Industry perspective on Chip Multi-Threading: Bridging the gap with academia using OpenSPARC

Dwayne Lee
OpenSPARC Community Manager

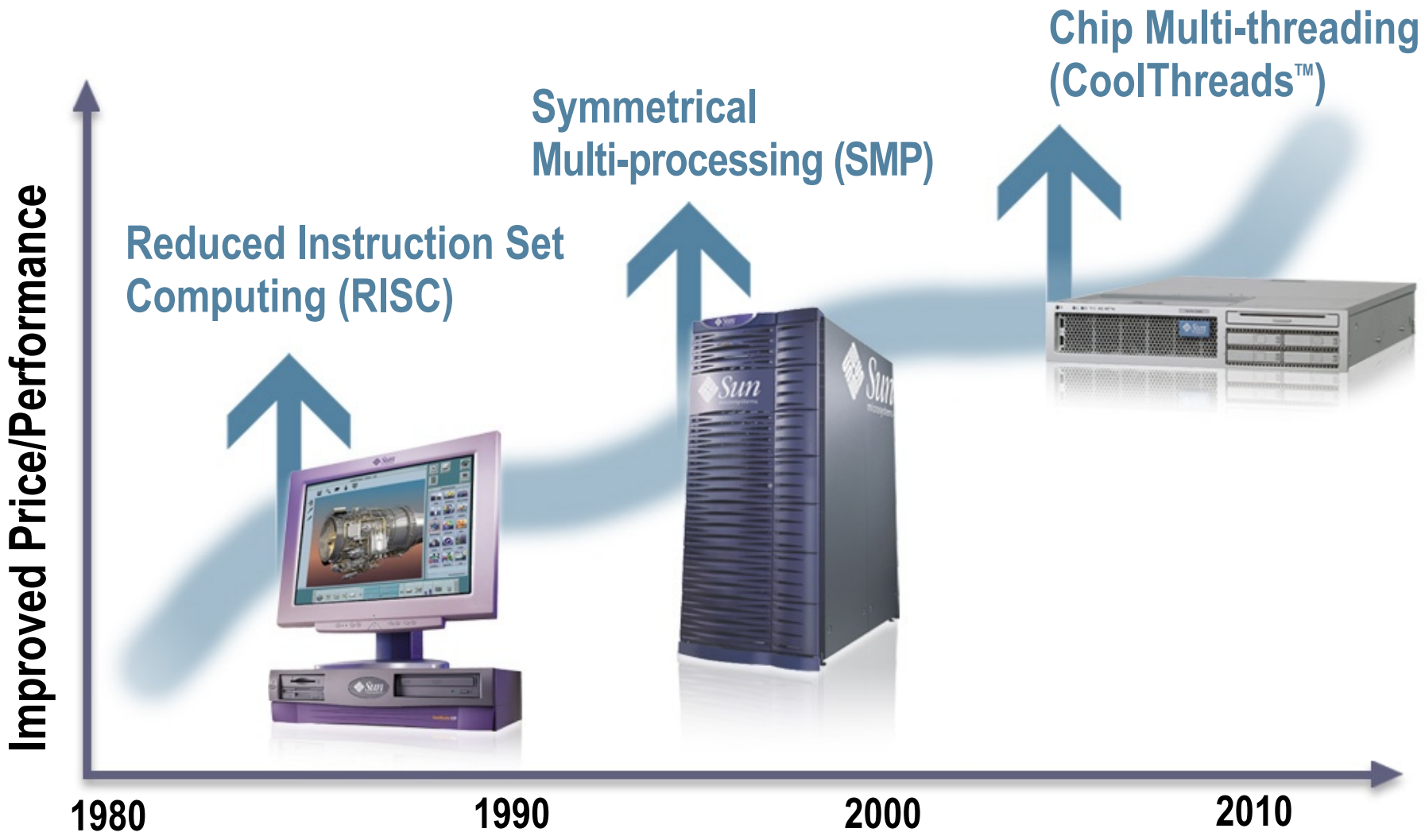
Shrenik Mehta
Senior Director,
Frontend Technologies & OpenSPARC Program

Microelectronics
Sun Microsystems, Inc.
www.opensparc.net

Agenda

1. Chip Multi-Threading (CMT) Era
2. Microarchitecture of OpenSPARC T1
3. OpenSPARC T1 Program
4. OpenSPARC in Academia

Making the Right Waves



The Processor Growth

1982 RISC 32 bit

1990 SMP 64 bit VIS 1.0

1998 RAS

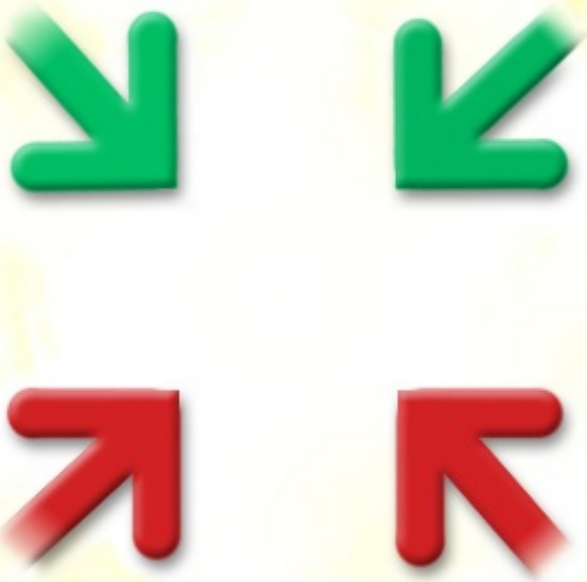
2005 SWAP CMT Hypervisor

Source: Sun Network San Francisco, NC03Q3, Sep. 17, 2003

The Big Bang Is Happening—Four Converging Trends

Network Computing Is Thread Rich

Web services, Java™ applications, database transactions, ERP . . .



Moore's Law

A fraction of the die can already build a good processor core; how am I going to use a billion transistors?

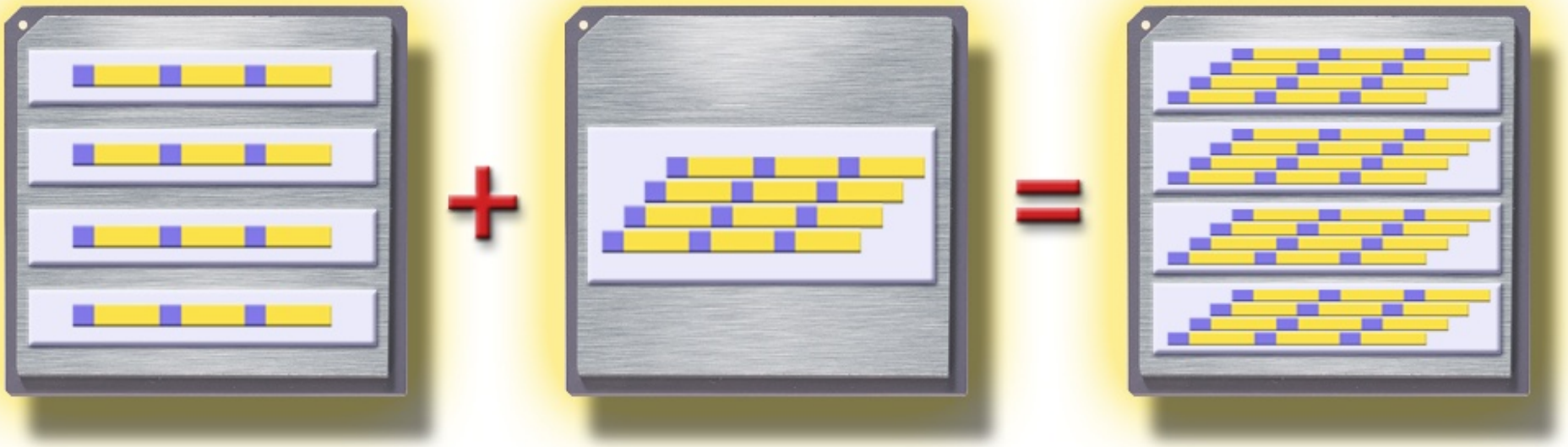
Worsening Memory Latency

It's approaching 1000s of CPU cycles! Friend or foe?

Growing Complexity of Processor Design

Forcing a rethinking of processor architecture – modularity, less is more, time-to-market

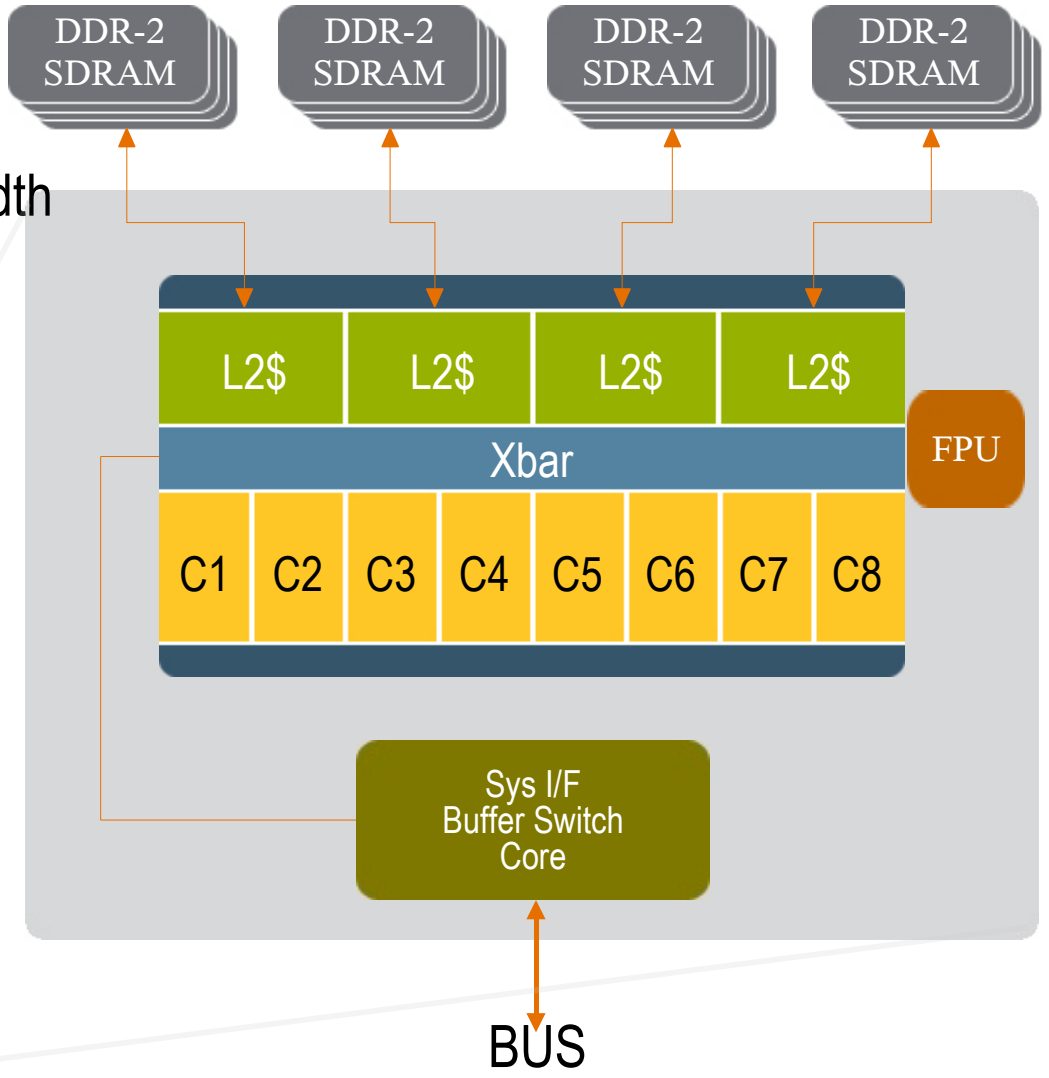
Chip Multi-Threading (CMT) to the rescue



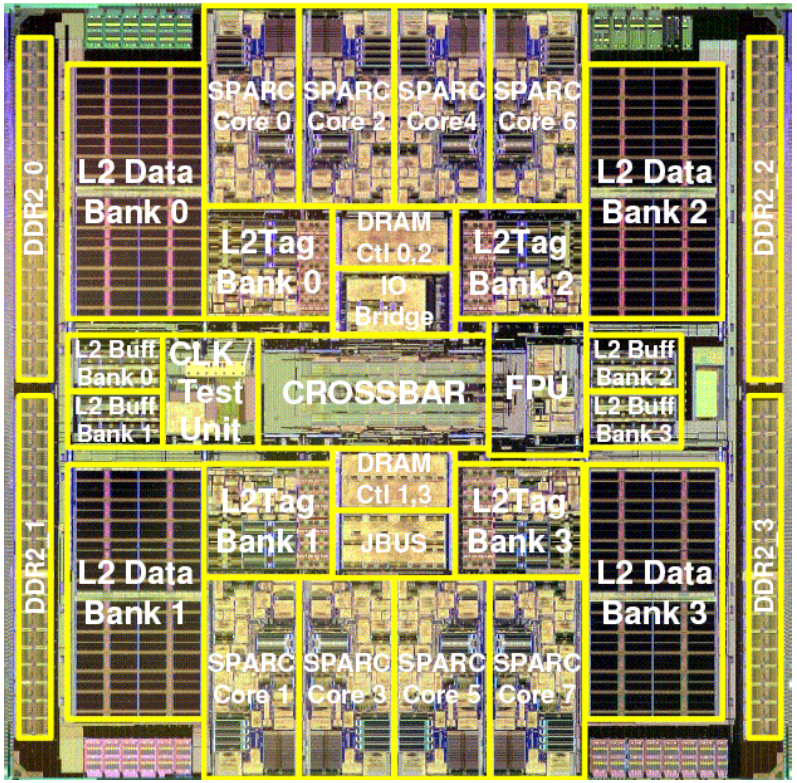
CMP	HMT	CMT
(chip multiprocessing)	(hardware multithreading)	(chip multithreading)
n cores per processor	m strands per core	n x m threads per processor

UltraSPARC T1 Processor

- SPARC V9 (Level 1) implementation
- Up to eight 4-threaded cores (32 simultaneous threads)
- All cores connected through high bandwidth (134.4GB/s) crossbar switch
- High-bandwidth, 12-way associative 3MB Level-2 cache on chip
- 4 DDR2 channels (23GB/s)
- Power : < 80W
- ~300M transistors
- 378 sq. mm die



UltraSPARC-T1: Choices & Benefits



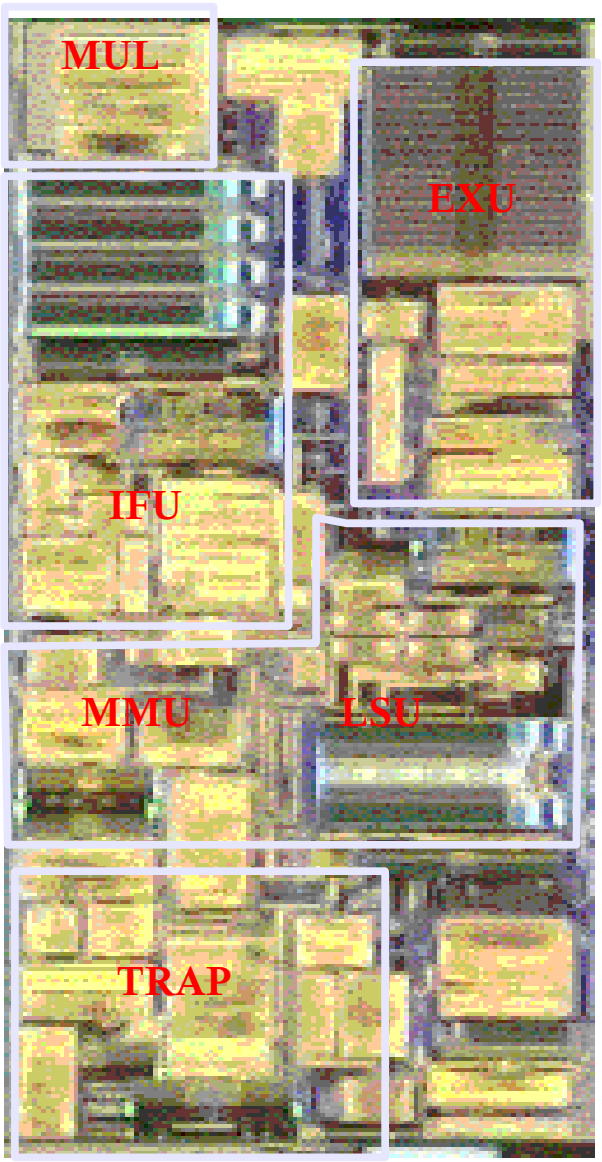
- Simple core (6-stage, only 11mm^2 in 90nm), 1 FPU
  - → maximum # of cores/threads on die
  - → pipeline built from scratch, useful for multiple generations
  - → modular, flexible design ... ***scalable*** (up and down)
- Caches, DRAM channels shared across cores
  - → better area utilization
- Shared L2 cache
  - → cost of coherence misses decrease by order of magnitude
  - → enables highly efficient multi-threaded software
- On-die memory controllers
  - → reduce miss latency
- Crossbar switch
  - → good for b/w, latency, functional verification

For reference: in 90nm technology, included 8 cores, 32 threads, and only dissipate 70^W

Sun microsystems

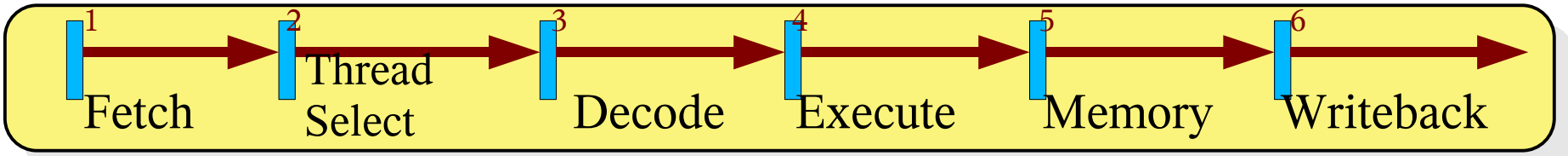
Microarchitecture details of the UltraSPARC -T1 CPU

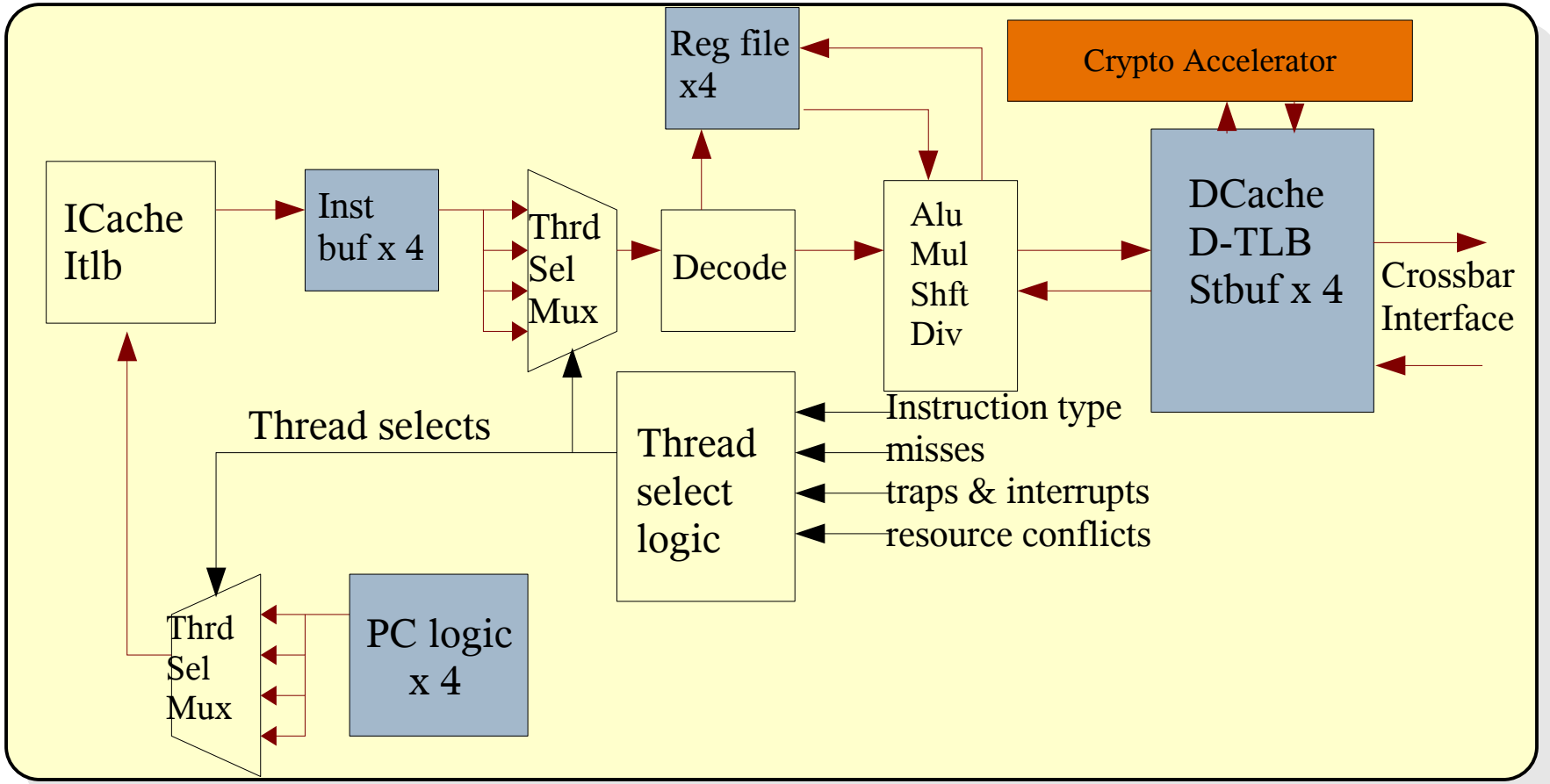
UltraSPARC-T1 Processor Core



- Four threads per core
- Single issue 6 stage pipeline
- 16KB I-Cache, 8KB D-Cache
- Unique resources per thread
  - Registers
  - Portions of I-fetch datapath
  - Store and Miss buffers
- Resources shared by 4 threads
  - Caches, TLBs, Execution Units
  - Pipeline registers and DP
- Core Area = 11mm2 in 90nm
- MT adds ~20% area to core

UltraSPARC T1 Processor Core Pipeline



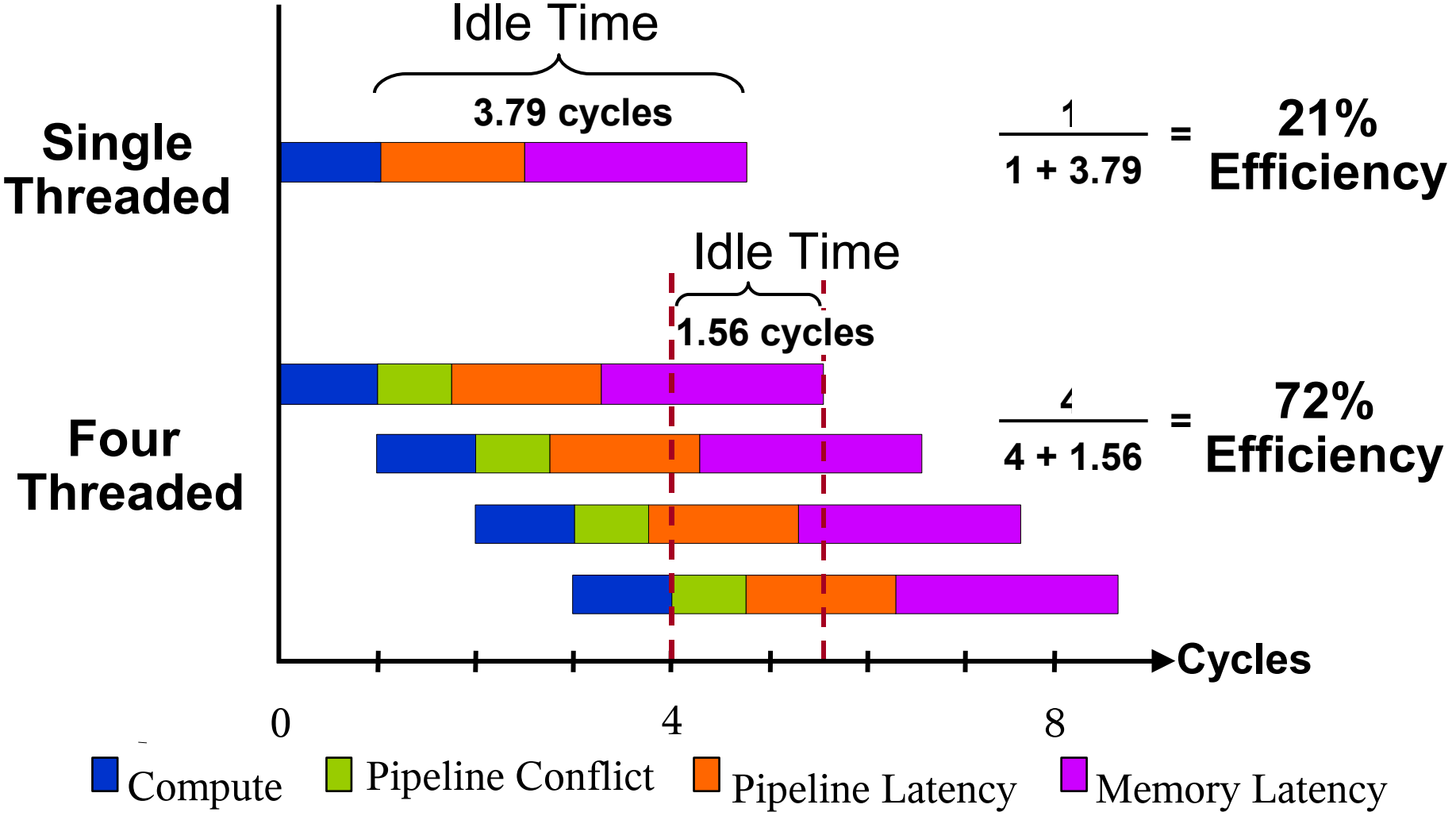


...blue units are replicated *per thread* on core

Thread Selection Policy

- Every cycle, switch among available (ready to run) threads
  - priority given to least-recently-executed thread
- Thread becomes not-ready-to-run due to:
  - Long latency operation like load, branch, mul, or div
  - Pipeline stall such as cache miss, trap, or resource conflict
- Loads are speculated as cache hits, and the thread is switched in with lower priority

Example - SpecJBB Execution Efficiency



A. S. Leon *et al.*, "A Power-Efficient High Throughput 32-Thread SPARC Processor," ISSCC06, Paper 5.1

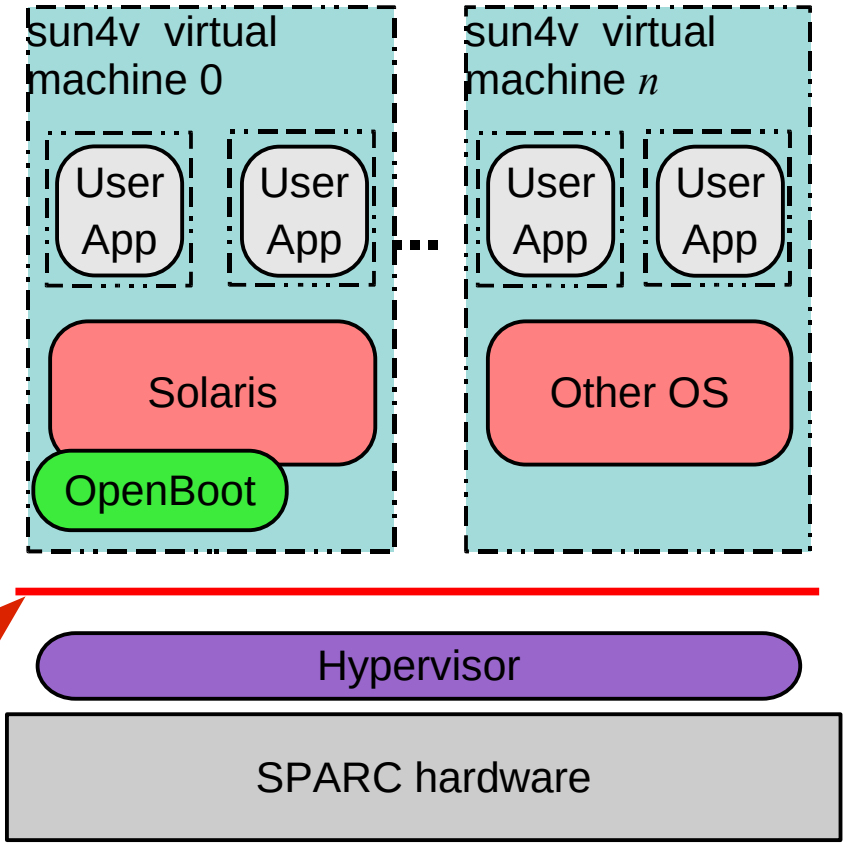
Virtualization on UltraSPARC T1

- Implementation on UltraSPARC-T1
  - Hypervisor uses Physical Addresses
  - Supervisor sees 'Real Addresses' – a PA abstraction
  - VA translated to 'RA' and then PA. Niagara MMU and TLB provides h/w support.
  - Upto 8 partitions can be supported. 3Bit partion ID is part of TLB translation checks
  - Additional trap level added for hypervisor use

Virtualization

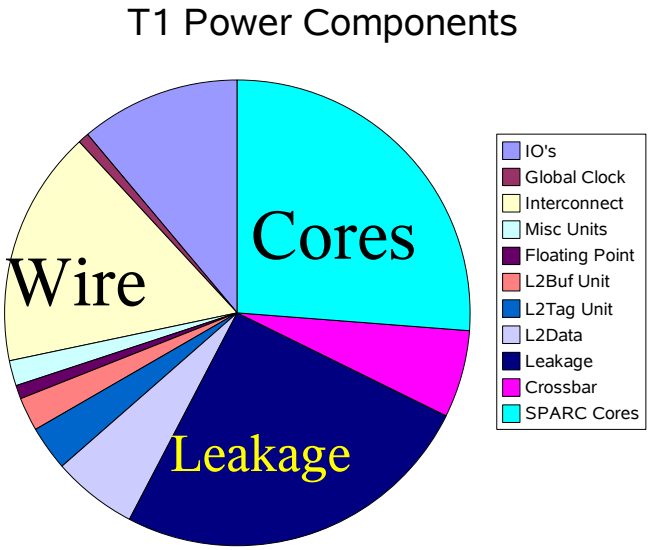
- Thin software layer between OS and platform hardware
- Para-virtualized OS
- Hypervisor + sun4v interface
  - Virtualizes machine HW and isolates OS from register-level
  - Delivered with *platform*, not with OS
  - Not itself an OS

stable interface "**sun4v**"



T1 Power

- Power Efficient Architecture
  - Single issue, in-order six stage pipeline
  - Minimal speculation, predication or branch prediction
- Thermal monitoring for power throttling
  - 3 external power throttle pins
    - Controlled by thermal diodes
    - Stall cycles injected, affecting all threads
  - Memory throttling
    - Open page limit
- Design Implementation
  - Fully static design
  - Fine granularity clock gating
    - Limited clock issue on stall, FGU
    - Limited L2 Cache & Memory clock gating
  - Wire classes optimized for power * delay



News from the Pacific Gas & Electric Company of California

Special Offer

Sun Servers Qualify for First-Ever Energy Rebate from PG&E

Pacific Gas & Electric rewards California customers with cash rebates when they upgrade to Sun Fire CoolThreads servers, the most energy-efficient servers in their class.

World's First Open Source Microprocessor

OpenSPARC.net

- Governed by GPL (2)
- Complete chip architecture
- Register Transfer Logic (RTL)
- Hypervisor API
- Verification suite and architectural models
- Simulation model for Solaris bringup on s/w
- 14 million lines of code

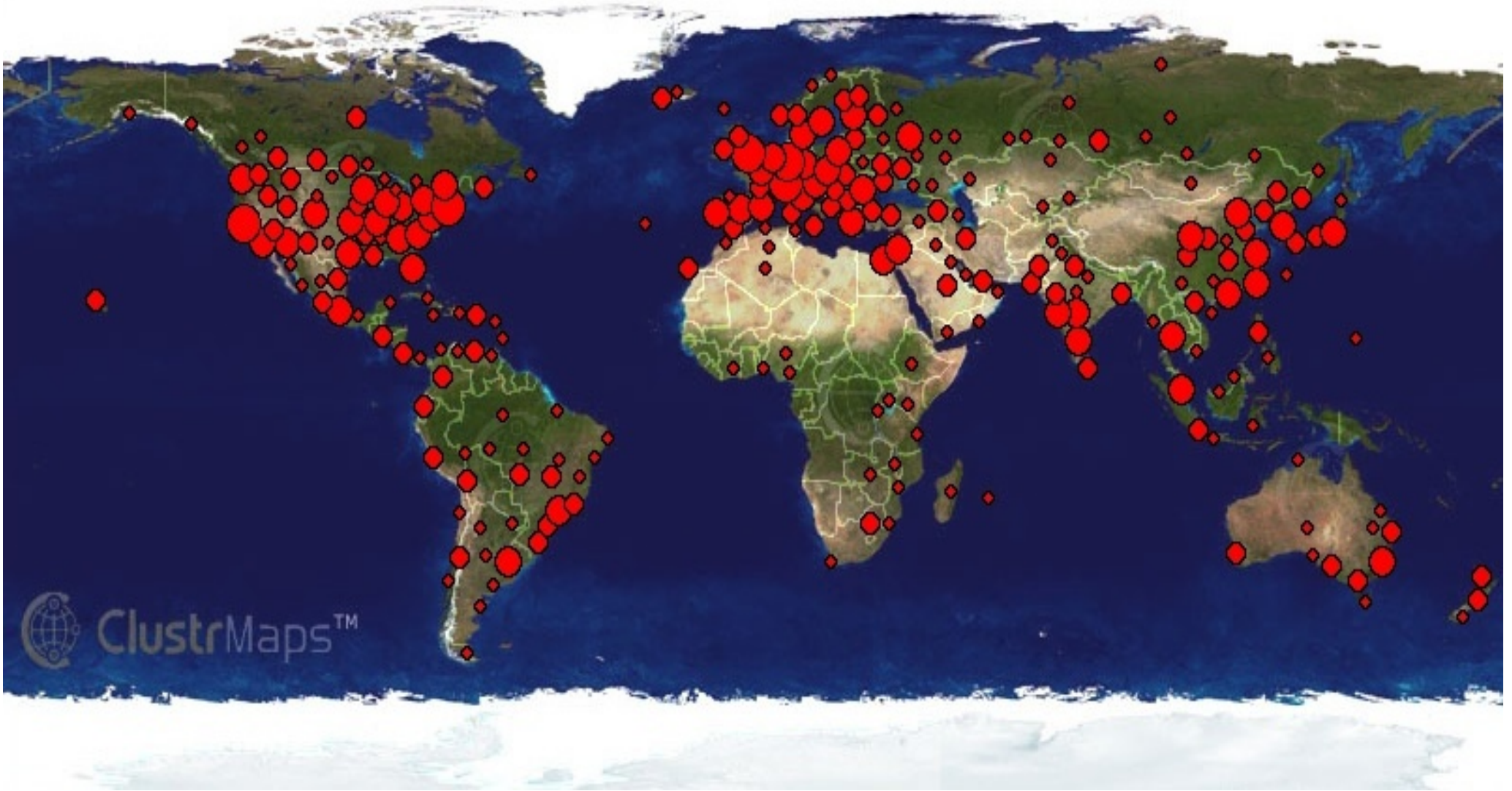
solaris™ OpenSPARC™

Linux

ubuntu

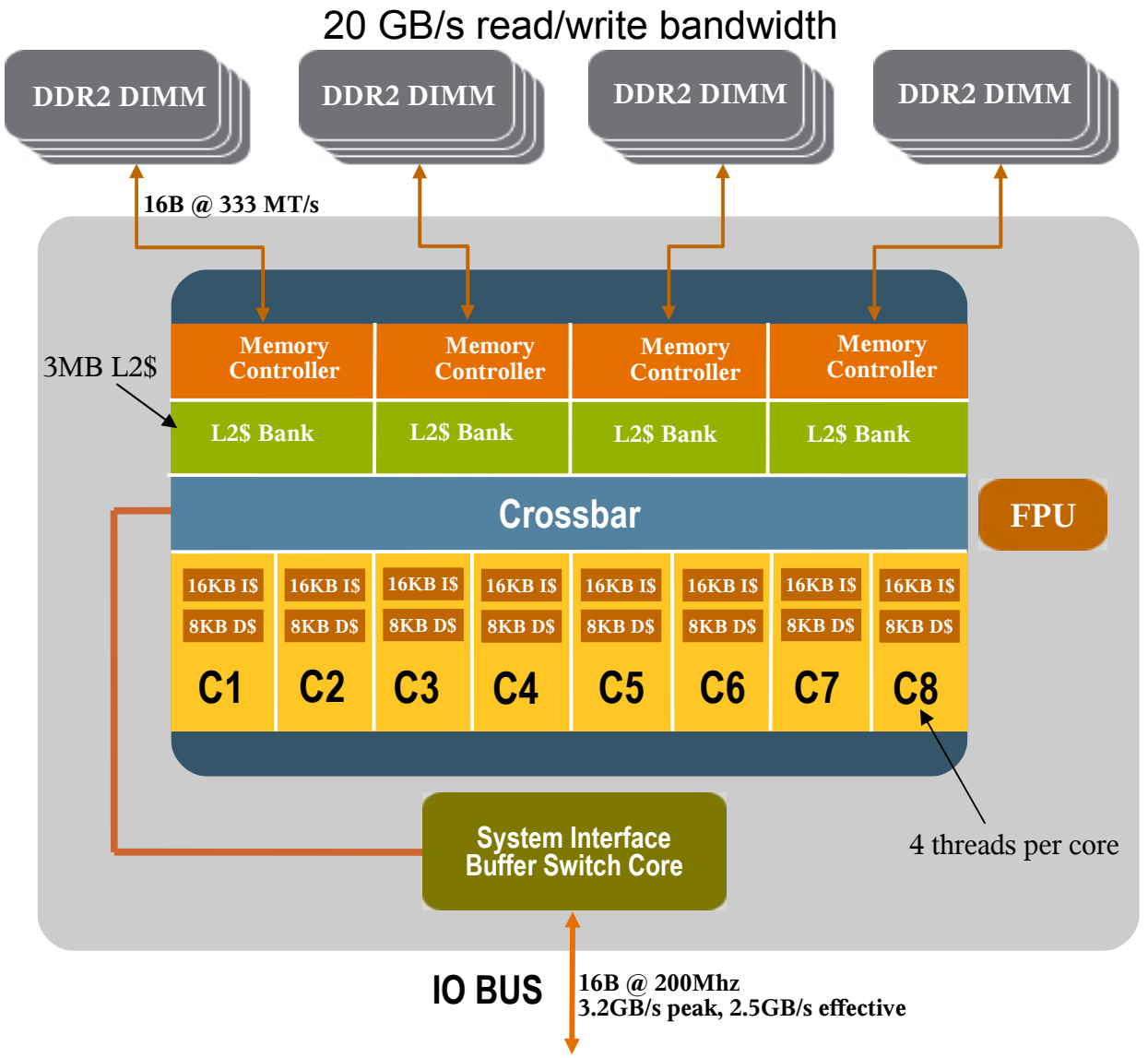
BSD

OpenSPARC momentum



Innovation Happens Everywhere > 5100 downloads

Get the Source ... Start Innovating!

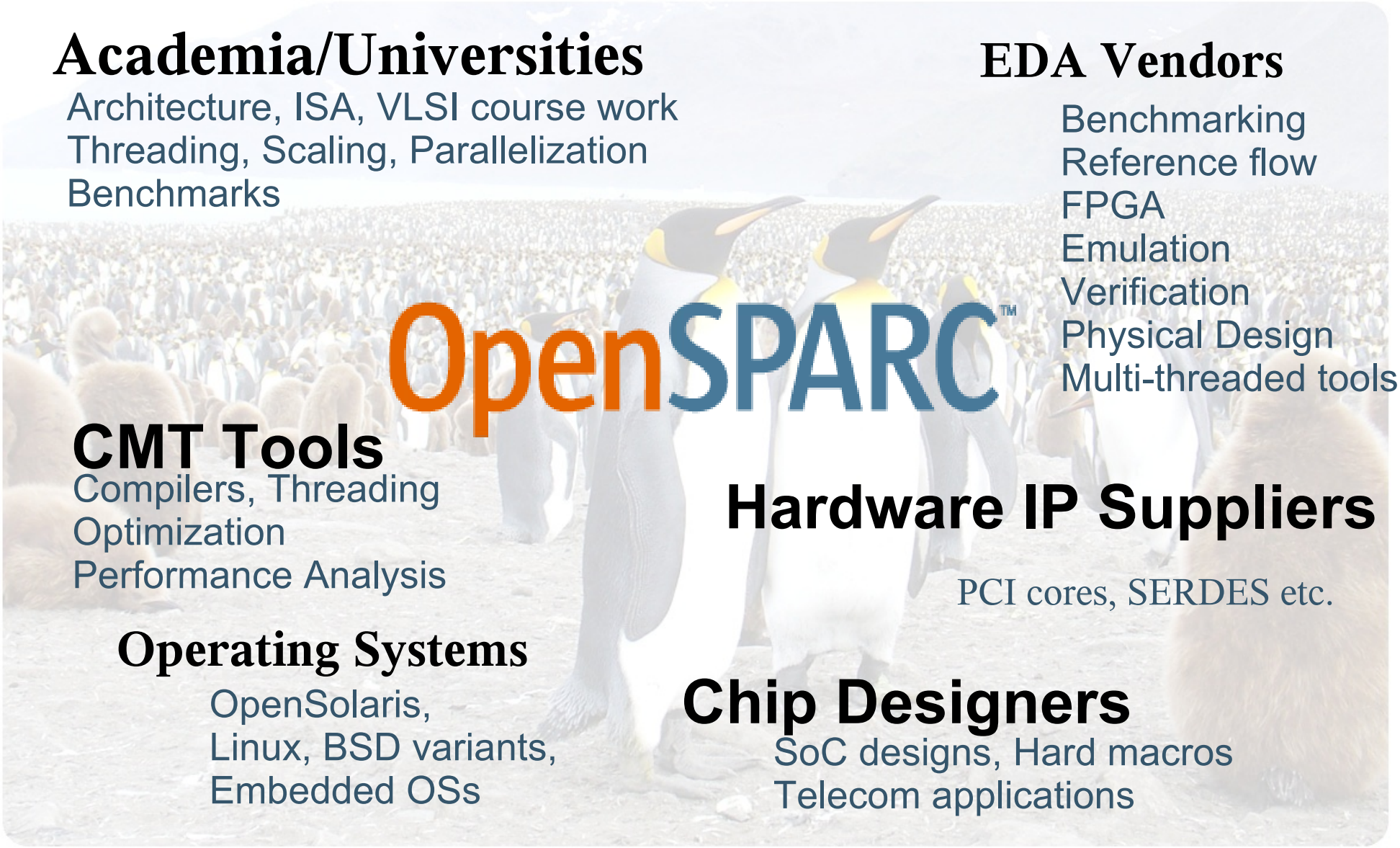


Things you can do:

- use as is
- add/delete threads
- add/delete cores
- add new instructions
- change or add FPUs
- add custom coprocessors
- add video/graphics
- add network interface
- change memory interface
- change I/O interface
- change cache/mem interface
- etc...

Innovate anywhere – within it or outside it

OpenSPARC Communities

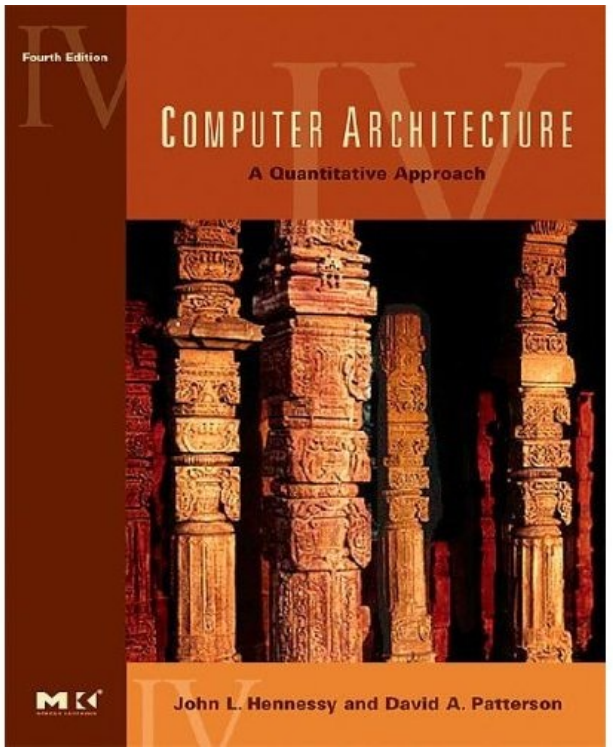


OpenSPARC community achievements

- Single core (S1) design released by Simply RISC based in Italy (less than 6 months of effort)
- Polaris Micro (China) taped out a chip in 130nm technology
- David Miller ported Linux in less than 6 weeks to T2000 system

OpenSPARC in the Curriculum

- UltraSPARC T1 is now the "putting it all together" example of multiprocessors in the world's most iconic textbook on Computer Architecture
- OpenSPARC Center of Excellence at University of California, SantaCruz



Computer Architecture:
A Quantitative Approach, 4th ed.
by John Hennessy and David Patterson

Processor Performance

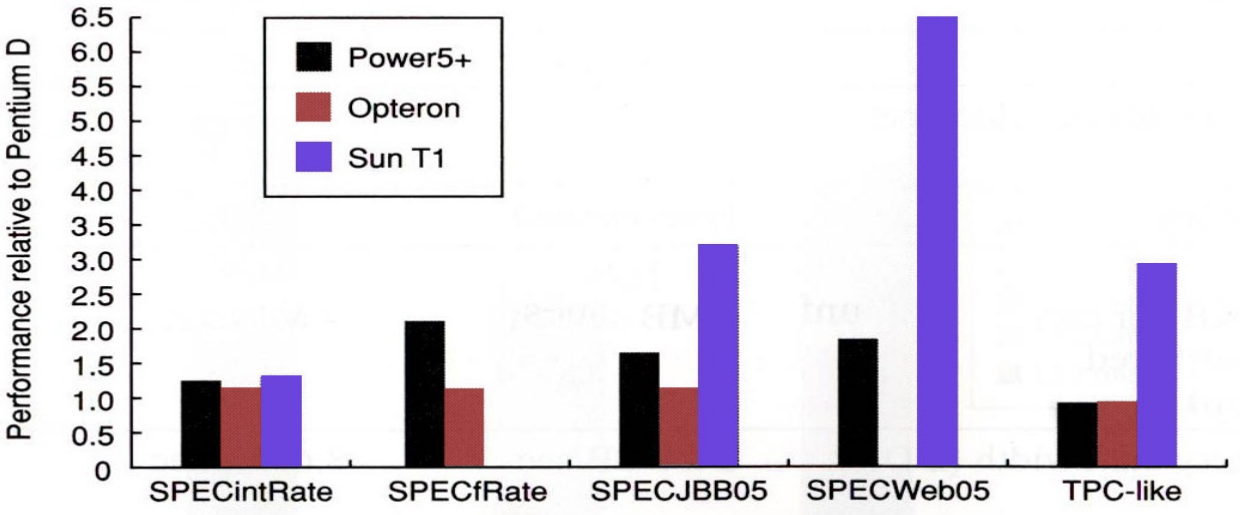


Figure 4.33 Four dual-core processors showing their performance on a variety of SPEC benchmarks and a TPC-C-like benchmark. All the numbers are normalized to the Pentium D (which is therefore at 1 for all the benchmarks). Some results are estimates from slightly larger configurations (e.g., four cores and two processors, rather than two cores and one processor), including the Opteron SPECJBB2005 result, the Power5 SPECWeb05 result, and the TPC-C results for the Power5, Opteron, and Pentium D. At the current time, Sun has refused to release SPECRate results for the FP portion of the suite.

Source: Computer Architecture, 4th ed. John Hennessy & David Patterson

Processor Performance

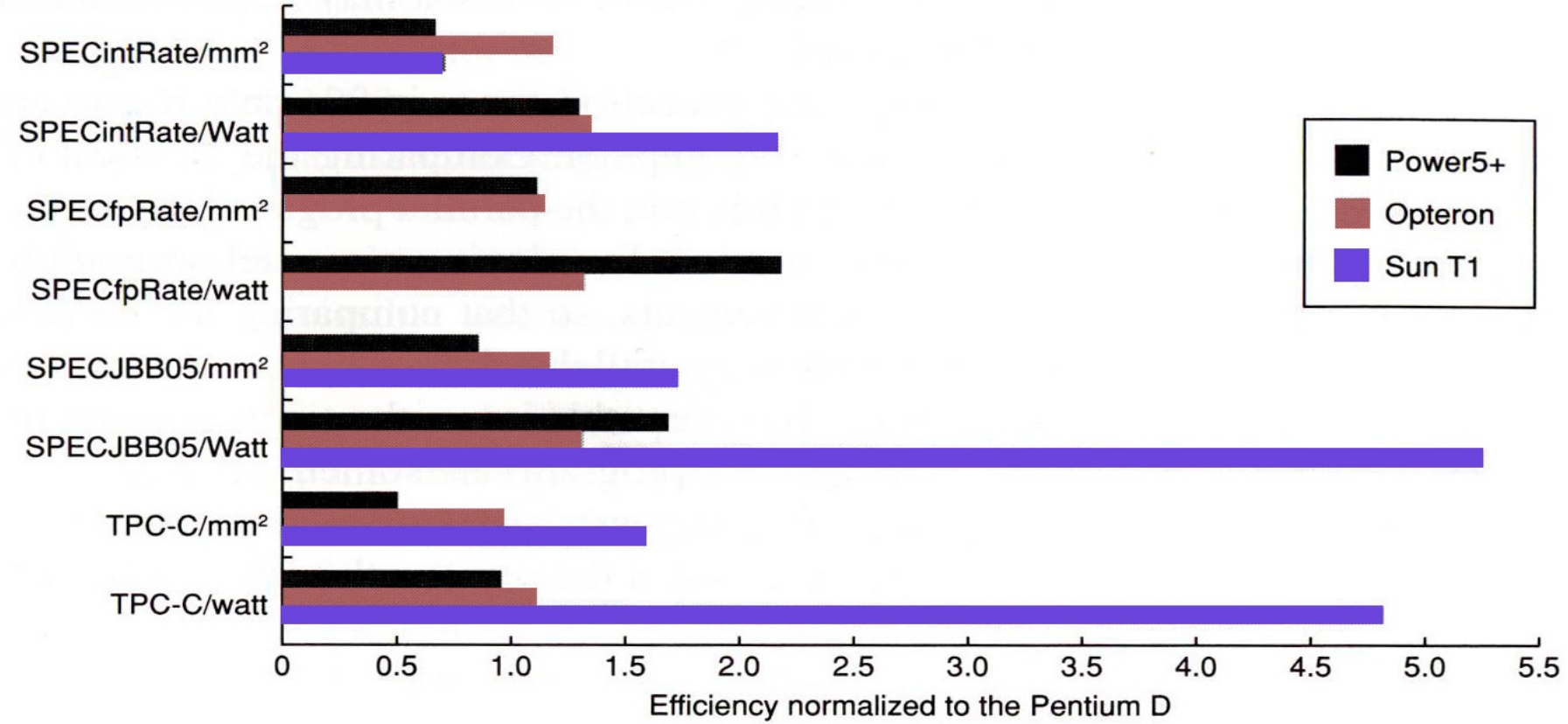


Figure 4.34 Performance efficiency on SPECRate for four dual-core processors, normalized to the Pentium D metric (which is always 1).

Source: Computer Architecture, 4th ed. John Hennessy & David Patterson

What's Available – for HW Engineering

- RTL (Verilog) of OpenSPARC T1 design
- RTL for *reduced (1 core, 1 thread)* OpenSPARC, for FPGA
- Synthesis scripts for RTL
- Verification test suites
- UltraSPARC Architecture 2005 spec
- UltraSPARC T1 implementation spec
- Full OpenSPARC simulation environment
- "CoolTools", including Sun Studio software, SPARC-optimized GCC compiler, development tools, ATS, etc

What's Available – for SW Engineering

- Architecture and Performance Modeling Package, including:
  - SAS – Instruction-accurate SPARC Architecture Simulator (includes source code)
  - SAM – SPARC instruction-accurate full-system simulator (includes source code)
  - Solaris Images for simulation: Solaris 10, Hypervisor, OBP
  - Legion – SPARC full-system simulation model for Software Developers (includes source code)
  - Hypervisor source code
  - Documentation

OpenSPARC -- What's Available

FPGA Implementations

FPGA Implementation

- Initial version released May 2006

  (on OpenSparc.net website)

  > full 8-core, 32-thread

  > First-cut implementation;
  not yet optimized for Area/Timing

  > Synplicity scripts for Xilinx/Altera FPGAs

- Reduced version released Mar 2007 – Release 1.4

  > single-core, single-thread

  > Reduced TLB

  > Optimizations for Area

OpenSPARC FPGA Implementation

- Single core, single thread implementation of T1
  - Small, clean and modular FPGA implementation
    - About 39K 4-input LUTs, 123 BRAMs (synplicity on Virtex{2/2Pro/4})
    - Synchronous, no latches or gated clocks
    - Better utilization of FPGA resources (BRAMs, Multiplier)
  - Functionally equivalent to custom implementation, except
    - 8 entry Fully Associative TLB as opposed to 64 entry
    - Removed Crypto unit (modular arithmetic operations)

Single Thread T1 on FPGAs

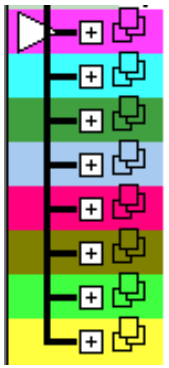
- Functionally stable
  - Passing mini and full regressions
- Completely routed
  - No timing violations
  - Easily meets 20ns (50MHz) cycle time
- Expandable to more threads
  - Reasonable overhead for most blocks (~30% for 4 threads)
  - Some bottlenecks exist (Multi-port register files)

System Theory of Operation – T1 on FPGA

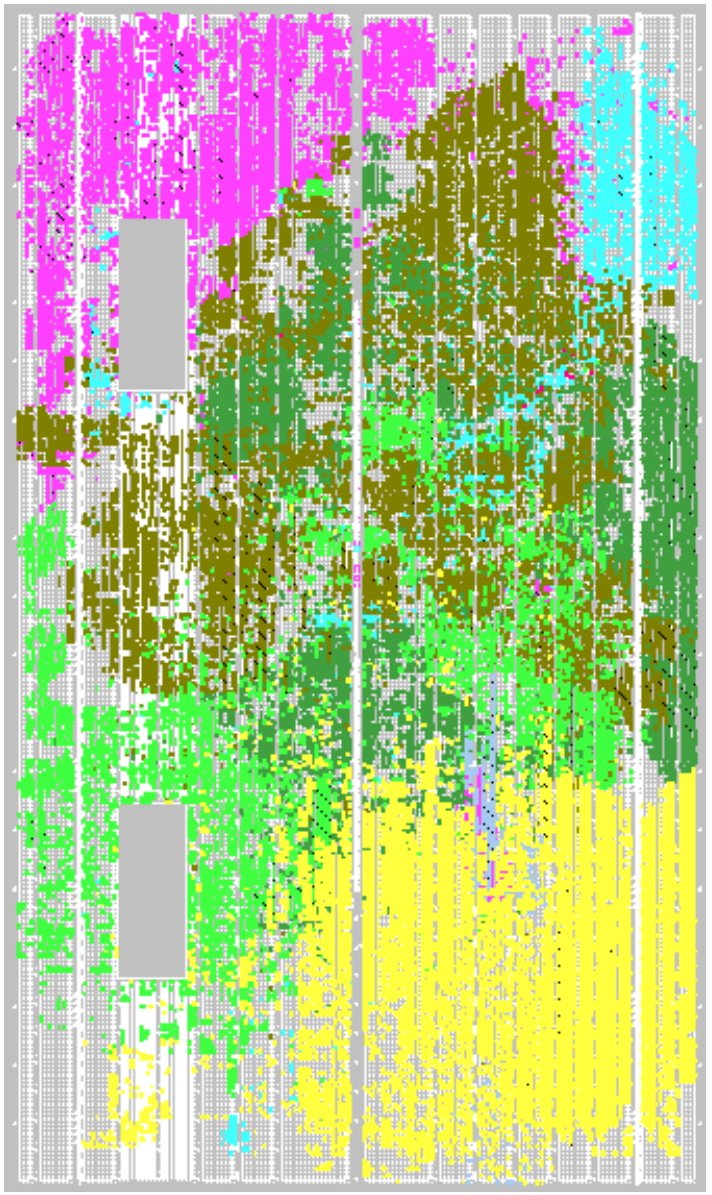
- OpenSPARC T1 core communicates exclusively via the processor-to-crossbar interface (PCX)
  - PCX is a packet based interface
- Microblaze softcore will sit in a polling loop and accept these packets, perform any protocol conversion, and forward them to the appropriate peripheral
  - Could even implement floating point operations via the Microblaze FPU unit
- Microblaze will also poll (or accept interrupts from) the peripherals, convert the info to a PCX packet, and forward it to the PCX interface
  - Microblaze has its own UART for its own diagnostic input/output

Implementation Results

- XC4VFX100-11FF1152 FPGA
  - > 42,649/84,352 LUT4s (50%)
  - > 131/376 BRAM-16kbits (34%)
  - > 50MHz operation
    - > Have not attempted any faster
  - > Synplicity Synthesis: 25 minutes
  - > Place and Route: 42 minutes



GRP1 "Grouped_by_User" *(Microblaze & Related Logic)*
iop_fpga_0/iop_fpga_0/sparc0/ffu "sparc_ffu"
iop_fpga_0/iop_fpga_0/sparc0/ifu "sparc_ifu"
iop_fpga_0/iop_fpga_0/sparc0/mul "sparc_mul_top"
iop_fpga_0/iop_fpga_0/sparc0/test_stub "test_stub_bist"
iop_fpga_0/iop_fpga_0/sparc0/lsu "lsu"
iop_fpga_0/iop_fpga_0/sparc0/tlu "tlu"
iop_fpga_0/iop_fpga_0/sparc0/exu "sparc_exu"



Preliminary Virtex5 Results

- Virtex5 xc5vlx1 10tff1 136
  - > Same as FPGA in RAMP Bee3 board
- 30,508 6-input LUTs used out of 69,120 (44%)
- 119 used out of148 BRAM-36kbits (80%)
  - > Working through mapping issues…
- 50MHz placed and routed design
  - > Have not attempted any faster

Results (2)

Design	LUTs	Block RAMs	Latches	Comments
4-thread SPARC core (Before)	135190	4	3123	OpenSPARC T1 v 1.3
4-thread SPARC core (After)	70152	127	0	OpenSPARC T1 v1.4 FPGA_SYN opt
1-thread SPARC core (After)	39084	123	0	OpenSPARC T1 v1.4 FPGA_SYN, 1THREAD, and NO_CRYPTO

Results from Synplicity Sinplify Pro v5.8 on Virtex-4 devices

- Functionally clean
  - > Passes verification suit that is part of OpenSPARC T1

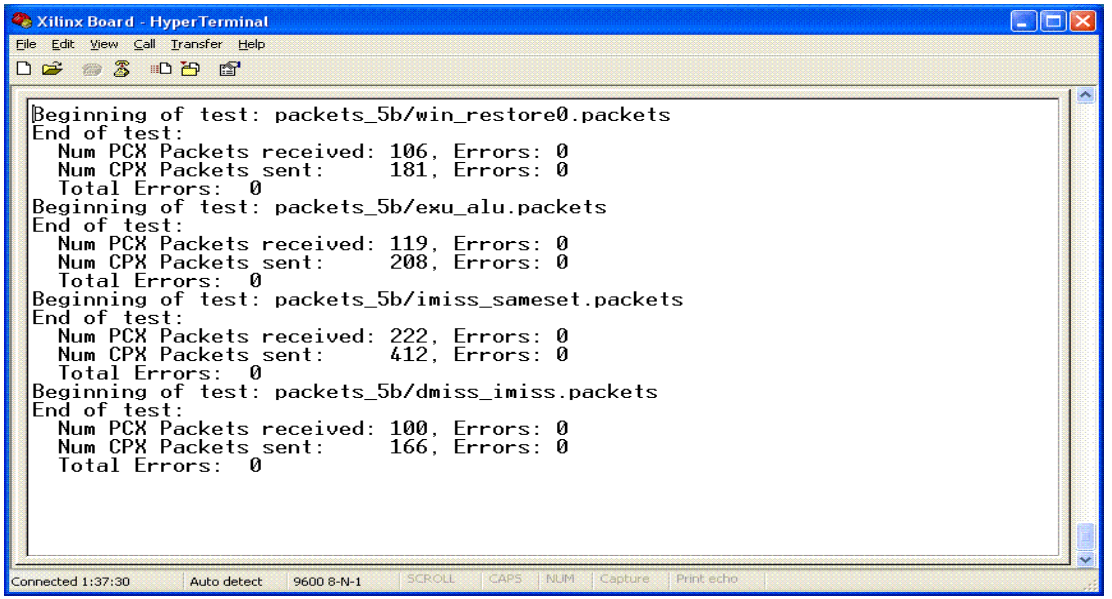
Future Directions (1)

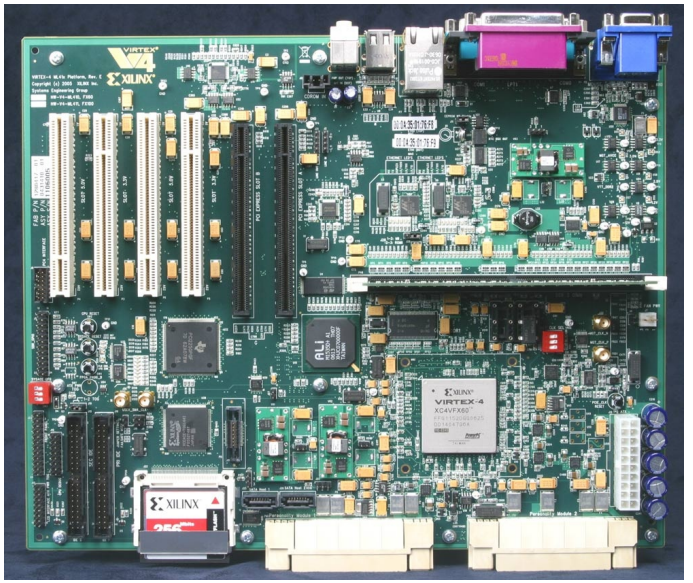
- Efficient Multi-Threaded (MT) processor on FPGAs
- Very interesting for hardware emulation
- Most blocks scale between 30-35% for 4-thread MT
  - Except Execute Unit – analyze bottlenecks

Blocks	Single thread	Four thread	MT Overhead
Fetch	7432	10210	0.27
Execute	12252	31018	0.61
Load/Store	9559	12939	0.26
Trap Logic	5727	8410	0.32
Floating point frontend	2168	2388	0.09

FPGA Reference Design

- ml410 board with Virtex4-100 FPGA (aka ml411)
  - > Bit file and elf is stored on CompactFlash card
- Each design is a hardware implementation of one regression suite test
  - > Microblaze soft-core sends the test packets to the OpenSPARC core and verifies the return packets





OpenSPARC SAM-T1/Legion Simulator

Software simulators

OpenSPARC T1 Arch Tools Download

- http://opensparc-t1.sunsource.net/download_sw.html
- OpenSPARCT1-Arch_1.3.tar.bz
  - SAM: instruction-accurate SPARC full-system simulator
  - SAS: instruction-accurate SPARC arch. simulator
  - Binary images for simulation: Solaris 10, Hypervisor, OBP, etc
  - Legion: SPARC full-system simulator for software development
  - Hypervisor source code
  - Documentation

Legion

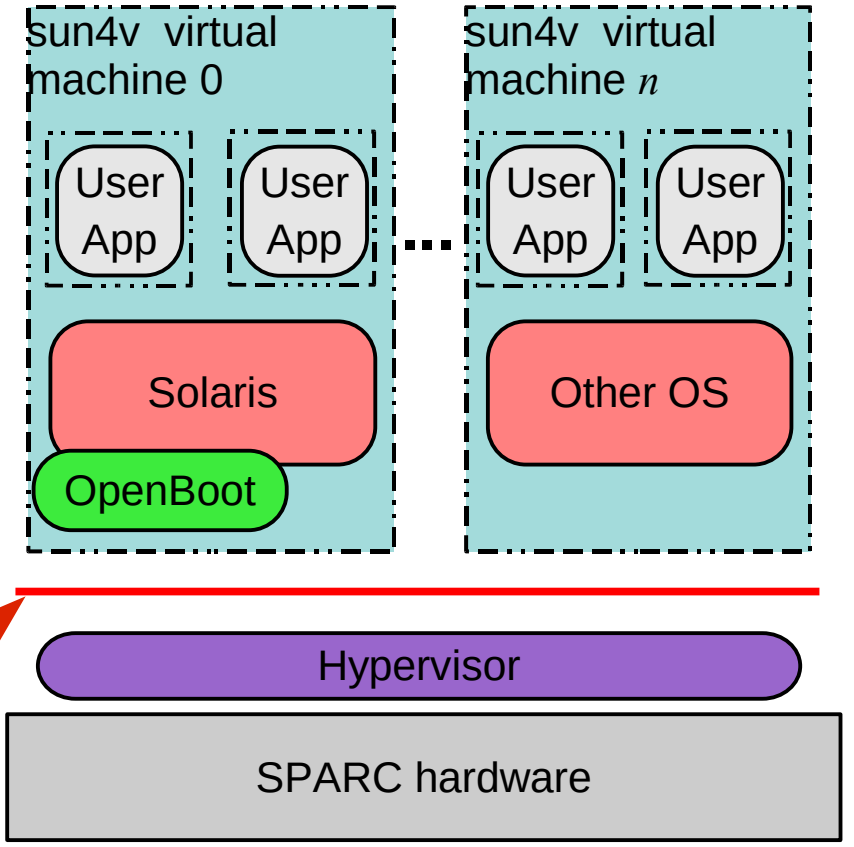
- Full-system simulator for firmware and software development
- Implement enough architecture state to boot up Solaris
- Share the same disk image and binary files (Hypervisor/OBP/reset/etc) with SAM
- Startup script run_legion.sh
- Available configuration: 1-thread, 2-thread, 32-thread

```
> run_legion.sh 1/2/32  [options]
```

Virtual Machine for SPARC

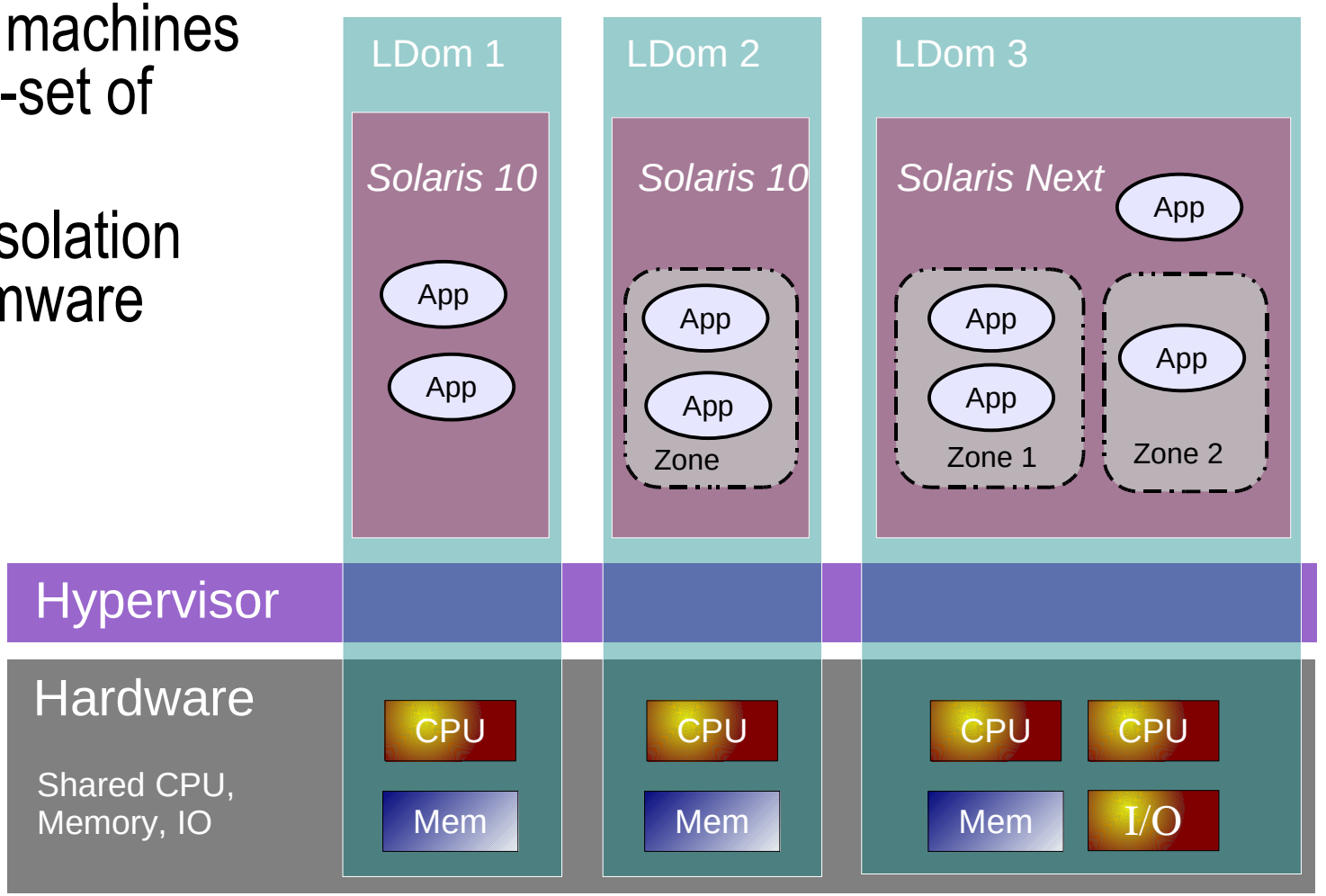
- Thin software layer between OS and platform hardware
- Para-virtualised OS
- Hypervisor + sun4v interface
  - Virtualises machine HW and isolates OS from register-level
  - Delivered with platform not OS
  - Not itself an OS

stable interface "**sun4v**"



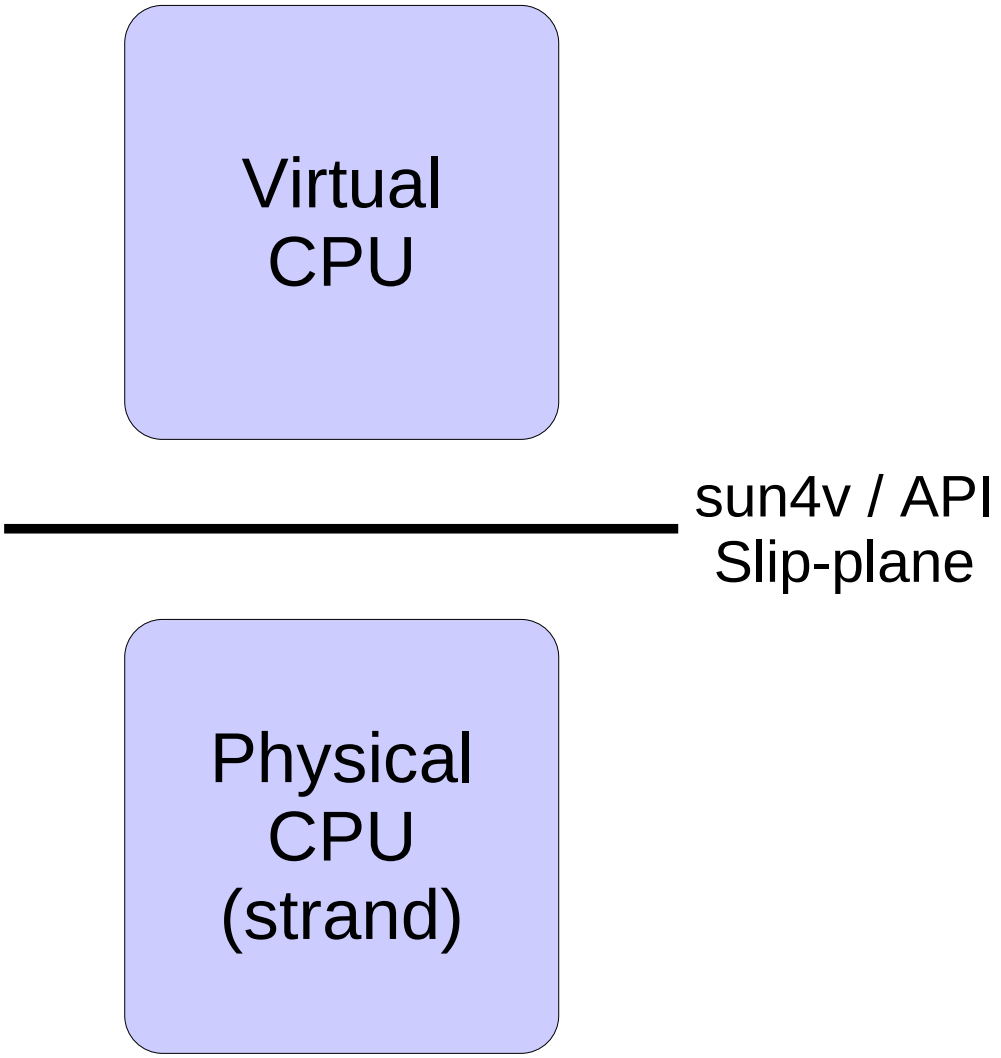
Logical Domains

- Partitioning capability
  - > Create virtual machines each with sub-set of resources
  - > Protection & Isolation using HW+firmware combination



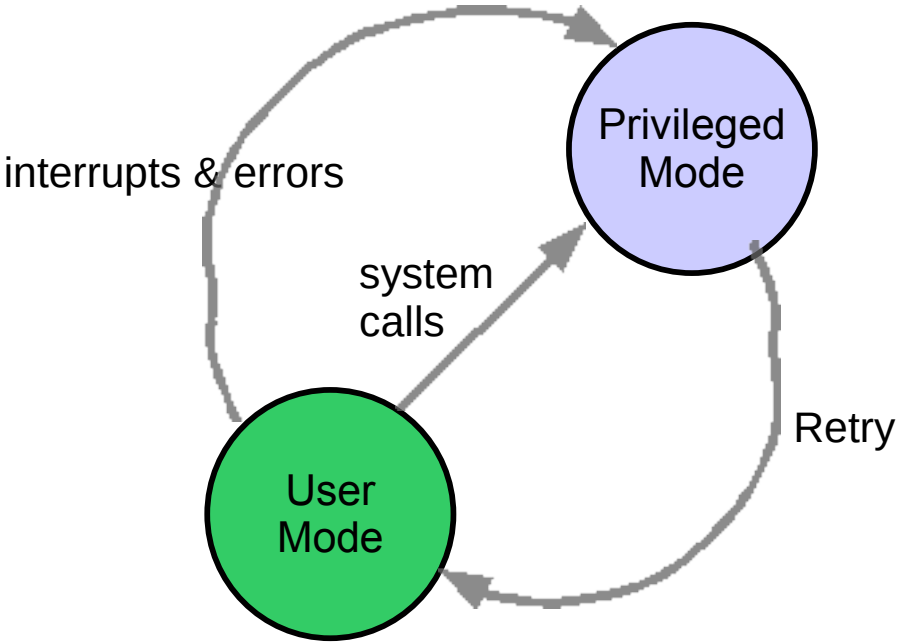
Basic Principles

- Ability to rebind virtual resources to physical components at any time
- Minimal state held in Hypervisor to describe guest OS
- *Never* trust Guest OS

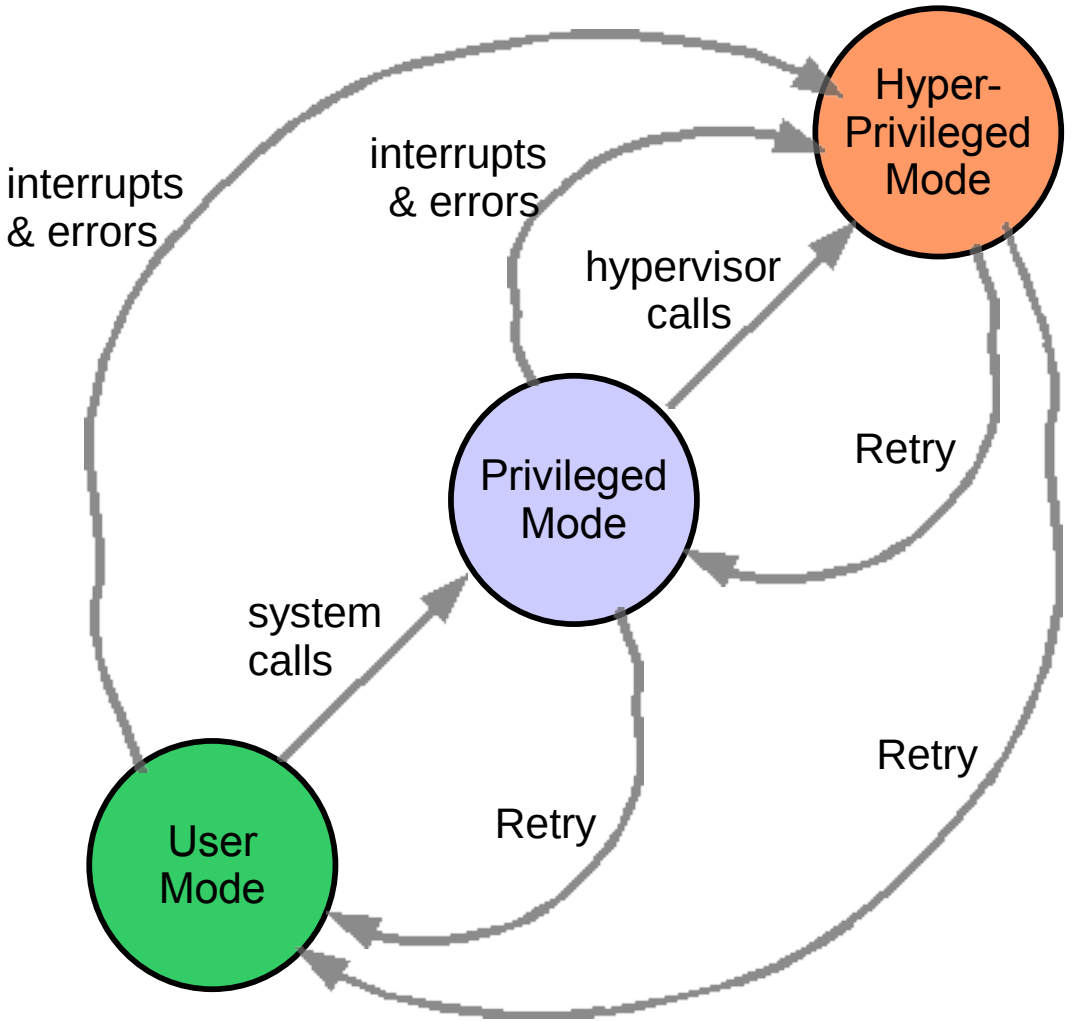


Legacy SPARC execution mode

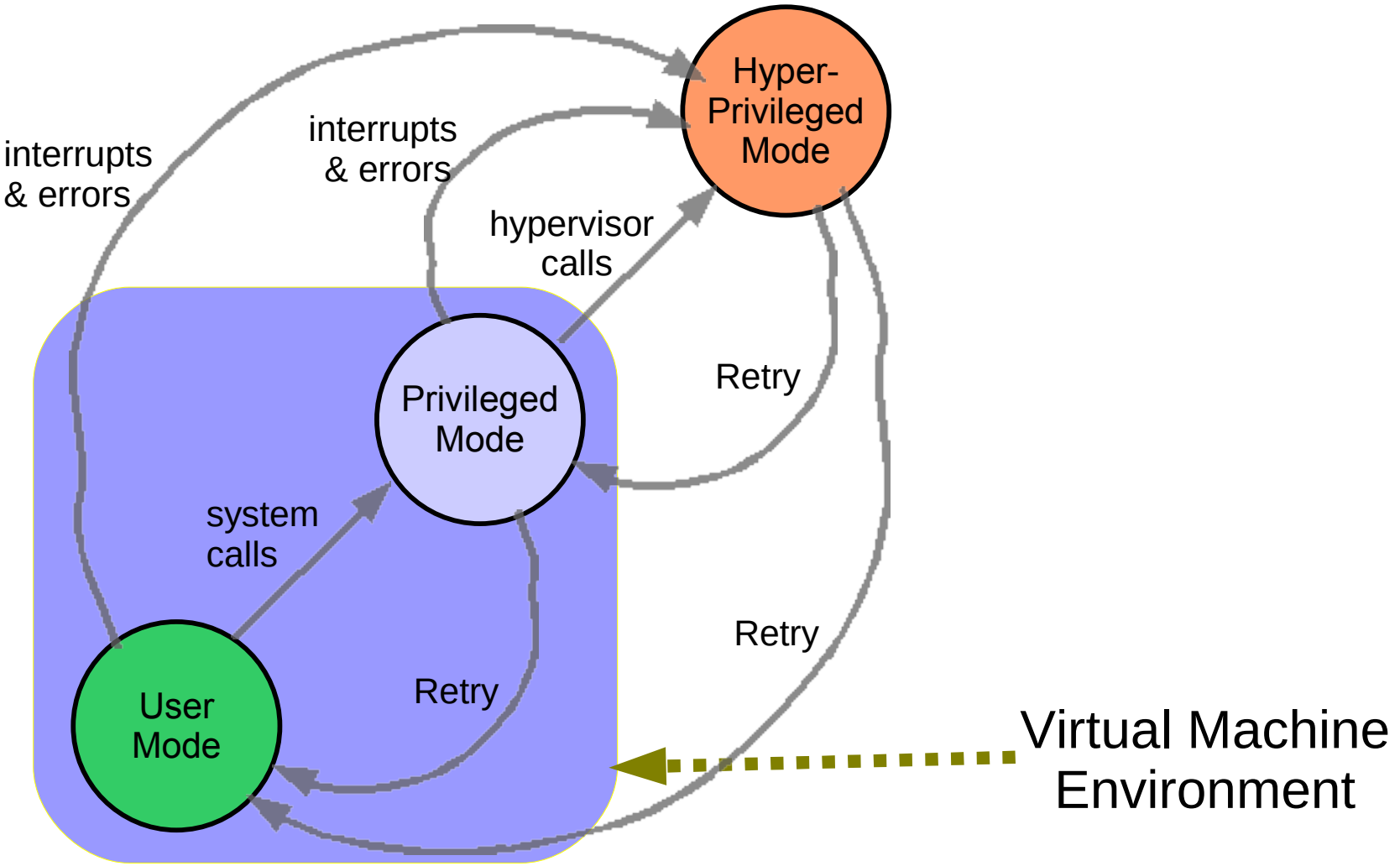
- Existing sun4u chips



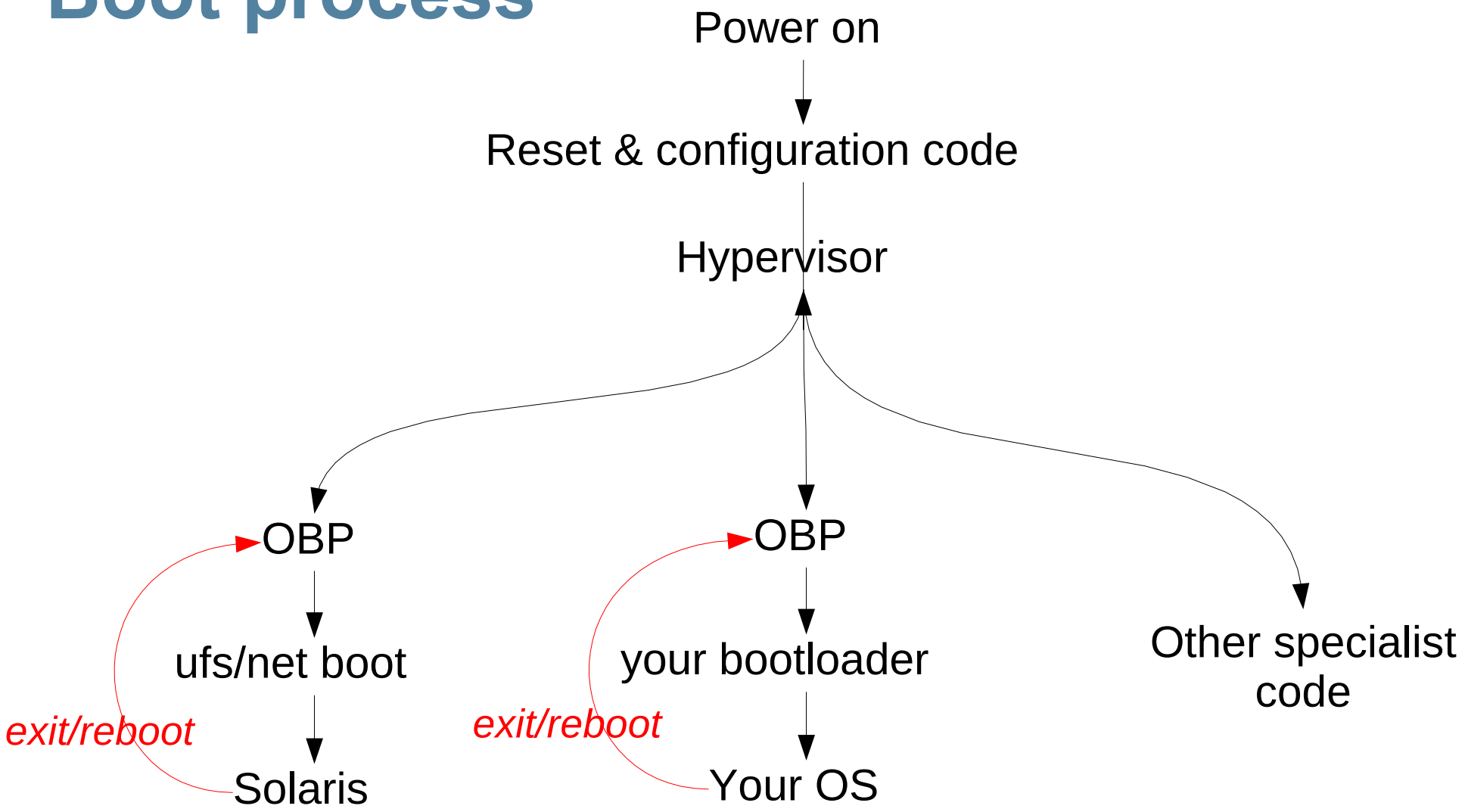
New SPARC Execution mode



New SPARC Execution mode



Boot process



Synergies Within a System

- Hardware
  - > Adequate cache/memory, I/O, and networking bandwidth, plus RAS for large, parallel workloads
- Operating System
  - > Reliable and scalable OS for optimal management of parallel threads
- Developer Tools
  - > Compilers and tools to make application development easy and efficient

Focus of this section:
C/C++/Fortran Compilers & Tools
Free download: cooltools.sunsource.net

OpenSPARC in Academia

University Programs

- Sun supports/encourages academic use of OpenSPARC
  - Collaborations
  - Centers of Excellence (CoE)
    - First OpenSPARC CoE announced Feb 2007
    - more to come
  - technology access, greater equipment discounts, equipment grants, publicity and prestige that aids in obtaining other grants, other support

- For university program info, contact:
  `David.Weaver@sun.com`

OpenSPARC Curriculum Options

- Computer Architecture, micro-architecture
- VLSI design (frontend, physical design), lab work
- Performance studies
- Chip Multi-threaded programming
- OS port (OpenSolaris, Linux, FreeBSD, NetBSD, OpenBSD)
- FPGA implementation
- Embedded systems
- Developer tools for the community and use of Cool Tools
- Blend of teaching class, academic and research projects

Call for Action

- Participate in OpenSPARC community
  - Download, Innovate, Contribute

    http://OpenSPARC.net

- Academia: apply for OpenSPARC University collaboration or Center of Excellence programs

OpenSPARC participation

- Community Registration:
  - http://www.sunsource.net/servlets/Join After registration and confirming password, you can join the mailing lists: http://www.sunsource.net.servlets/ProjectMailingListsList
- Forums:
  - http://forum.java.sun.com/category.jspa?categoryID=120 (separate registration required for posting)

OpenSPARC participation

- Add your university (or company) to the marketplace: http://www.opensparc.net/community-marketplace/
- Send us your profile and we'll post it: http://www.opensparc.net/profiles/
- Add yourself to our Frappr!!: http://wwwopensparc.net/frappr.html
- Contribute to our OpenSPARC Book: http://wiki.opensparc.net/bin/view.pl/Main/Webhome (separate registration required for editing)

Thank you

Dwayne.Lee@sun.com
Shrenik.Mehta@sun.com

www.opensparc.net