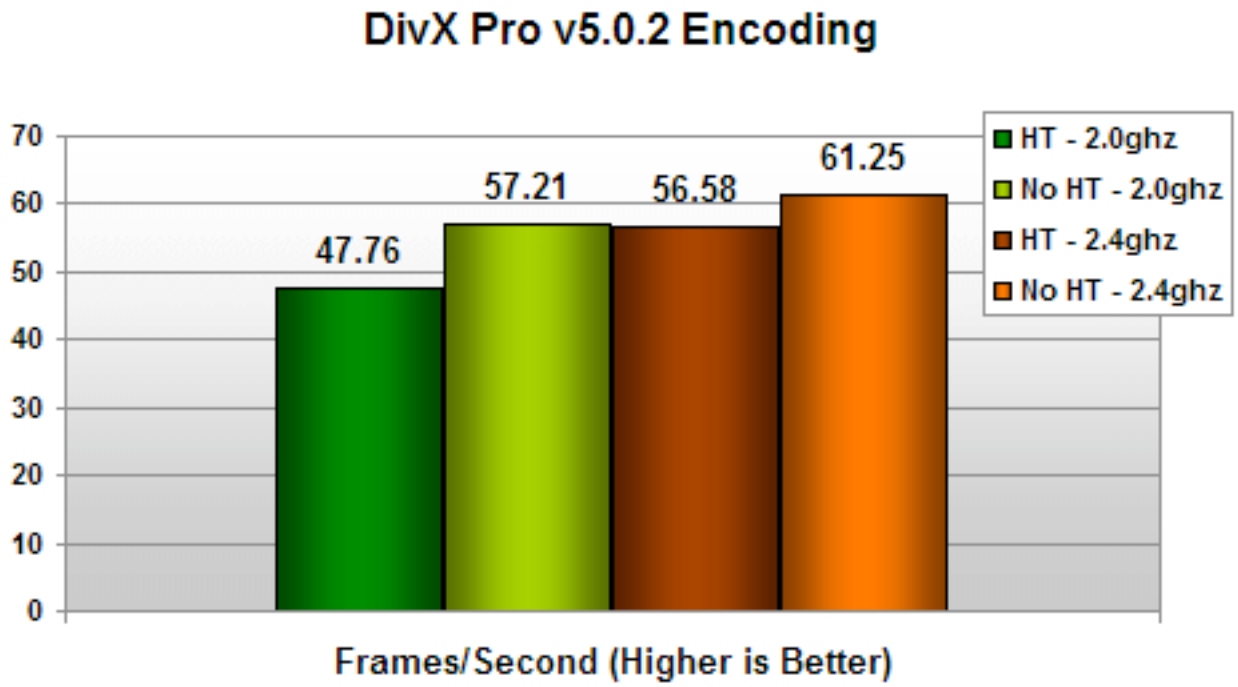
http://www.2cpu.com/articles/43_1.html

Hyperthreading Good!



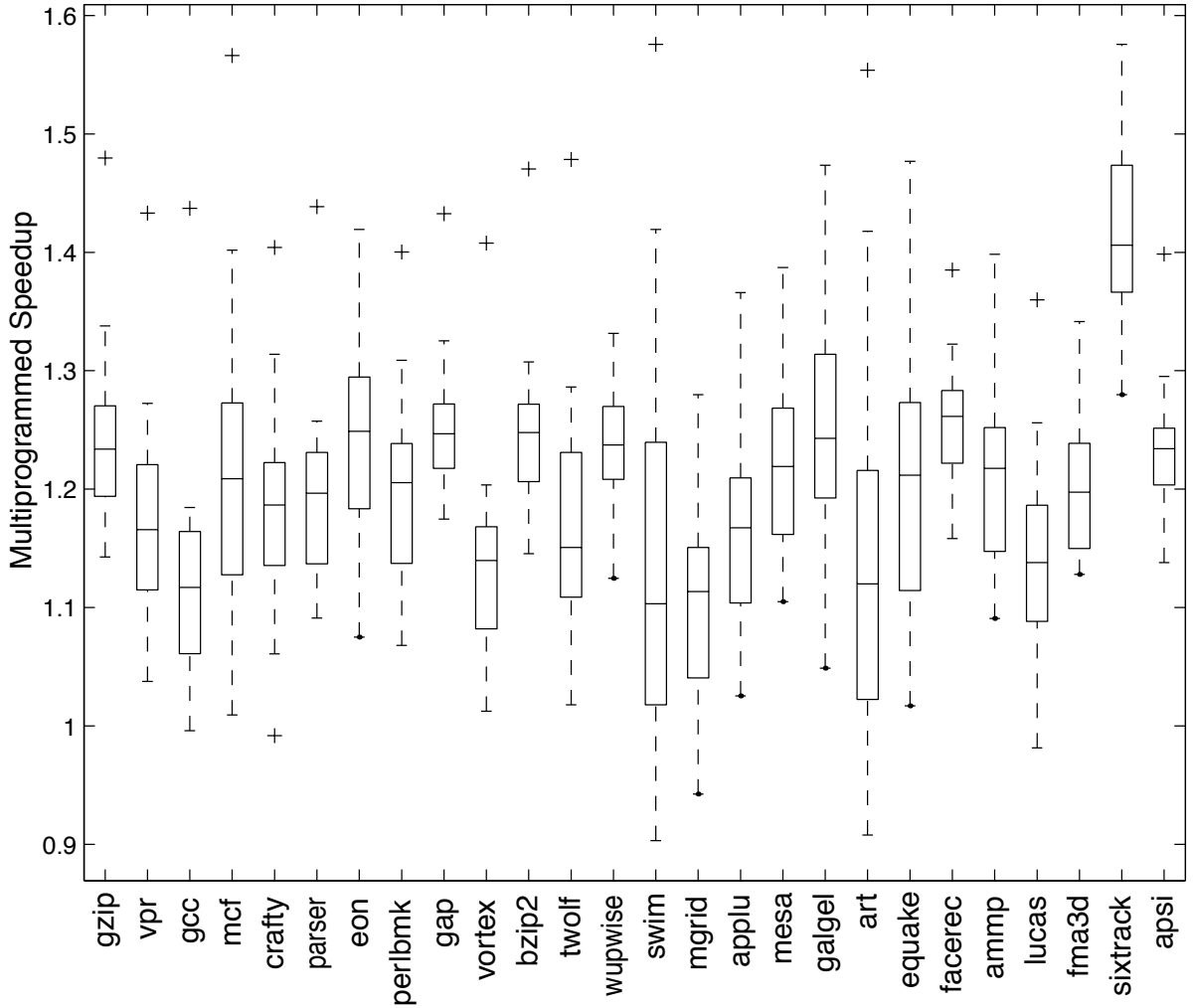
http://www.2cpu.com/articles/43_1.html

Hyperthreading Bad!



http://www.2cpu.com/articles/43_1.html

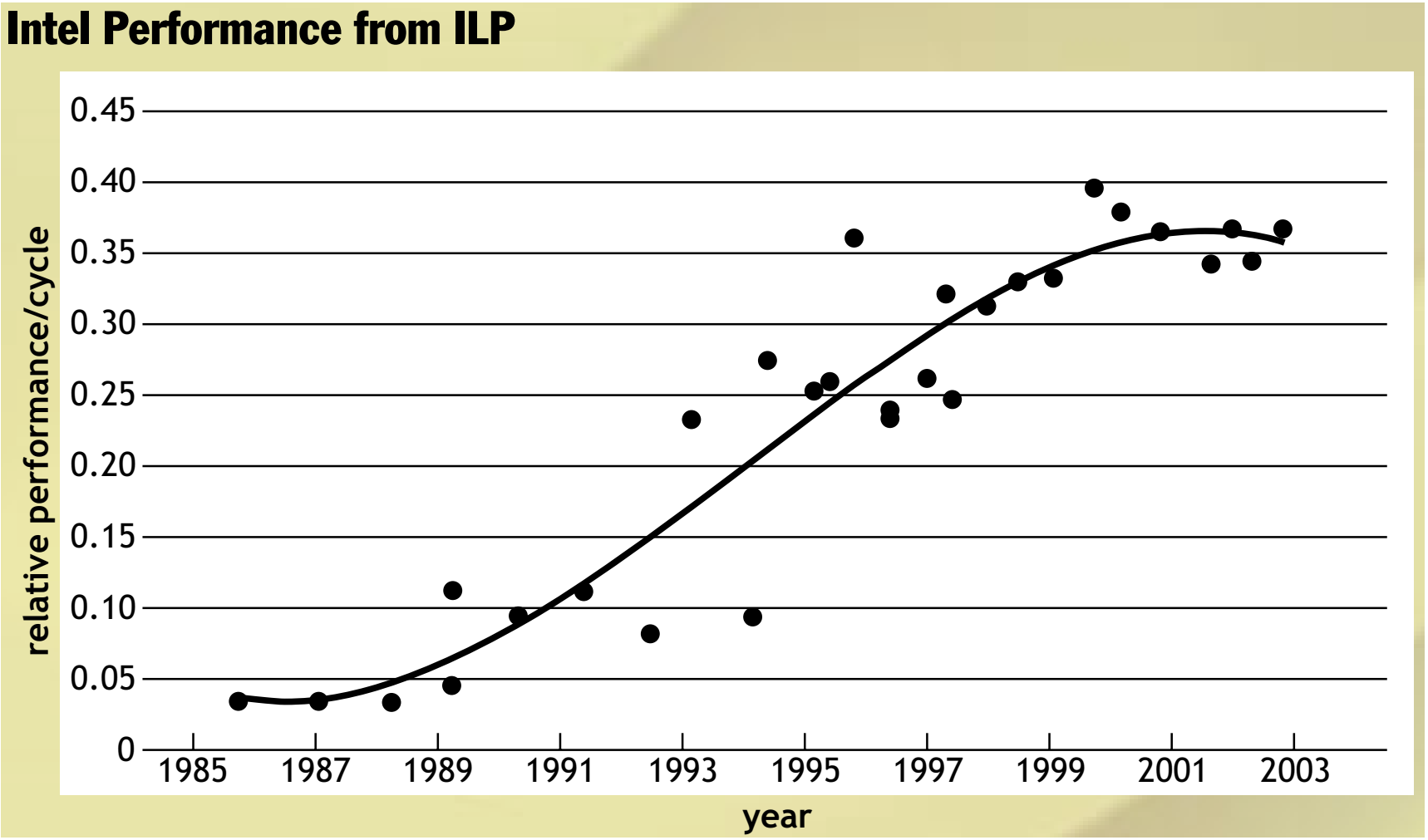
SPEC vs. SPEC (PACT '03)



- Avg. multithreaded speedup 1.20 (range 0.90–1.58)

"Initial Observations of the Simultaneous Multithreading Pentium 4 Processor", Nathan Tuck and Dean M. Tullsen (PACT '03)

ILP reaching limits



- Olukotun and Hammond, “The Future of Microprocessors”, *ACM Queue*, Sept. 2005

Olukotun's view

- "With the exhaustion of essentially all performance gains that can be achieved for 'free' with technologies such as superscalar dispatch and pipelining, we are now entering an era where programmers must switch to more parallel programming models in order to exploit multi-processors effectively, if they desire improved single-program performance."

Olukotun (pt. 2)

- “This is because there are only three real ‘dimensions’ to processor performance increases beyond Moore’s law: clock frequency, superscalar instruction issue, and multiprocessing. We have pushed the first two to their logical limits and must now embrace multiprocessing, even if it means that programmers will be forced to change to a parallel programming model to achieve the highest possible performance.”

Google's Architecture

- "Web Search for a Planet: The Google Cluster Architecture"
  - Luiz André Barroso, Jeffrey Dean, Urs Hölzle, Google
- Reliability in software not in hardware
- 2003: 15k commodity PCs
- July 2006 (estimate): 450k commodity PCs
  - $2M/month for electricity

Goal: Price/performance

- “We purchase the CPU generation that currently gives the best performance per unit price, not the CPUs that give the best absolute performance.”
  - Google rack: 40–80 x86 servers
    - “Our focus on price/performance favors servers that resemble mid-range desktop PCs in terms of their components, except for the choice of large disk drives.”
    - 4-processor motherboards: better perf, but not better price/perf
    - SCSI disks: better perf and reliability, but not better price/perf
  - Depreciation costs: $7700/month; power costs: $1500/month
    - Low-power systems must have equivalent performance

Google power density

- Mid-range server, dual 1.4 GHz Pentium III: 90 watts
  - 55 W for 2 CPUs
  - 10 W for disk drive
  - 25 W for DRAM/motherboard
    - so 120 W of AC power (75% efficient)
- Rack fits in 25 ft^2
  - 400 W/ft^2; high end processors 700 W/ft^2
  - Typical data center: 70–150 W/ft^2
  - Cooling is a big issue

Google Workload (1 GHz P3)

Table 1. Instruction-level measurements on the index server.

Characteristic	Value
Cycles per instruction	1.1
Ratios (percentage)	
Branch mispredict	5.0
Level 1 instruction miss*	0.4
Level 1 data miss*	0.7
Level 2 miss*	0.3
Instruction TLB miss*	0.04
Data TLB miss*	0.7

* Cache and TLB ratios are per instructions retired.

Details of workload

- “Moderately high CPI” (P3 can issue 3 instrs/cycle)
  - “Significant number of difficult-to-predict branches”
  - Same workload on P4 has “nearly twice the CPI and approximately the same branch prediction performance”
- “In essence, there isn’t that much exploitable instruction-level parallelism in the workload.”
- “Our measurements suggest that the level of aggressive out-of-order, speculative execution present in modern processors is already beyond the point of diminishing performance returns for such programs.”

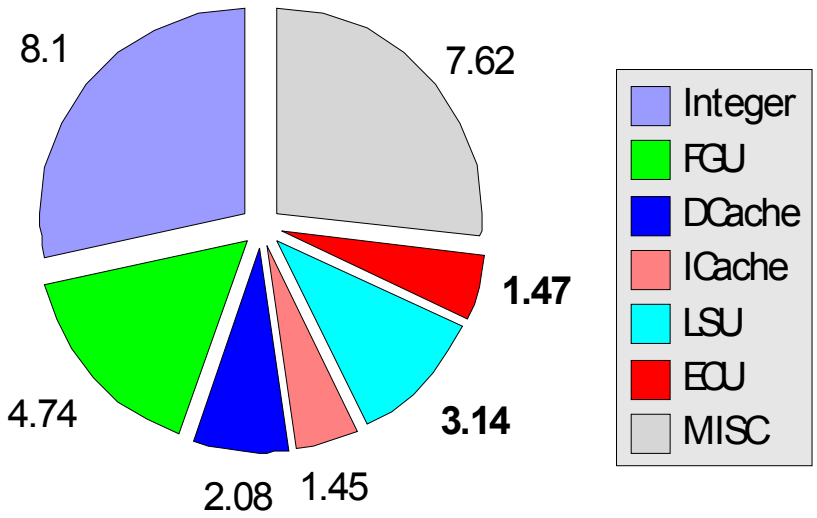
Google and SMT

- “A more profitable way to exploit parallelism for applications such as the index server is to leverage the trivially parallelizable computation.”
- “Exploiting such abundant thread-level parallelism at the microarchitecture level appears equally promising. Both simultaneous multithreading (SMT) and chip multiprocessor (CMP) architectures target thread-level parallelism and should improve the performance of many of our servers.”
- “Some early experiments with a dual-context (SMT) Intel Xeon processor show more than a 30 percent performance improvement over a single-context setup.”

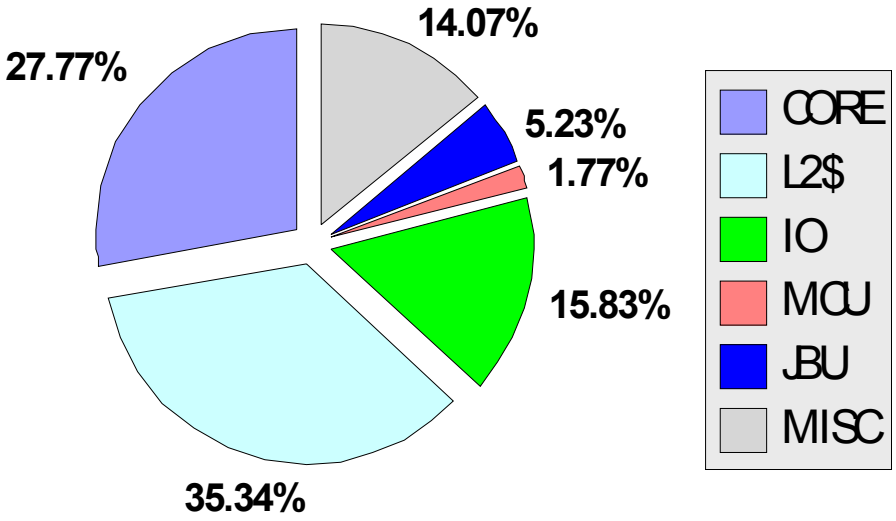
CMP: Chip Multiprocessing

- First CMPs: Two or more conventional superscalar processors on the same die
  - UltraSPARC Gemini, SPARC64 VI, Itanium Montecito, IBM POWER4
- One of the most important questions: What do cores share and what is not shared between cores?

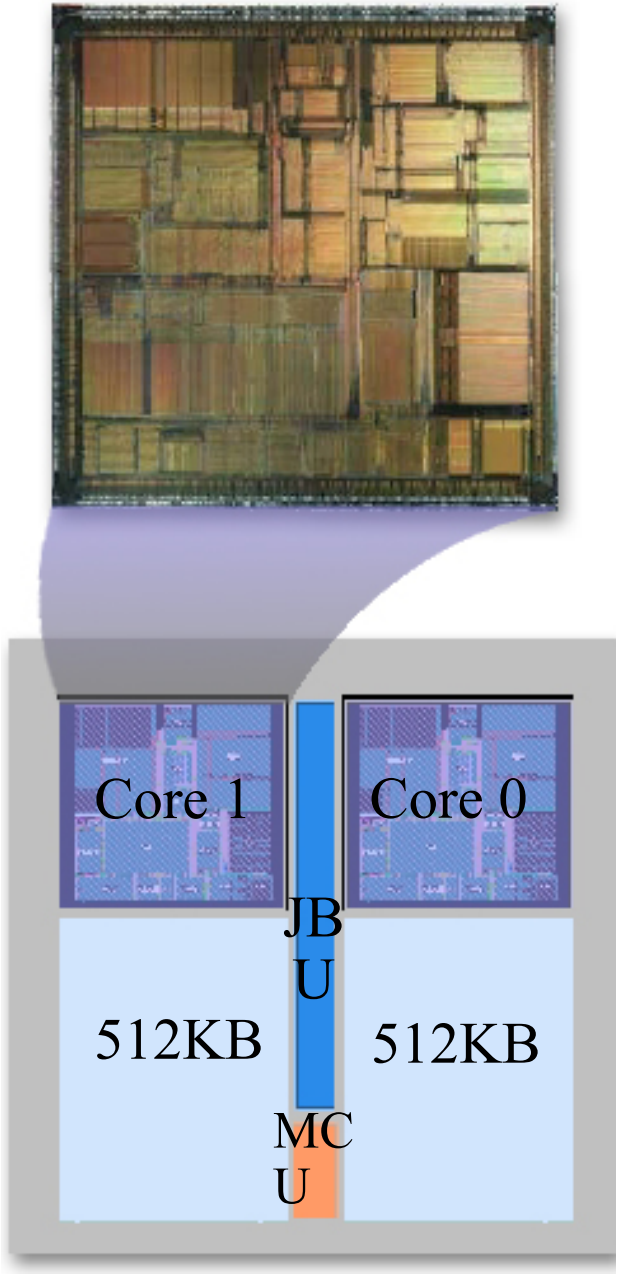
UltraSPARC Gemini



Core Area = 28.6 mm^2

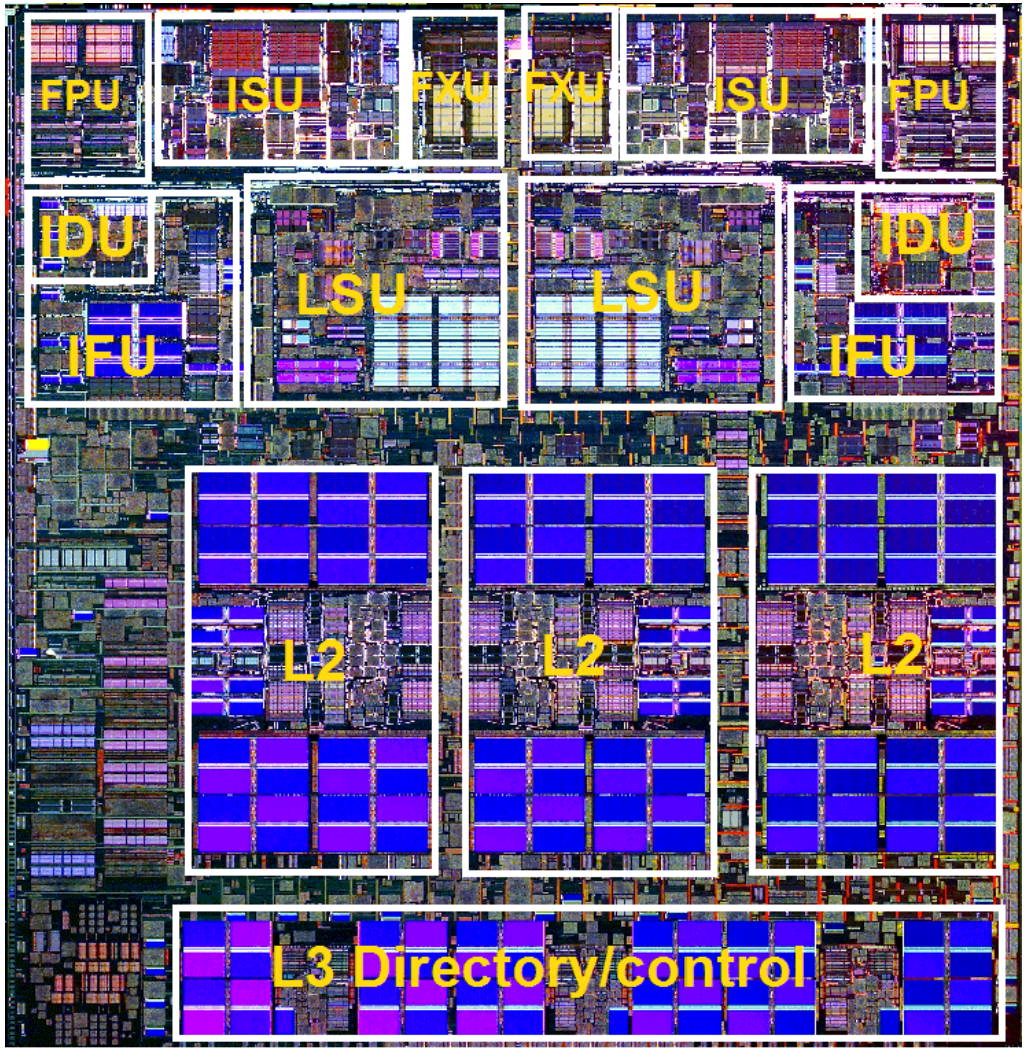


CPU Area = 206 mm^2



POWER5

- Technology: 130nm lithography, Cu, SOI
- Dual processor core
- 8-way superscalar
- Simultaneous multithreaded (SMT) core
  - Up to 2 virtual processors per real processor
  - 24% area growth per core for SMT
  - Natural extension to POWER4 design



CMP Benefits

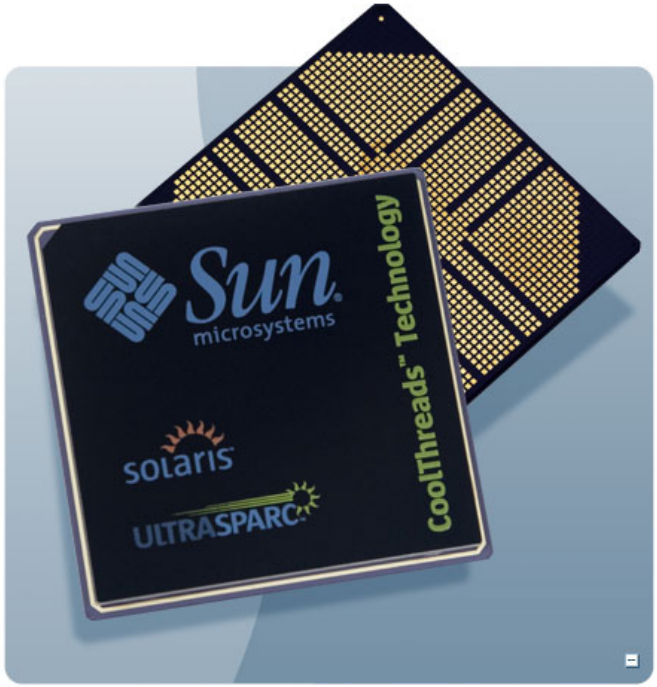
- Volume: 2 processors where 1 was before
- Power: All processors on one die share a single connection to rest of system

CMP Power

- Consider a 2-way CMP replacing a uniprocessor
- Run the CMP at half the uniprocessor's clock speed
  - Each request takes twice as long to process …
  - … but slowdown is less because request processing is likely limited by memory or disk
  - If there's not much contention, overall throughput is the same
- Half clock rate -› half voltage -› quarter power per processor, so 2x savings overall

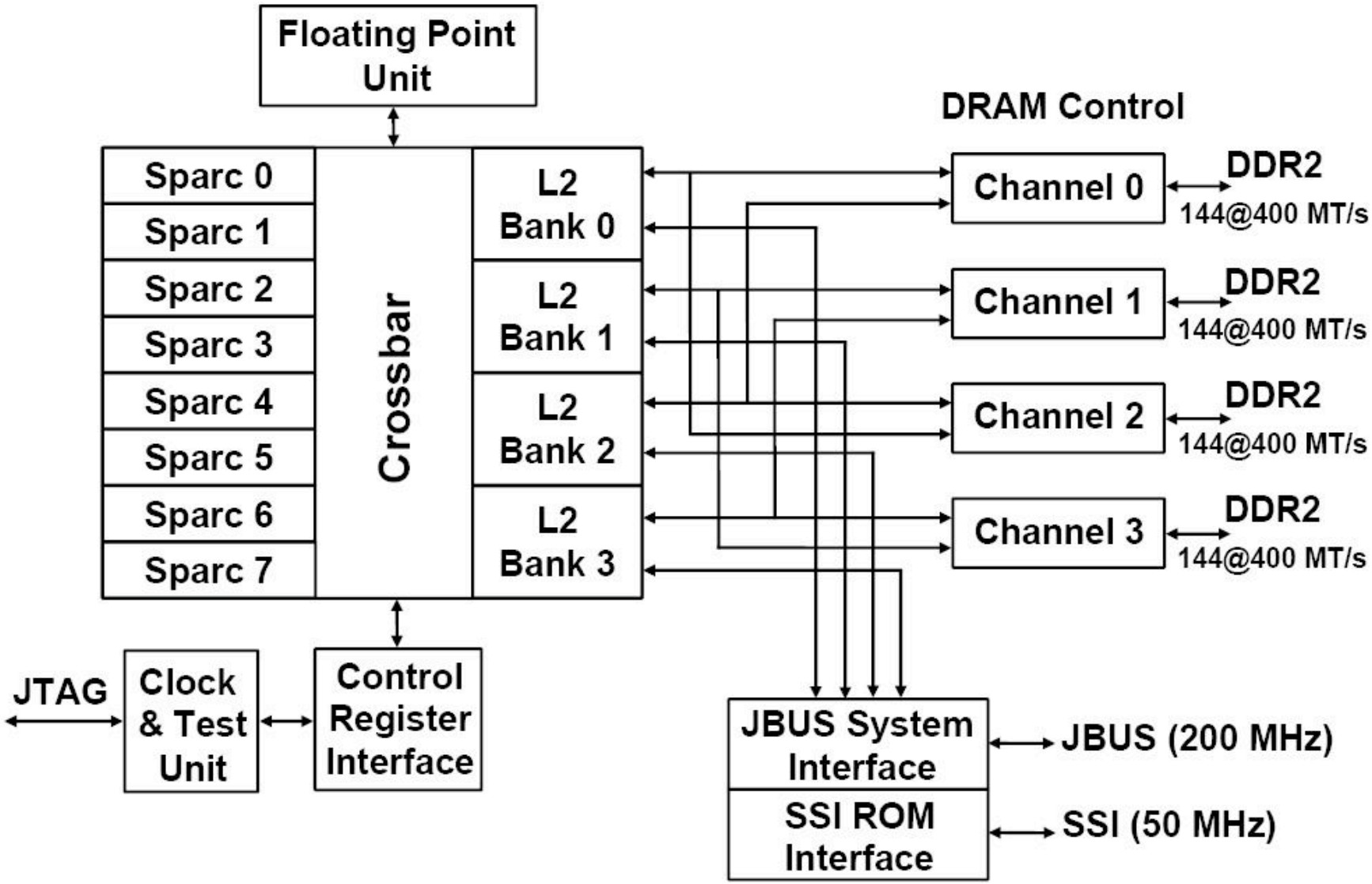
Sun T1 (“Niagara”)

- Target: Commercial server applications
  - High thread level parallelism (TLP)
    - Large numbers of parallel client requests
  - Low instruction level parallelism (ILP)
    - High cache miss rates
    - Many unpredictable branches
    - Frequent load-load dependencies



- Power, cooling, and space are major concerns for data centers
- Metric: Performance/Watt/Sq. Ft.
- Approach: Multicore, Fine-grain multithreading, Simple pipeline, Small L1 caches, Shared L2

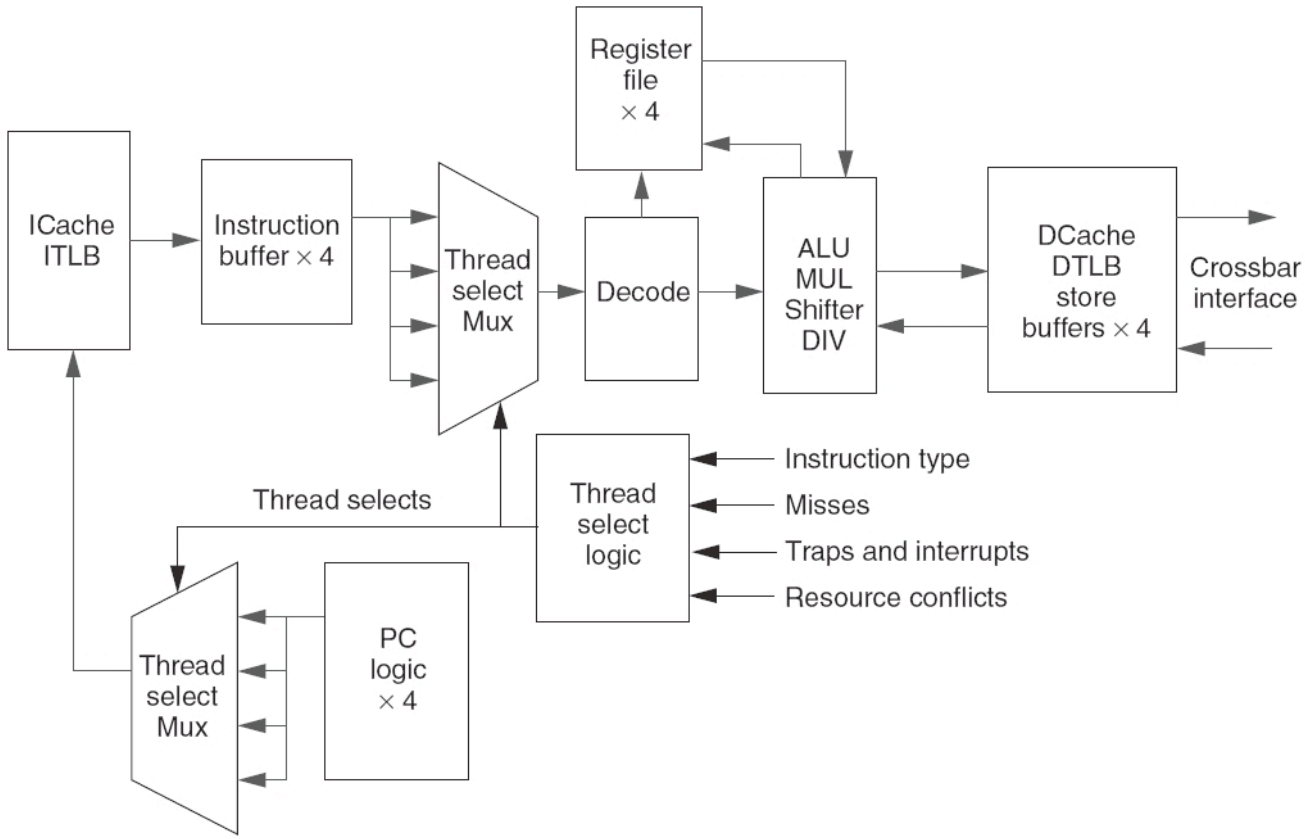
T1 Architecture



- Also ships with 6 or 4 processors

T1 pipeline

- Single issue, in-order, 6-deep pipeline: F, S, D, E, M, W
  - “Only single-issue desktop or server microprocessor introduced in more than 5 years.”
- 3 clock delays for loads & branches
- Shared units:
  - L1 $, L2 $
  - TLB
  - X units
  - pipe registers
- 1.2 GHz at ≈72W typical, 79W peak



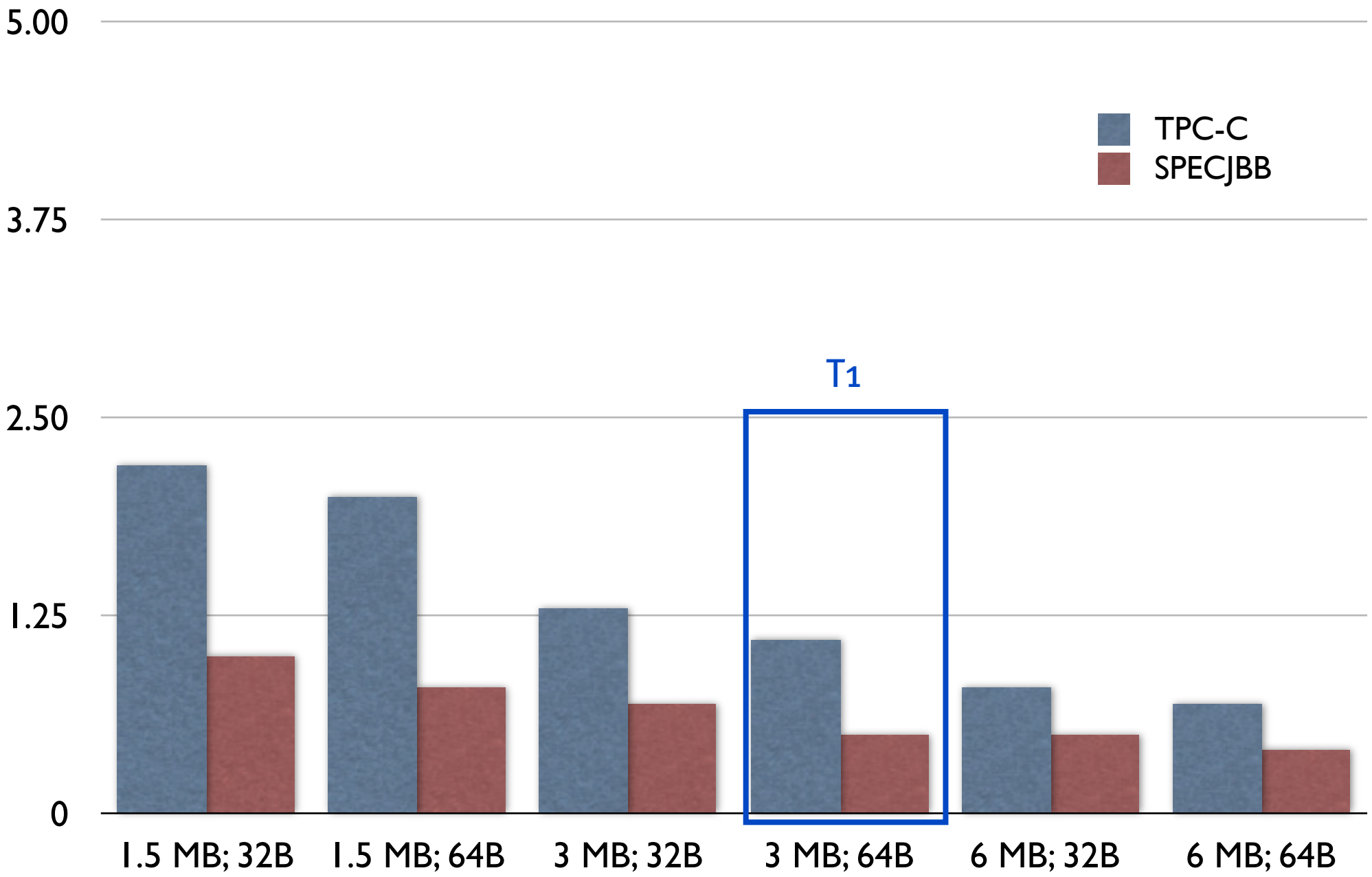
T1 Fine-Grained Multithreading

- Each core supports four threads and has its own level one caches (16 KB for instructions and 8 KB for data)
- Switching to a new thread on each clock cycle
- Idle threads are bypassed in the scheduling
  - Waiting due to a pipeline delay or cache miss
  - Processor is idle only when all 4 threads are idle or stalled
- Both loads and branches incur a 3 cycle delay that can only be hidden by other threads
- A single set of floating point functional units is shared by all 8 cores
  - Floating point performance was not a focus for T1

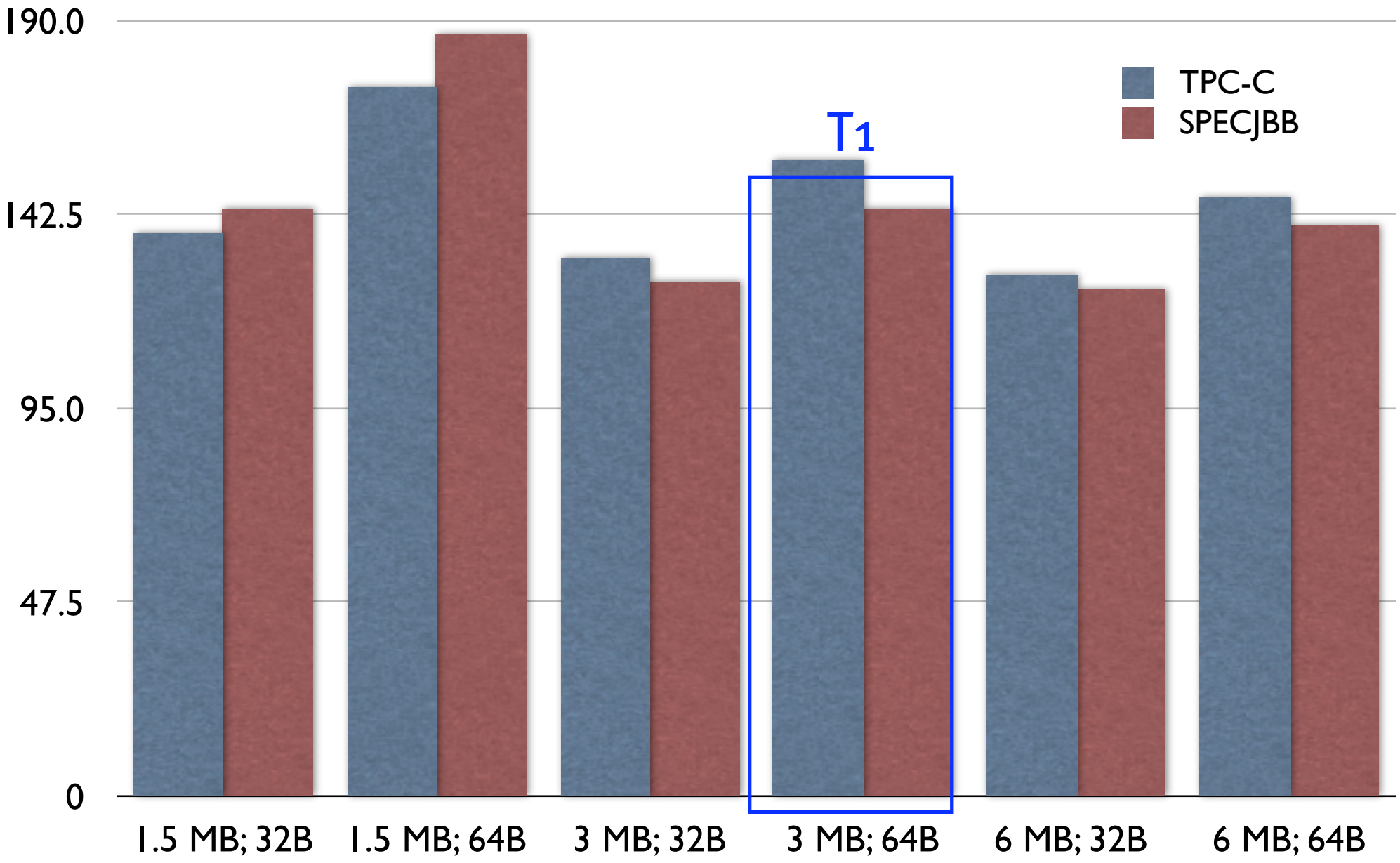
T1 Memory System

- 16 KB 4 way set assoc. I$/ core
- 8 KB 4 way set assoc. D$/ core
- 3MB 12 way set assoc. L2 $ shared
  - 4 x 750KB independent banks
  - Write-through, allocate for loads, no-allocate for stores
  - crossbar switch to connect
  - 2 cycle throughput, 8 cycle latency
  - Direct link to DRAM & Jbus
  - Manages cache coherence for the 8 cores
  - CAM based directory

Miss Rates: L2 Cache Size, Block Size



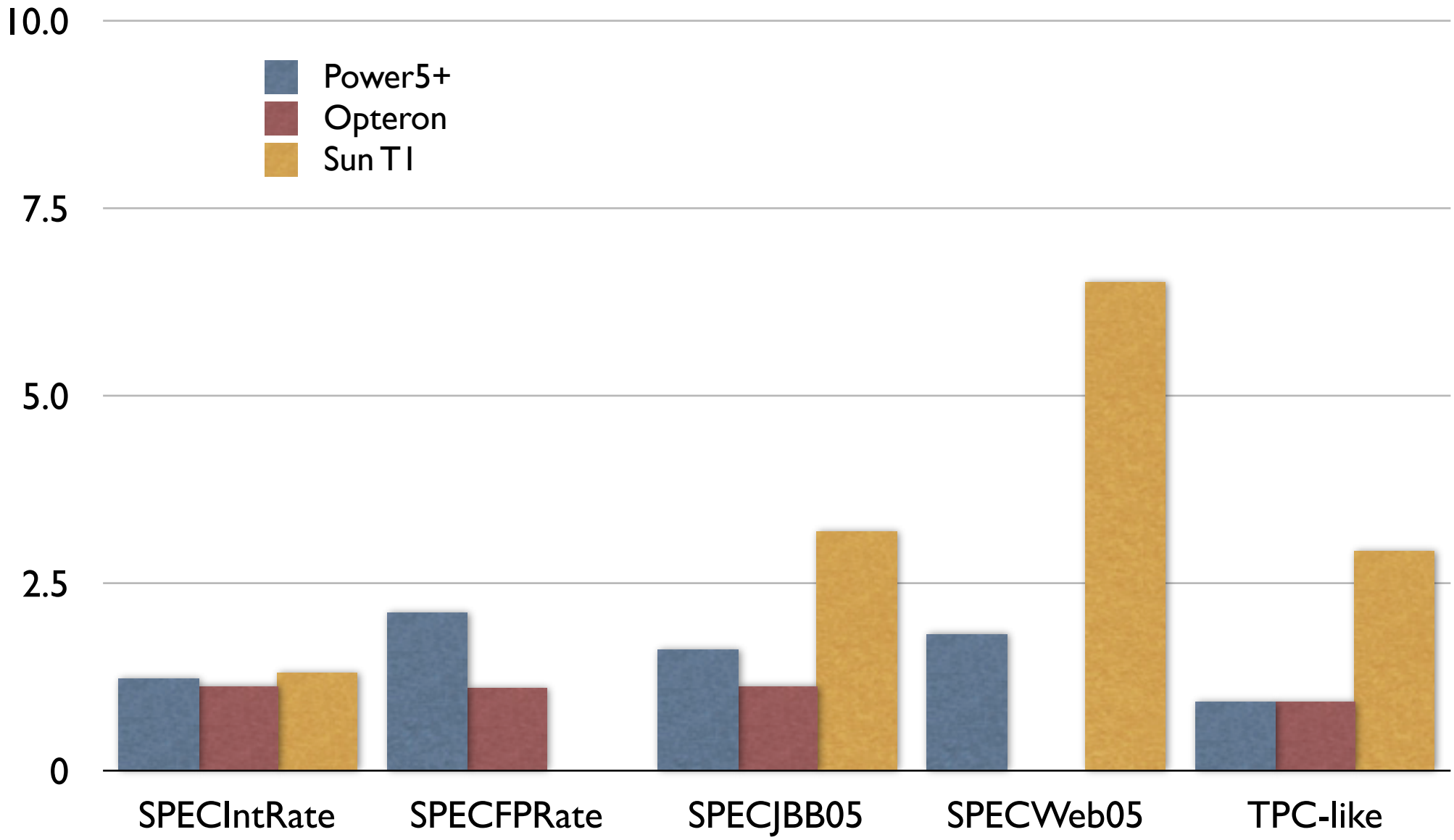
Miss Latency: L2 Cache Size, Block Size



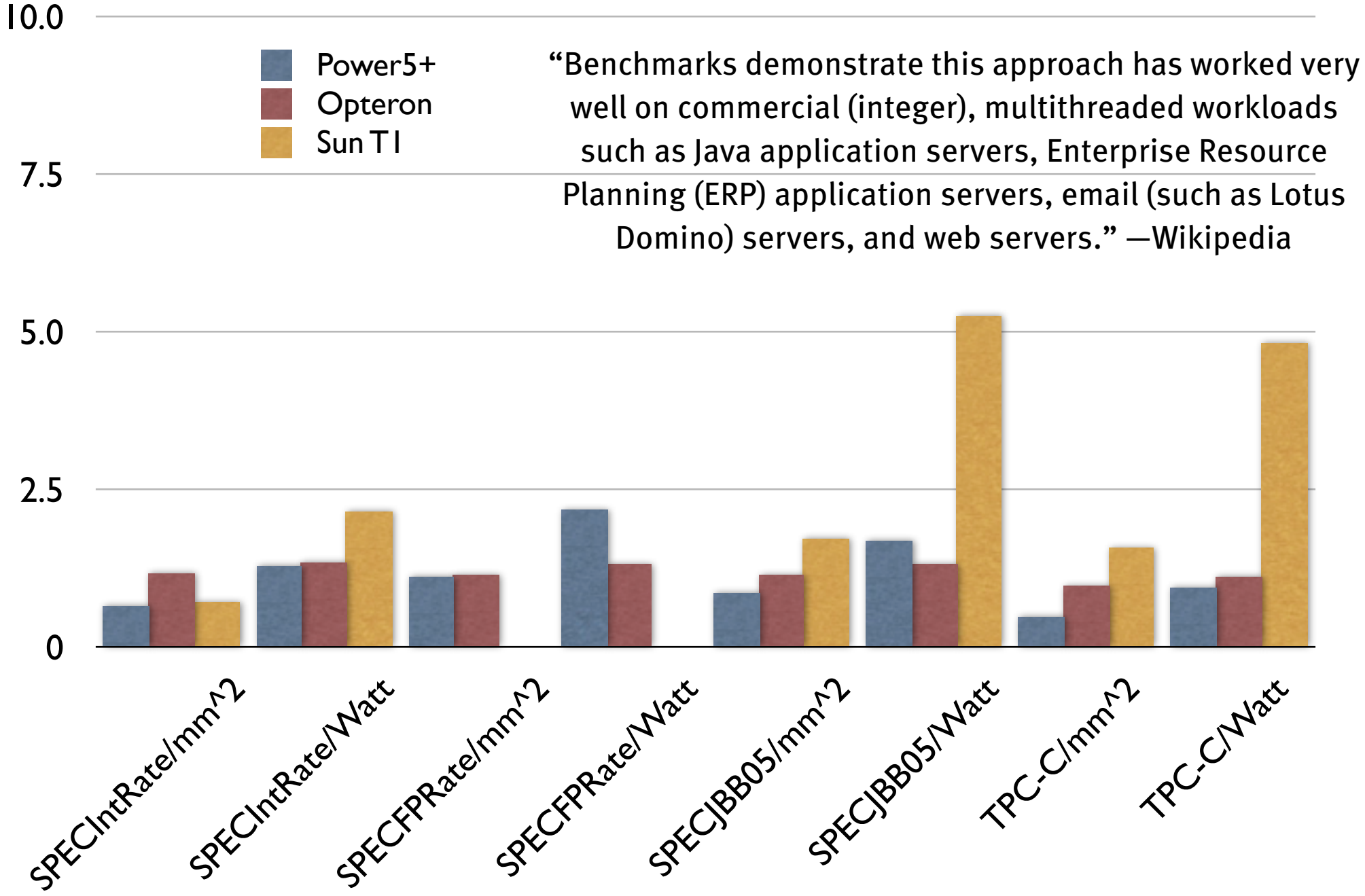
CPI Breakdown of Performance

Benchmark	Per Thread CPI	Per core CPI	Effective CPI for 8 cores	Effective IPC for 8 cores
TPC-C	7.20	1.80	0.23	4.4
SPECJBB	5.60	1.40	0.18	5.7
SPECWeb99	6.60	1.65	0.21	4.8

Performance Relative to Pentium D



Performance/mm², Performance/Watt



"Benchmarks demonstrate this approach has worked very well on commercial (integer), multithreaded workloads such as Java application servers, Enterprise Resource Planning (ERP) application servers, email (such as Lotus Domino) servers, and web servers." —Wikipedia

Microprocessor Comparison

Processor	SUN T1	Opteron	Pentium D	IBM Power 5
Cores	8	2	2	2
Instruction issues / clk / core	1	3	3	4
Peak instr. issues / chip	8	6	6	8
Multithreading	Fine-grained	No	SMT	SMT
L1 I/D in KB per core	16/8	64/64	12K uops/16	64/32
L2 per core/shared	3 MB shared	1 MB / core	1 MB/ core	1.9 MB shared
Clock rate (GHz)	1.2	2.4	3.2	1.9
Transistor count (M)	300	233	230	276
Die size (mm^2)	379	199	206	389
Power (W)	79	110	130	125

Niagara 2 (October 2007)

- Improved performance by increasing # of threads supported per chip from 32 to 64
  - 8 cores * 8 threads per core [now has 2 ALUs/core, 4 threads/ALU]
- Floating-point unit for each core, not for each chip
- Hardware support for encryption standards EAS, 3DES, and elliptical-curve cryptography
- Added 1 8x PCI Express interface directly into the chip in addition to integrated 10 Gb Ethernet XAU interfaces and Gigabit Ethernet ports.
- Integrated memory controllers will shift support from DDR2 to FB-DIMMs and double the maximum amount of system memory.
- Niagara 3 rumor: 45 nm, 16 cores, 16 threads/core

Kevin Krewell, “Sun's Niagara Begins CMT Flood—The Sun UltraSPARC T1 Processor Released”. Microprocessor Report, January 3, 2006