What will you open?

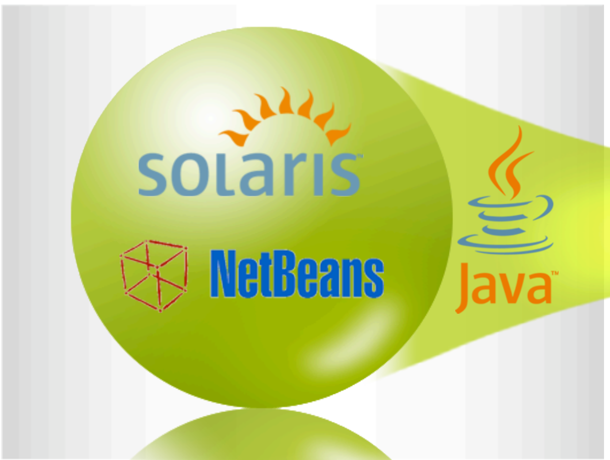
Highly Threaded SPARC Architectures

James Hughes
Sun Fellow
Sun Microsystems

Note: future dates and features within this presentation reflect Sun's current plans but may change without notice.

Driving Volume to Value

Open Source
Development Choice



- Solaris
- NetBeans
- Glassfish
- Single sign-on
- All Middleware

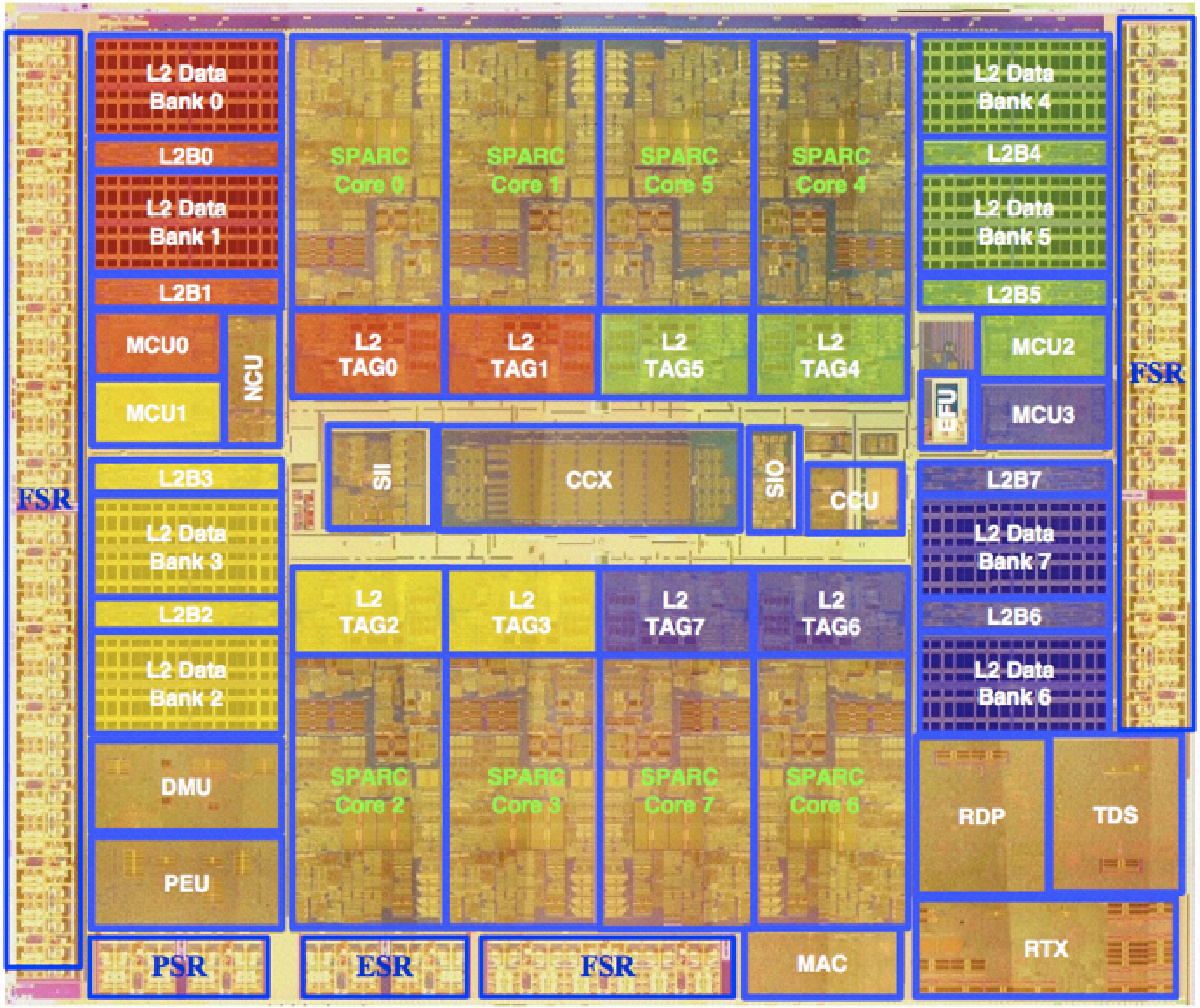
Free Access
Increases Volume

- Free RTU (unsupported)
- Extended Try and Buy Program
- Connected to Sun

Revenue Making SW
Business Deployment

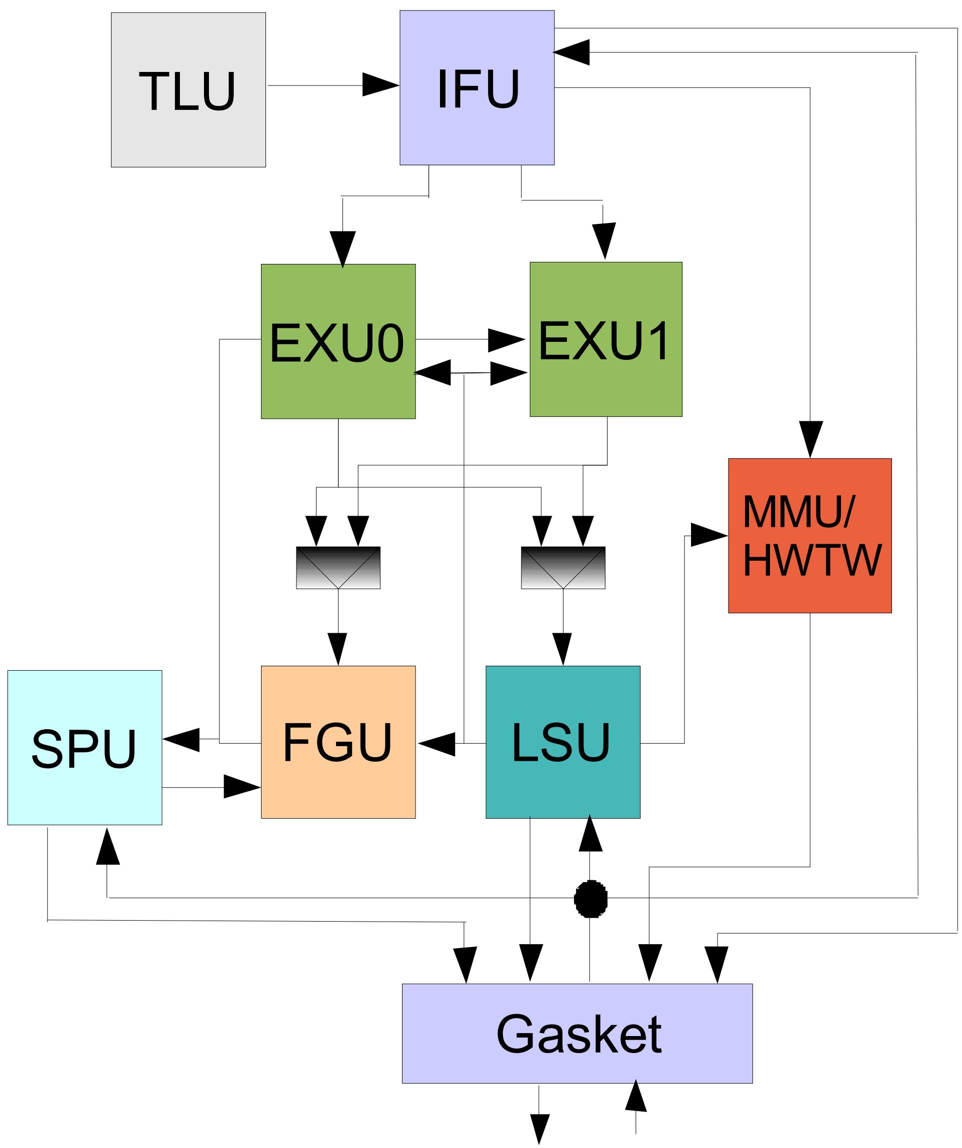
- Solaris 10
- Java ES
- ID Management
- Java CAPS
- Mobile

Niagara2 Chip Overview



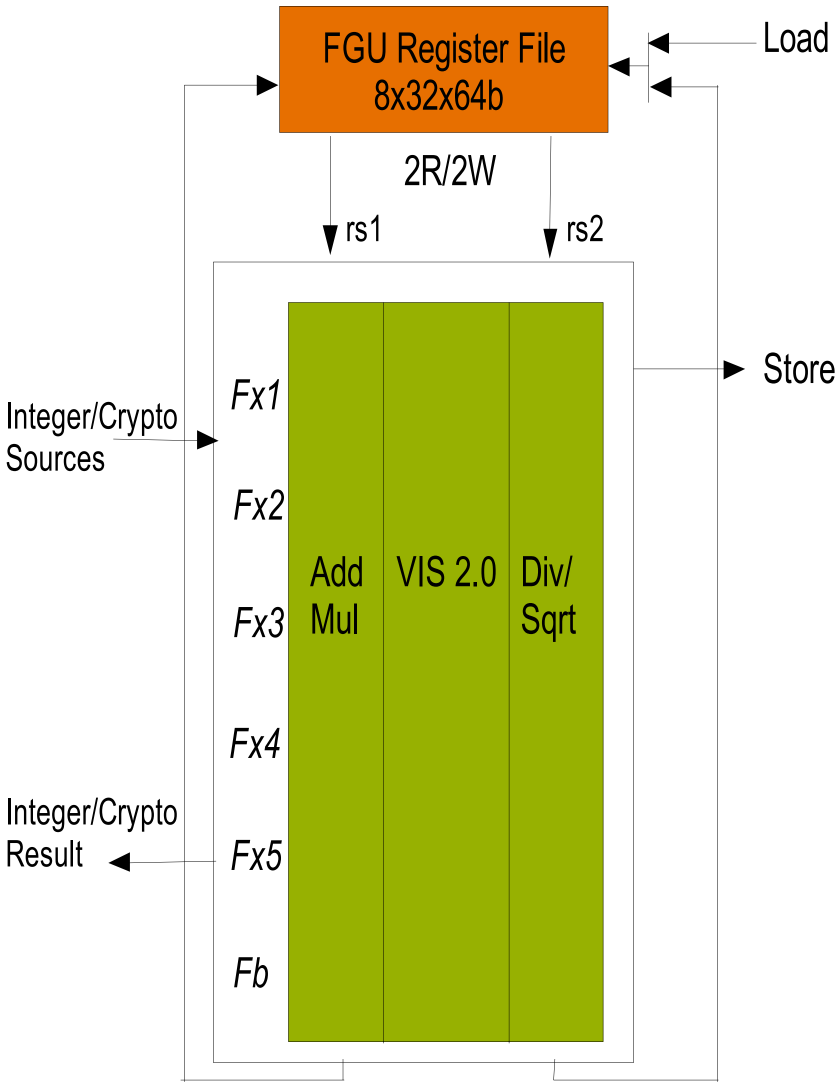
- 8 Sparc cores, 8 threads each
- Shared 4MB L2, 8-banks, 16-way associative
- Four dual-channel FBDIMM memory controllers
- Two 10/1 Gb Enet ports
- One PCI-Express x8 1.0A port
- 342 mm^2 die size in 65 nm
- 711 signal I/O, 1831 total

Sparc Core Block Diagram



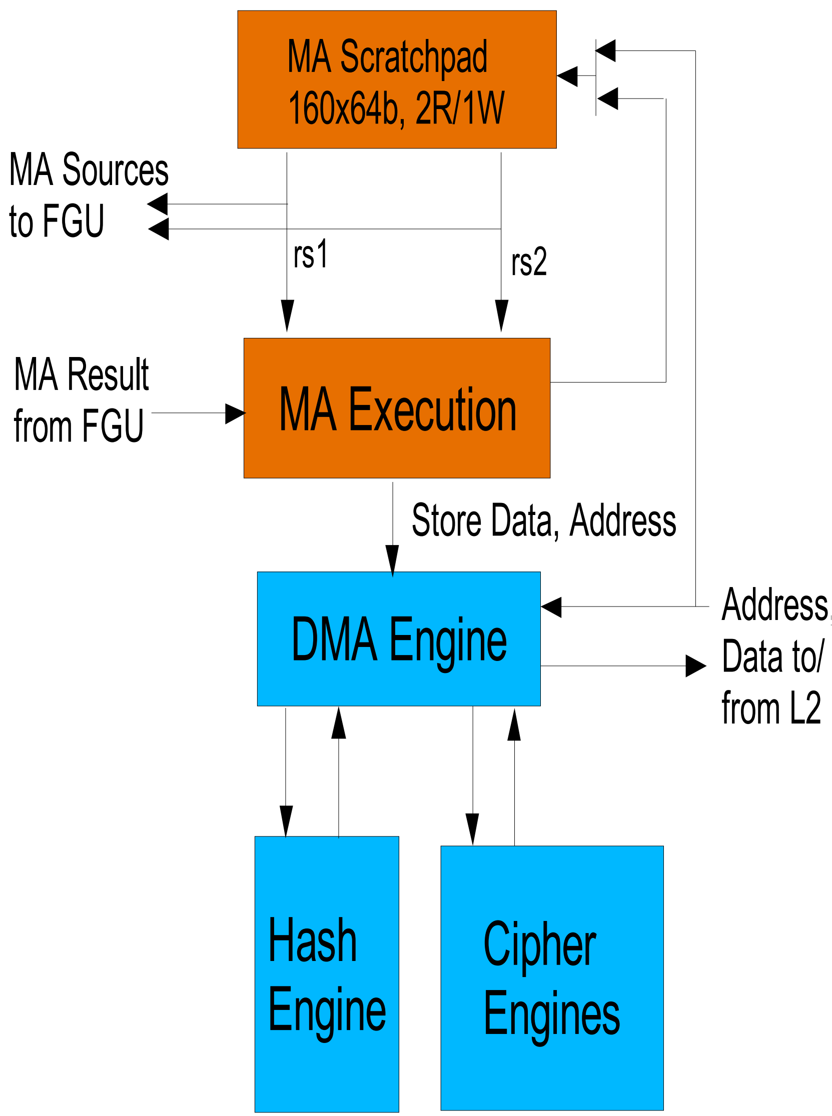
- IFU – Instruction Fetch Unit
  - 16 KB I$, 32B lines, 8-way SA
  - 64-entry fully-associative ITLB
- EXU0/1 – Integer Execution Units
  - 4 threads share each unit
  - Executes one integer instruction/cycle
- LSU – Load/Store Unit
  - 8KB D$, 16B lines, 4-way SA
  - 128-entry fully-associative DTLB
- FGU – Floating/Graphics Unit
- SPU – Stream Processing Unit
  - Cryptographic acceleration
- TLU – Trap Logic Unit
  - Updates machine state, handles exceptions and interrupts
- MMU – Memory Management Unit
  - Hardware tablewalk (HWTW)
  - 8KB, 64KB, 4MB, 256MB pages

FGU



- Fully-pipelined (except divide/sqrt)
  - Divide/sqrt in parallel with add or multiply operations of other threads
- FGU performs integer multiply, divide, population count
- Multiplier enhancements for modular arithmetic operations
  - Built-in accumulator
  - XOR multiply

SPU



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

Crypto Unit

- Symmetric Modes
  - Counter
  - CBC
  - CCM
- Keys stored in unit
  - Volatile
  - Operation of AES unit does not pollute cache
    - No side channel attack
- Random number generator in hardware
- Transparent to software
  - Solaris Crypto Framework

Victoria Falls: Supercomputer in 4U

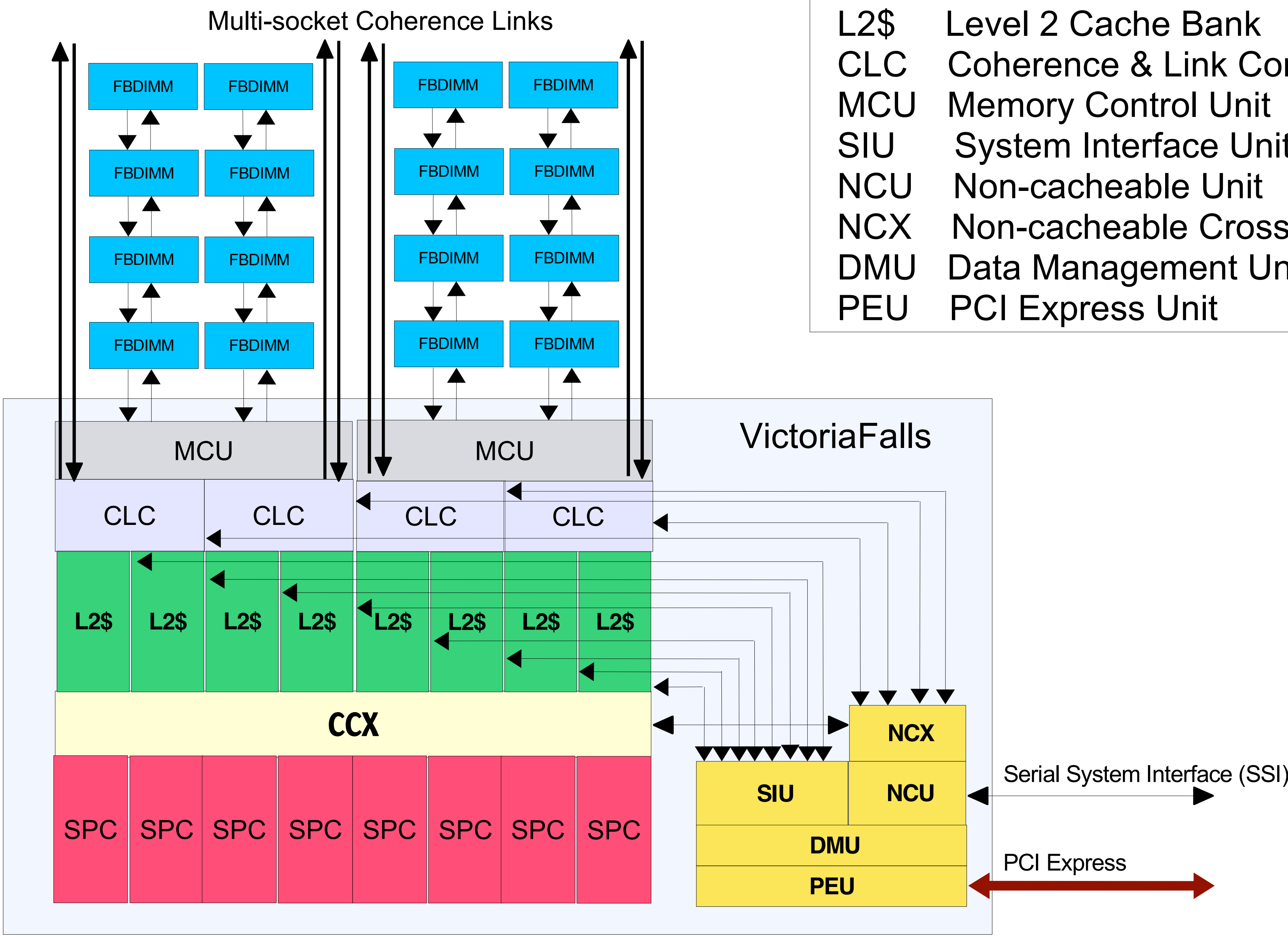
- 256 hardware threads
- 256GB shared RAM
- 139GB/s Bisection BW

VictoriaFalls Chip Multiprocessor

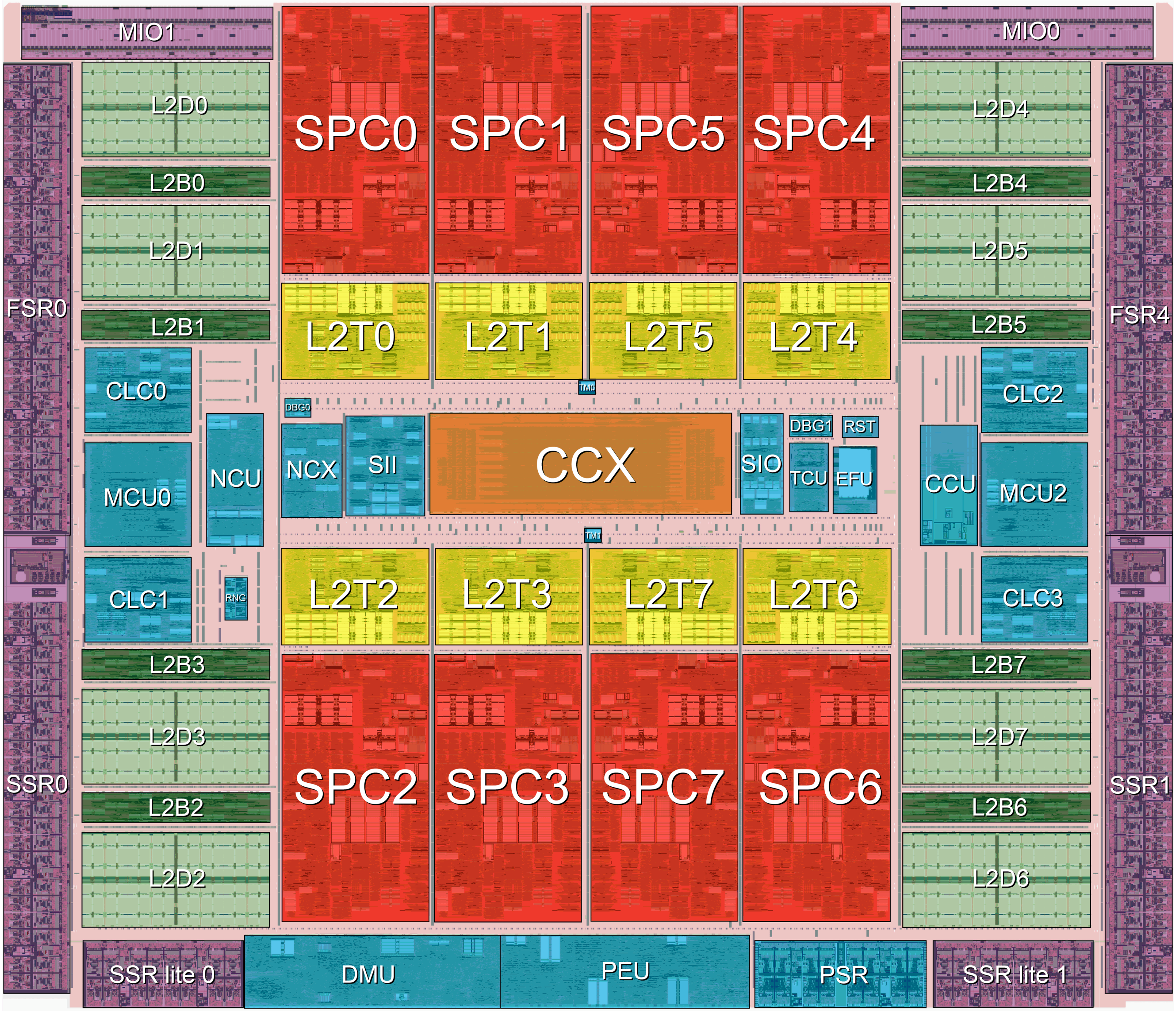
- 8 Core CMP with 8 Strands per Core @ 1.4Ghz
- Niagara2 SPARCv9 Core
  - 2 x 8-stage Integer Units (4 strands per pipe) – Single Issue per Pipe
  - 12-stage Pipelined FGU (except divide/sqrt)
  - Integrated Crypto Accelerator
  - 16KB 8-way SA L1-I$, 32B Lines, Write-through
  - 8KB 4-way SA L1-D$, 16B Lines, Write-through
  - 64 Entry Fully Associative I-TLB
  - 128 Entry Fully Associative D-TLB
- 4 MB Shared L2$
- 2 Dual-Channel FBDIMM Memory Controllers
- Integrated PCI Express I/O Bridge
- Multi-socket Coherence Links

High Level Block Diagram

SPC	SPARCv9 CMT Core
CCX	Core-Cache Crossbar
L2$	Level 2 Cache Bank
CLC	Coherence & Link Controller
MCU	Memory Control Unit
SIU	System Interface Unit
NCU	Non-cacheable Unit
NCX	Non-cacheable Crossbar
DMU	Data Management Unit
PEU	PCI Express Unit



VictoriaFalls Micrograph



65nm CMOS

11 Metal Layers

709 Signal I/O

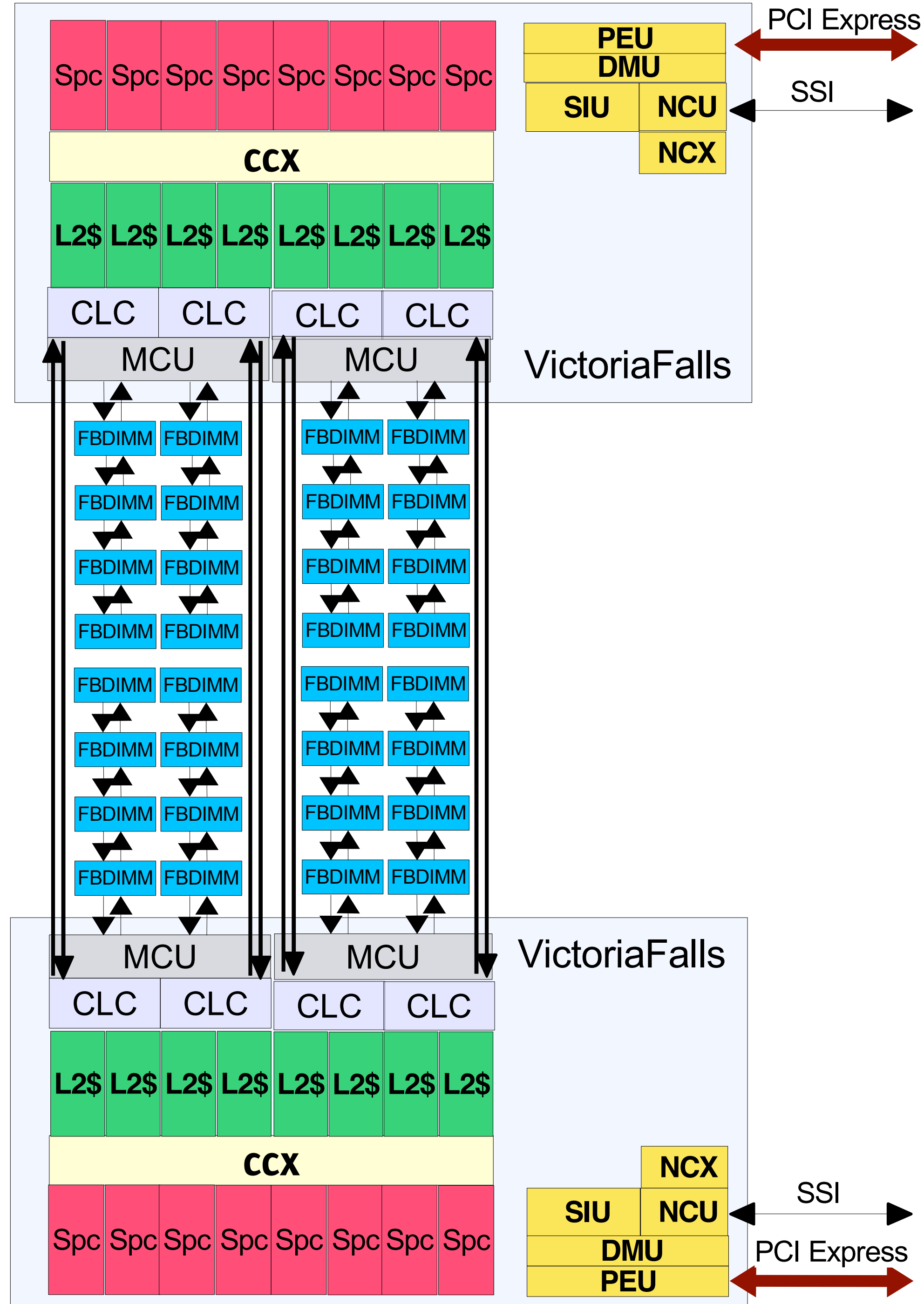
1831 Total I/O

~Niagara2 CMP Power

~Niagara2 CMP Die Area

Dual-socket Architecture

- 16 SPARC Cores
  - 128 Threads
- Coherent Interconnect
  - 4 Coherence Links, Full-Duplex
  - 65GB/sec Raw Bisection
- 8 FBDIMM Channels
  - 42GB/sec (Theoretical Peak) Read
  - 21GB/sec (Theoretical Peak) Write
  - DDR2-667
- Integrated I/O Bridges
  - 2 x8 Lane PCI Express Ports @ 2.5GT/sec per Lane Full Duplex
  - 1024 Concurrent IO Address Translations (Virtual-to-Real, Real-to-Physical)
  - Relaxed DMA Ordering



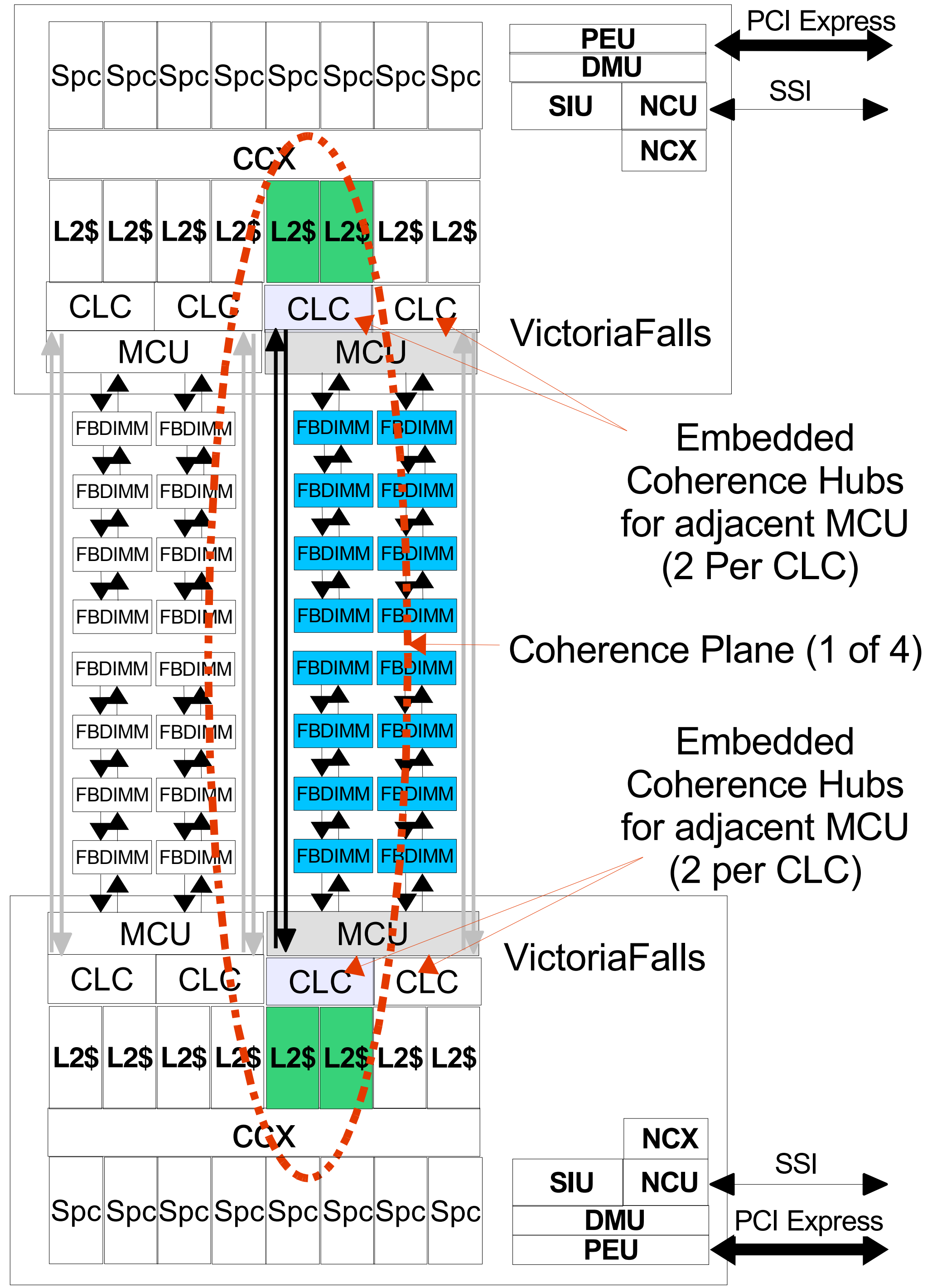
3rd Level Scaling: Memory & Multi-chip

- 4 Coherence Planes
  > Address Space Partitioned: PA<8:7>

  > Coherence Planes Operate Independent of One Another
- Cache Coherence
  > MOESI States in L2$

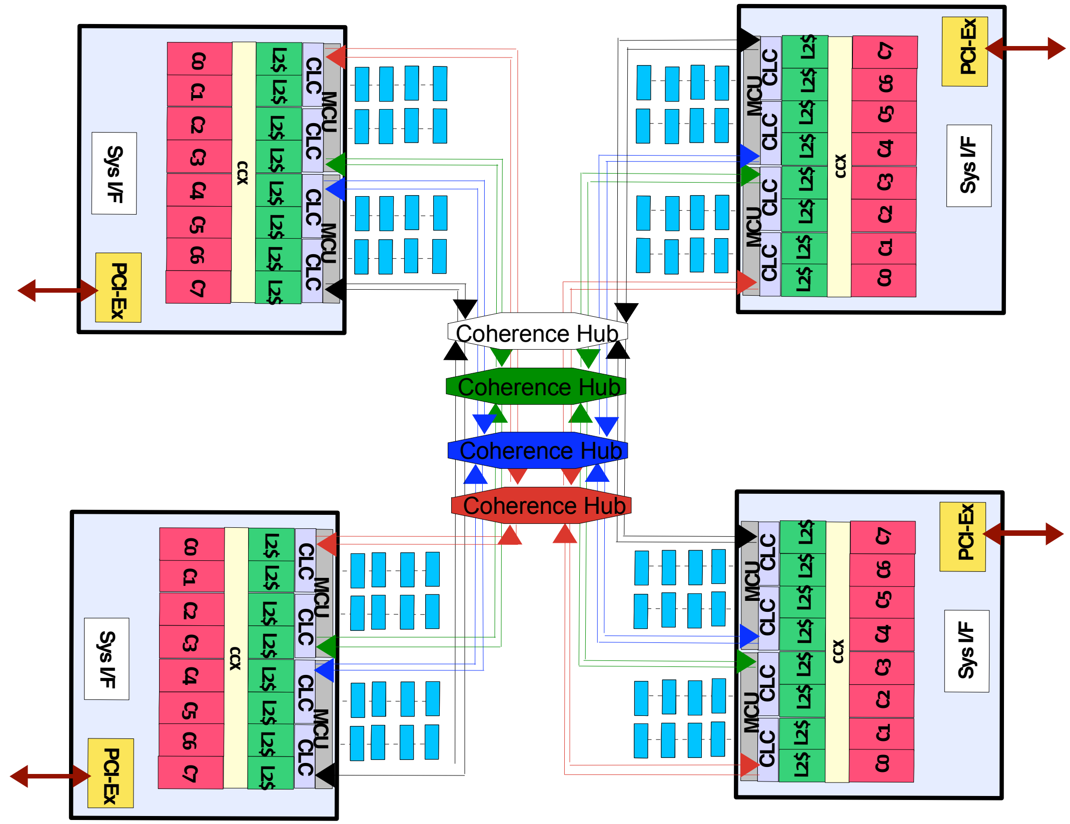
  > C2C Transfers on snoop hits to M, O, E and S (MCU Node) States
- Ordering & Conflict Mgmt
  > PA Conflicts are Serialized by the Coherence Hub (Linked List)

  > Requesting Node receives Serialization ACK and 1 Snoop response/WB-ack from Coherence Hub
- Up to 1600 Msnoops/sec (Link Protocol Limit)



Quad-socket Architecture

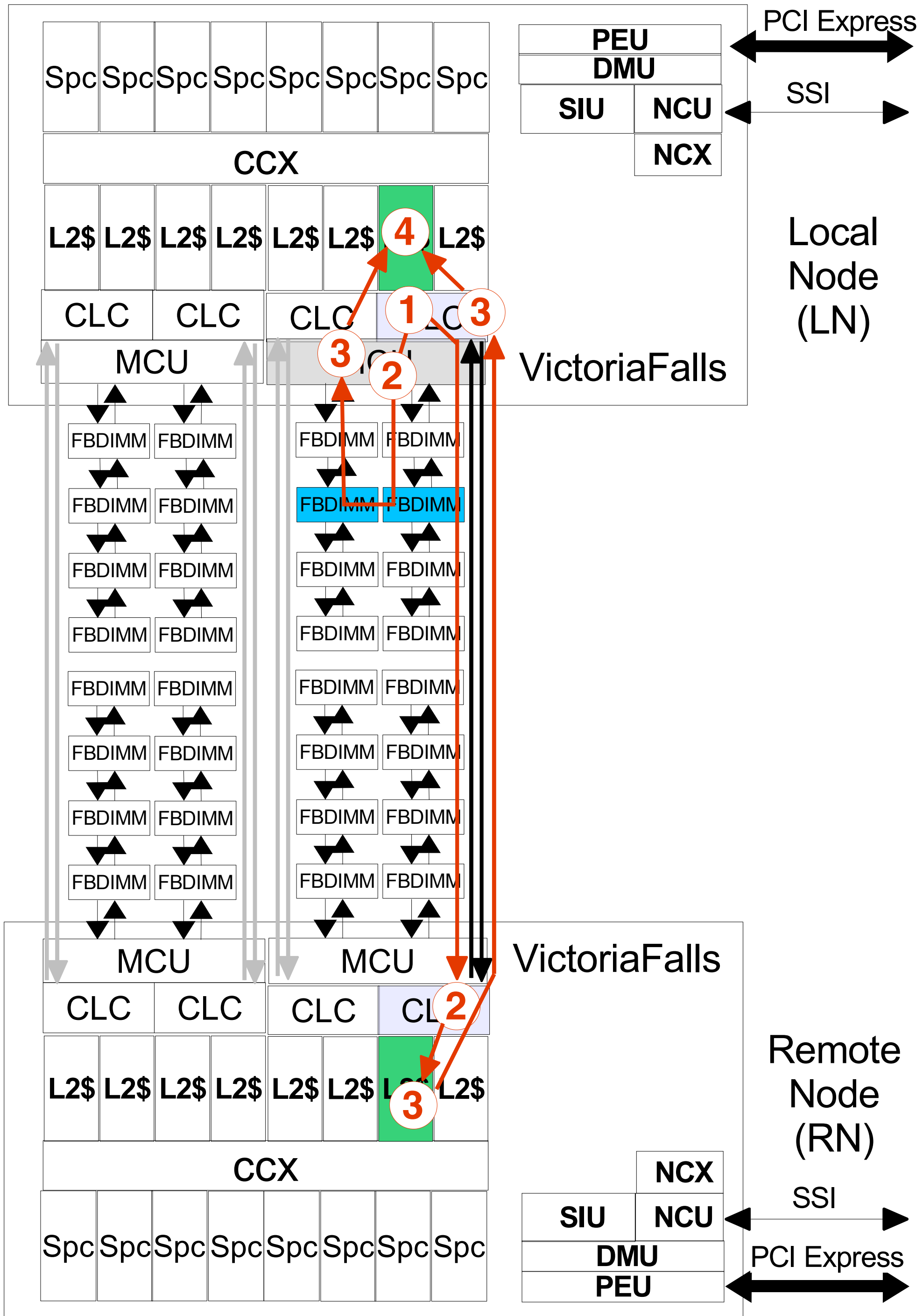
- 32 SPARC Cores
  - > 256 Threads
- Coherent Interconnect
  - > 4x4 Coherence Links, FD
  - > 130GB/sec Raw Bisection
- 16 FBDIMM Channels
  - > 84GB/sec (Theoretical Peak) Read
  - > 42GB/sec (Theoretical Peak) Write
  - > DDR2-667
- Integrated I/O Bridges
  - > 4 x8 Lane PCI Express Ports @ 2.5GT/sec per Lane FD
  - > 2048 Concurrent I/O Address Translations (Virtual-to-Real, Real-to-Physical)
  - > Relaxed DMA Ordering



- External Coherence Hub
  - > 4-port Arbiter/Switch ASIC
  - > 1 Device per Coherence Plane

Dual-socket Local Memory Coherent Read

1. LN Coherence Hub CAMs against read, invalidate and writeback transaction types. LN CLC forwards non-conflicting request-acknowledgement to LN L2$ bank, then read request to LN MCU & snoop request to RN CLC.
2. LN MCU FBDIMM access. RN CLC forwards snoop request to corresponding L2$ bank and CAMs RN CLC writeback buffer.
3. RN L2$ bank performs snoop operation, schedules required multicast invalidations to upstream caches, returns snoop response status and L2$ copyback data (if required) to LN CLC
4. LN CLC forwards snoop response to LN L2$ bank, forwards memory and C2C return data (if required) to LN L2$ bank and retires transaction. L2$ bypasses resolved return data to upstream cache and fills allocated L2$ data entry.



Dual-socket Remote Memory Coherent Read

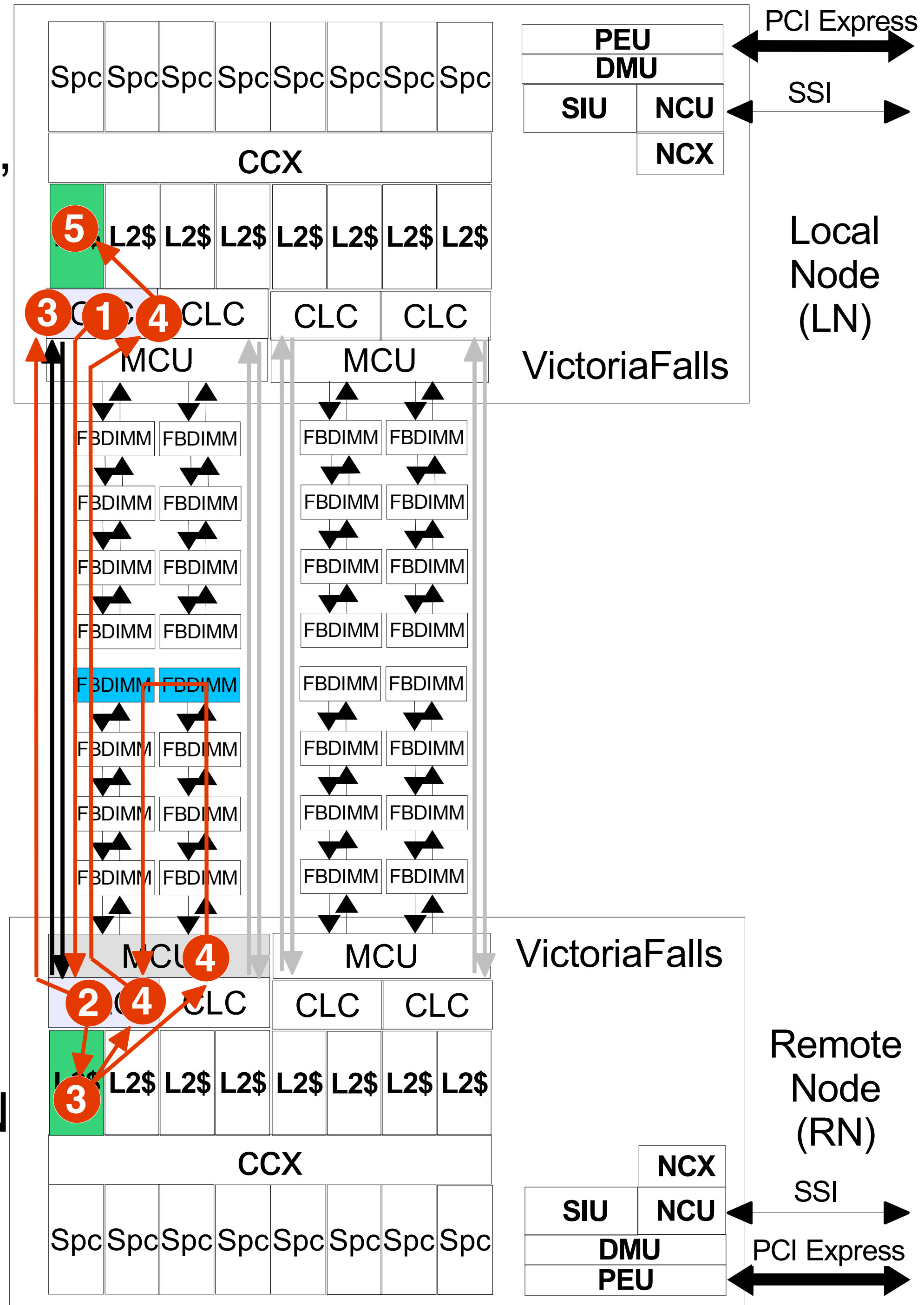
1. LN CLC forwards read request to RN CLC.

2. RN Coherence Hub CAMs against read, invalidate and writeback transaction types. RN CLC forwards non-conflicting request-acknowledgement to LN CLC, snoop request to RN L2$ bank.

3. LN CLC forwards read-acknowledgement to LN L2$ bank. RN L2$ bank perfoms snoop operation, schedules required multicast invalidations to upstream caches and returns snoop status result to RN CLC to activate read request to RN MCU.

4. LN MCU FBDIMM access if no L2$ copyback. RN CLC forwards snoop response and memory or copyback data to LN CLC.

5. LN CLC forwards snoop response to LN L2$ bank, forwards memory/C2C return data to local L2$ bank and retires transaction. L2$ forwards/bypasses resolved return data to upstream cache



Quad-socket Remote Memory Coherent Read

1. LN CLC forwards read request directly to ECH, bypassing internal coherence hub.

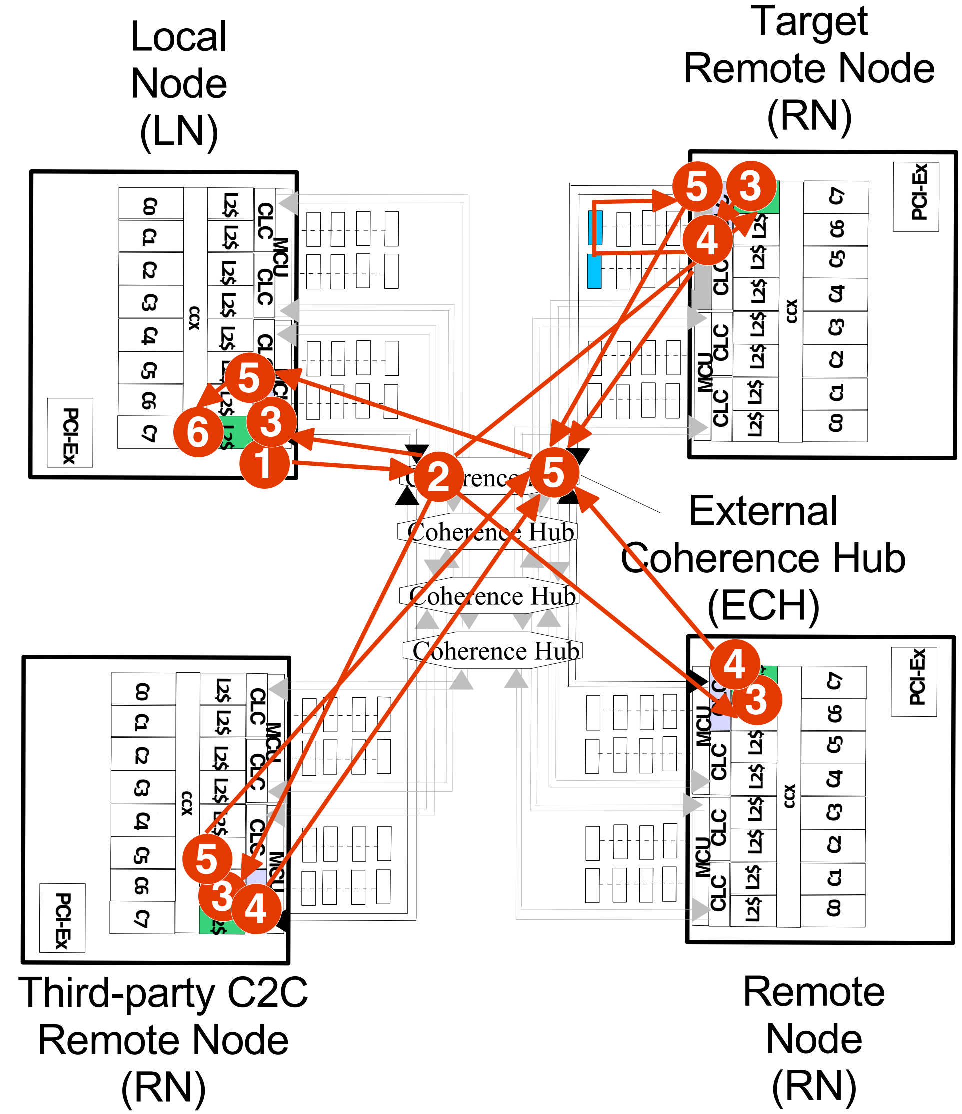
2. ECH CAMs against read, invalidate and writeback transaction types. Non-conflicting request-acknowledgement sent from ECH to LN CLC and snoop requests to RN CLC/L2$ banks.

3. LN CLC forwards request-acknowledgement to LN L2$ bank. Each RN CLC snoops Writeback buffer. RN L2$ banks perform snoop operation, schedule required multicast invalidations to upstream caches and return snoop status result to ECH via respective RN CLC. RN L2 bank performs copyback operation if required. Target RN CLC activates read request to target RN MCU.

4. Target RN MCU FBDIMM access if no L2$ copyback by Target RN. Each RN CLC forwards snoop response to ECH. RN L2 bank copyback data from non-target RN L2 bank sent to ECH.

5. Resolved snoop result sent from ECH to LN CLC. Memory or third-party copyback data sent to LN CLC, filtered by ECH.

6. Local Node CLC forwards snoop response to Local Node L2$ bank, forwards memory/C2C return data to local L2$ bank and retires transaction. L2$ forwards/bypasses resolved return data to upstream cache and fills allocated L2$ data entry.

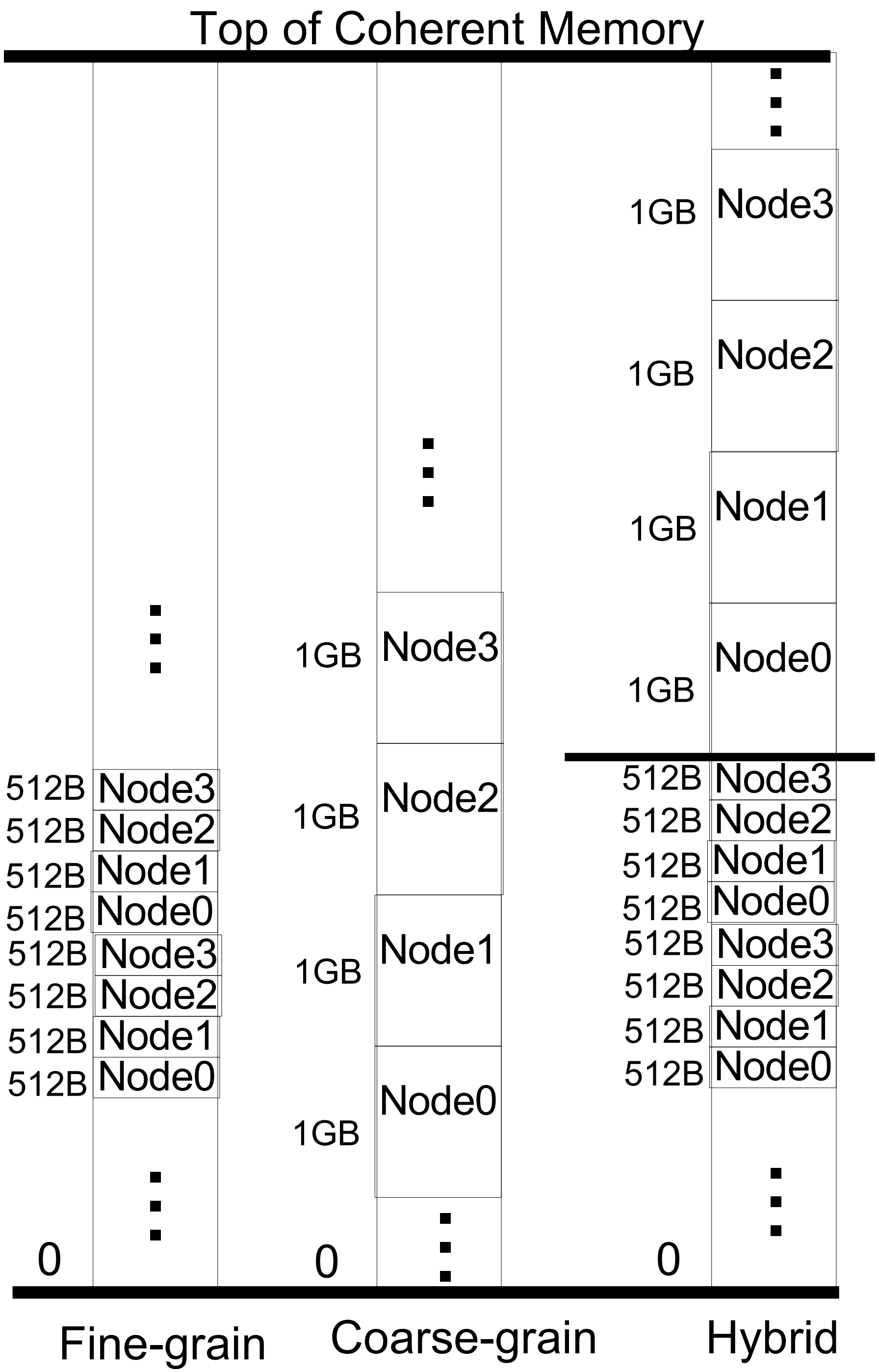


External Coherence Hub (ECH)

- 2X Bisection Bandwidth Increase
- Adapts Coherent Interconnect to Instantaneous Bandwidth Peaks Between Node Pairs
- Internal Coherence Hub Bypassed

Memory Addressing

- 512B and 1GB Interleaving Across Nodes
  - 1GB: NUMA-aware OS Support
  - 512B: Promotes Uniform Access and Reduced Probability of Hotspots
- Programmable Ceiling Mask Establishes 4GB Address Boundary Between Fine and Coarse Grain Interleave
- PA<39:32> of Access is Compared to Ceiling Mask [6:0]. If Greater, Belongs to Coarse Grain Interleave



Memory Placement

Motivation

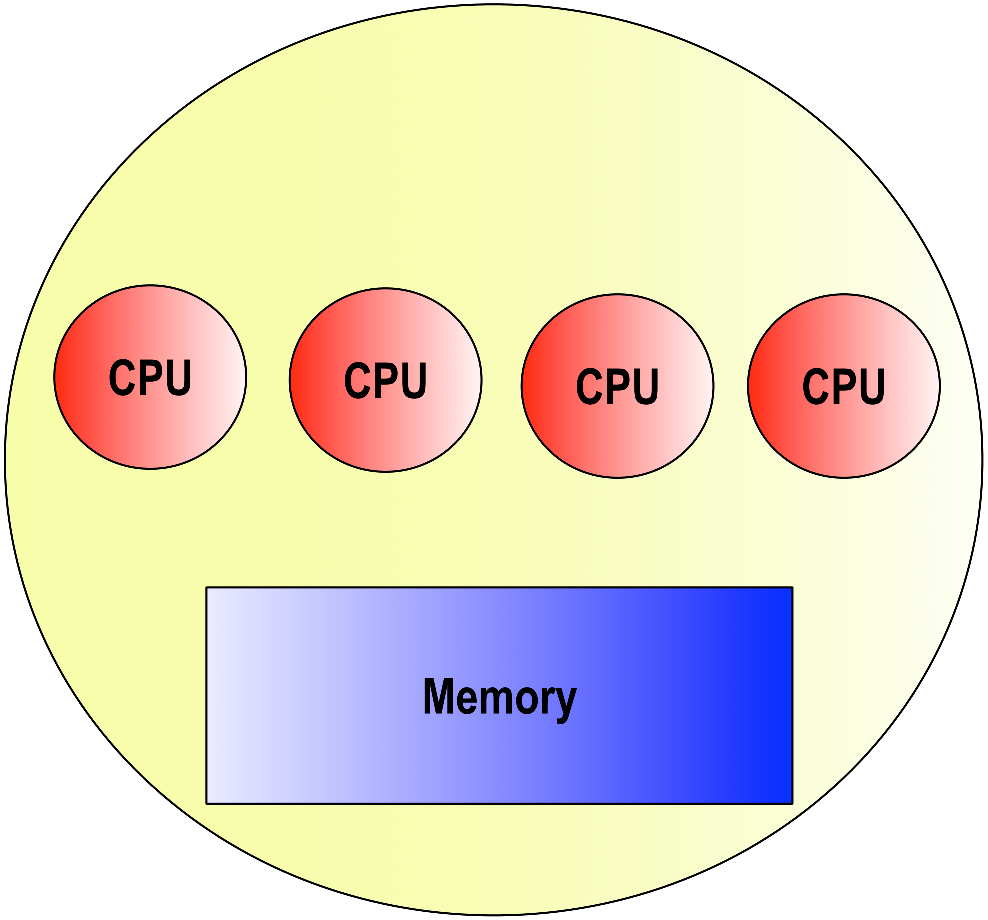
- Non Uniform Memory Access (NUMA) machines,
  - may not perform very well without knowing which CPUs and memory are near each other.
- Memory Placement Optimization
  - Solaris aware of NUMA to provide better performance on NUMA machines by optimizing for locality
- Discovered automatically

Design Goals

- Enhance performance and reproducibility (by default)
- Expose a common framework and means to exploit platform specific features
- Observability tools
- Benchmarks and techniques for evaluating NUMA performance

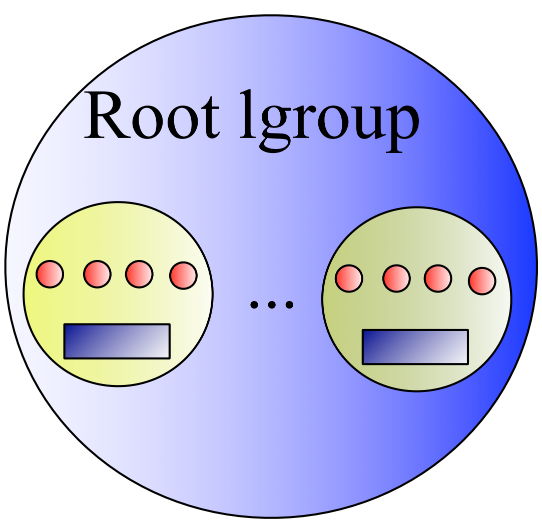
Uniform Memory Access

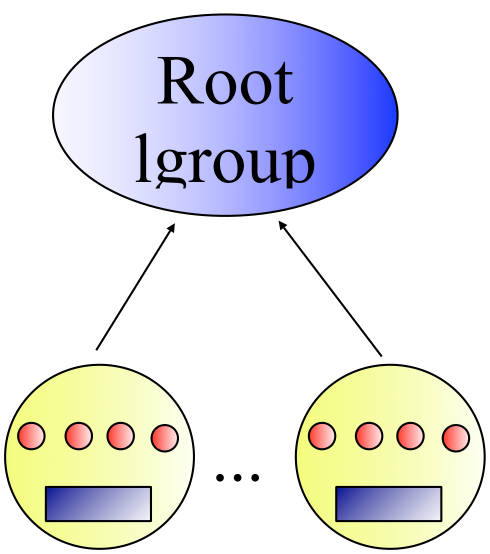
- 1 level of locality
- Same latency between all CPUs and all memory represented by one lgroup



NUMA

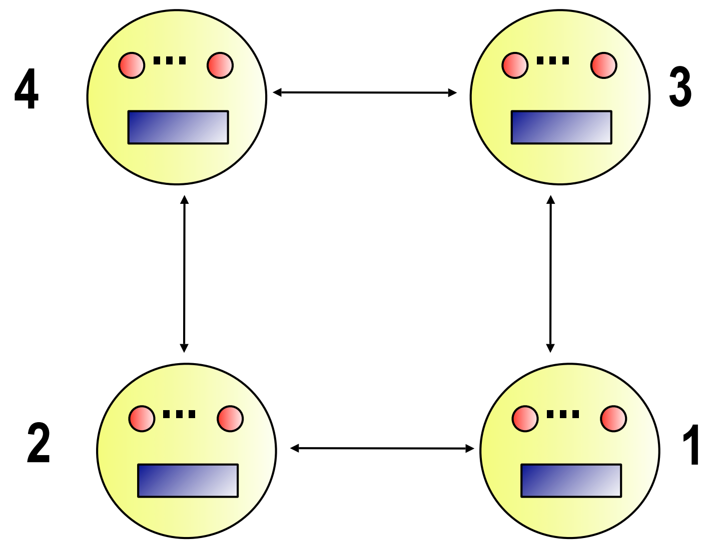
- 2 levels of locality
  - Local and remote memory latency
  - Children lgroups capture CPUs and memory within same local latency of each other
  - Root lgroup contains CPUs and memory within remote latency (eg all CPUs and memory)





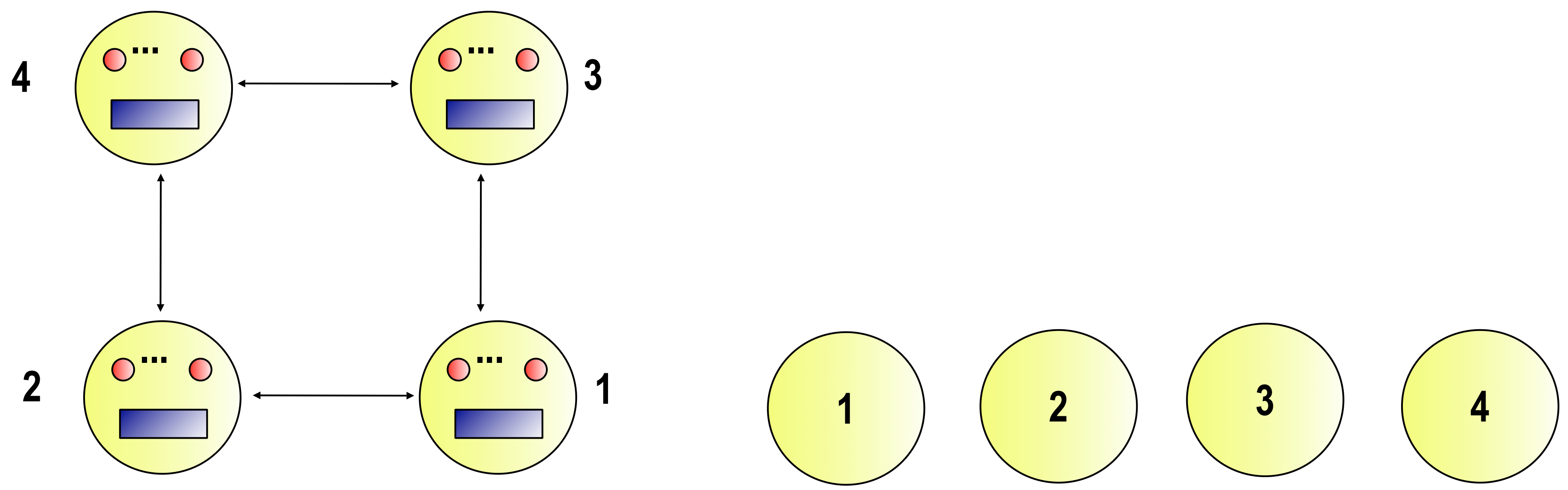
NUMA (ring)

- 3 levels of locality
  - 4 node ring topology
  - Same local latency within each node
  - Remote latency determined by sum of cost for each hop needed to reach memory



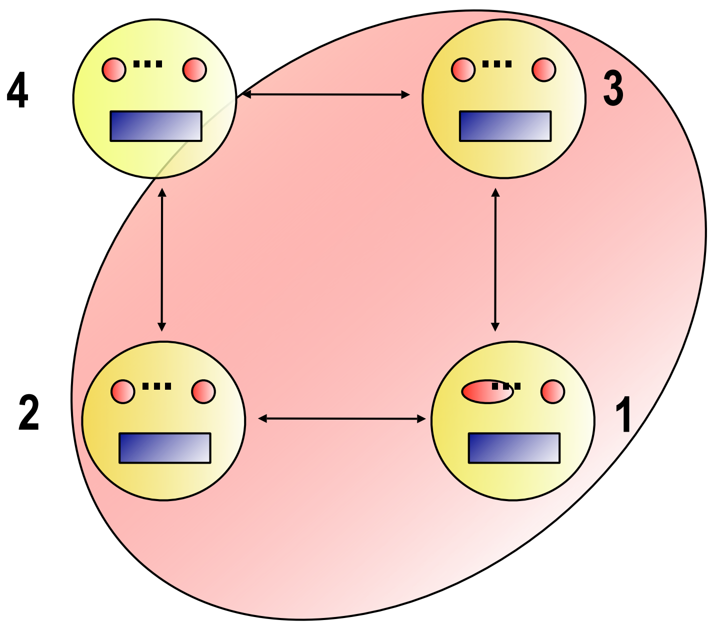
NUMA (ring)

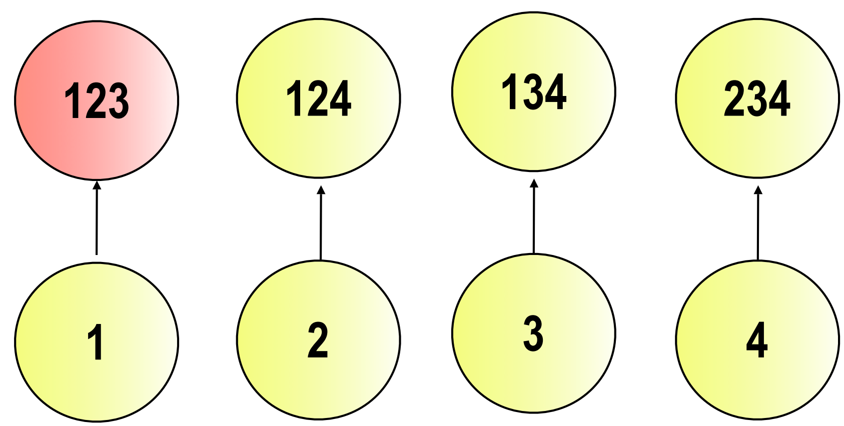
- 1st level of locality
  > Lgroups 1, 2, 3, and 4 containing CPU(s) and memory within some local latency



NUMA (ring)

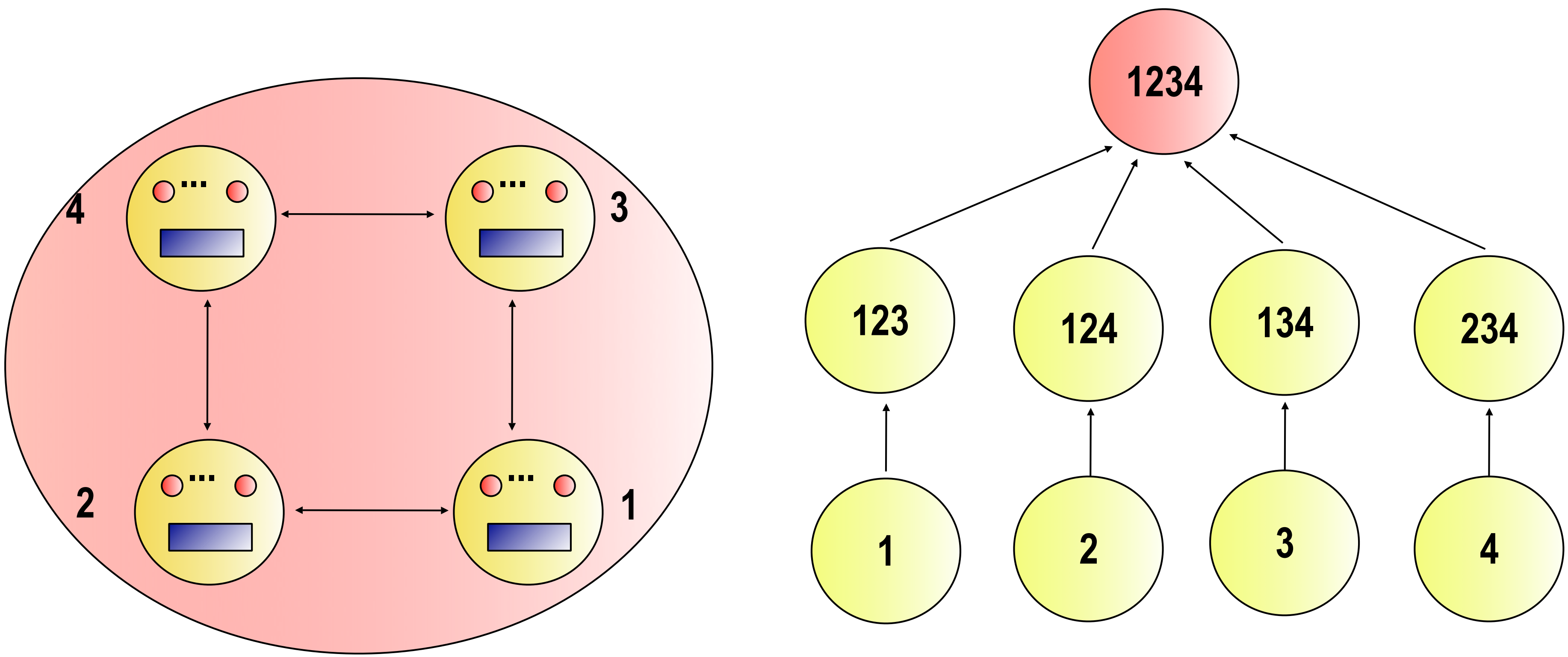
- 2nd level of locality
  - Lgroups 1, 2, 3, 4, and their parent lgroups containing their next nearest resources





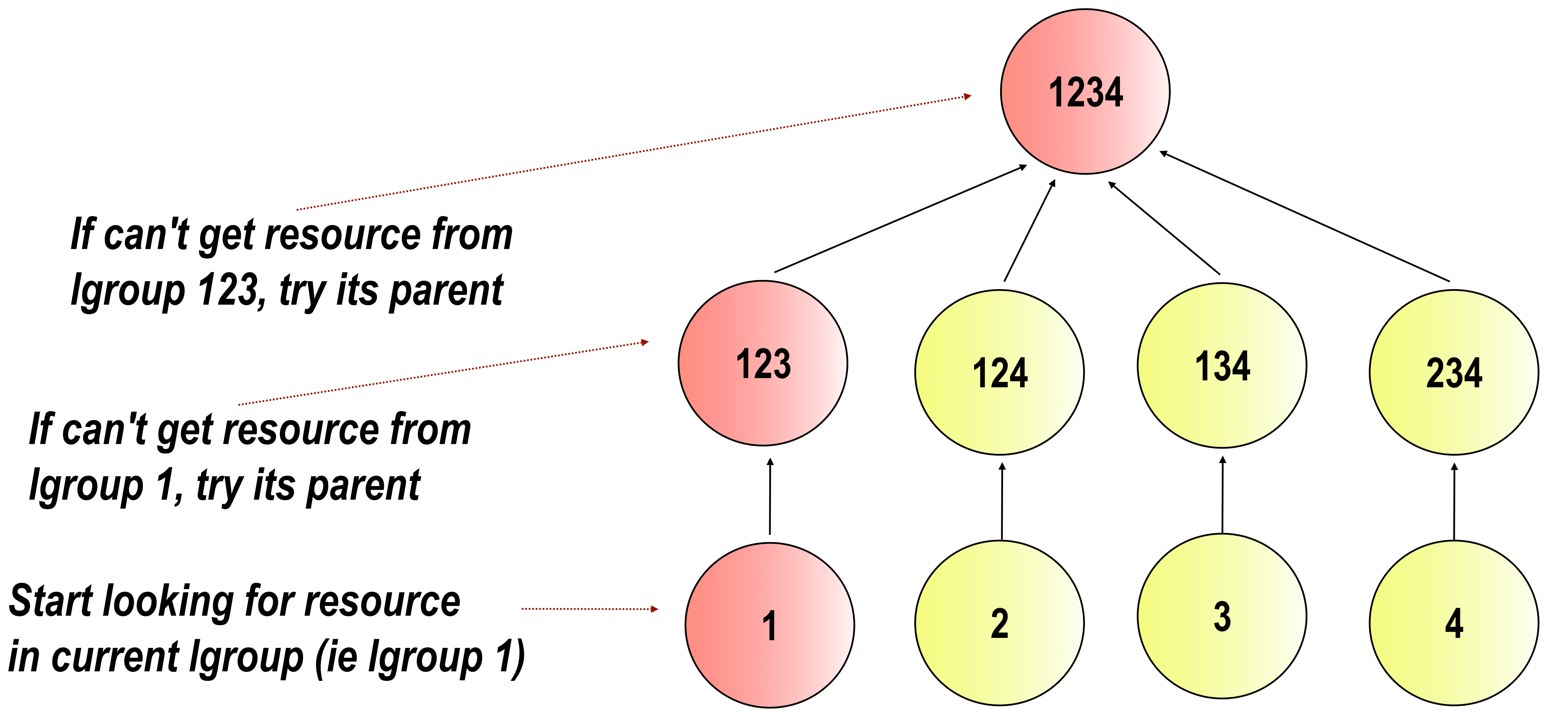
NUMA (ring)

- 3rd level of locality
  - > Root lgroup contains next nearest resources for leaf lgroups' parents and all resources in machine so buck stops here



Finding Resources

- Assume thread is running in lgroup 1 and needs memory



Future?

- New Instructions
- New Locking Paradigms

Specialized Sparc Instructions

- Currently in Sparc
  - Population count
- Possible future instructions
  - Leading zero detector
  - Pipelined sheep-and-goat (form of centrifuge)
  - 64-bit mul-high (getting the upper 64 bits)
  - xor multiplier instruction, aka GF(2) multiply

Transactional Memory

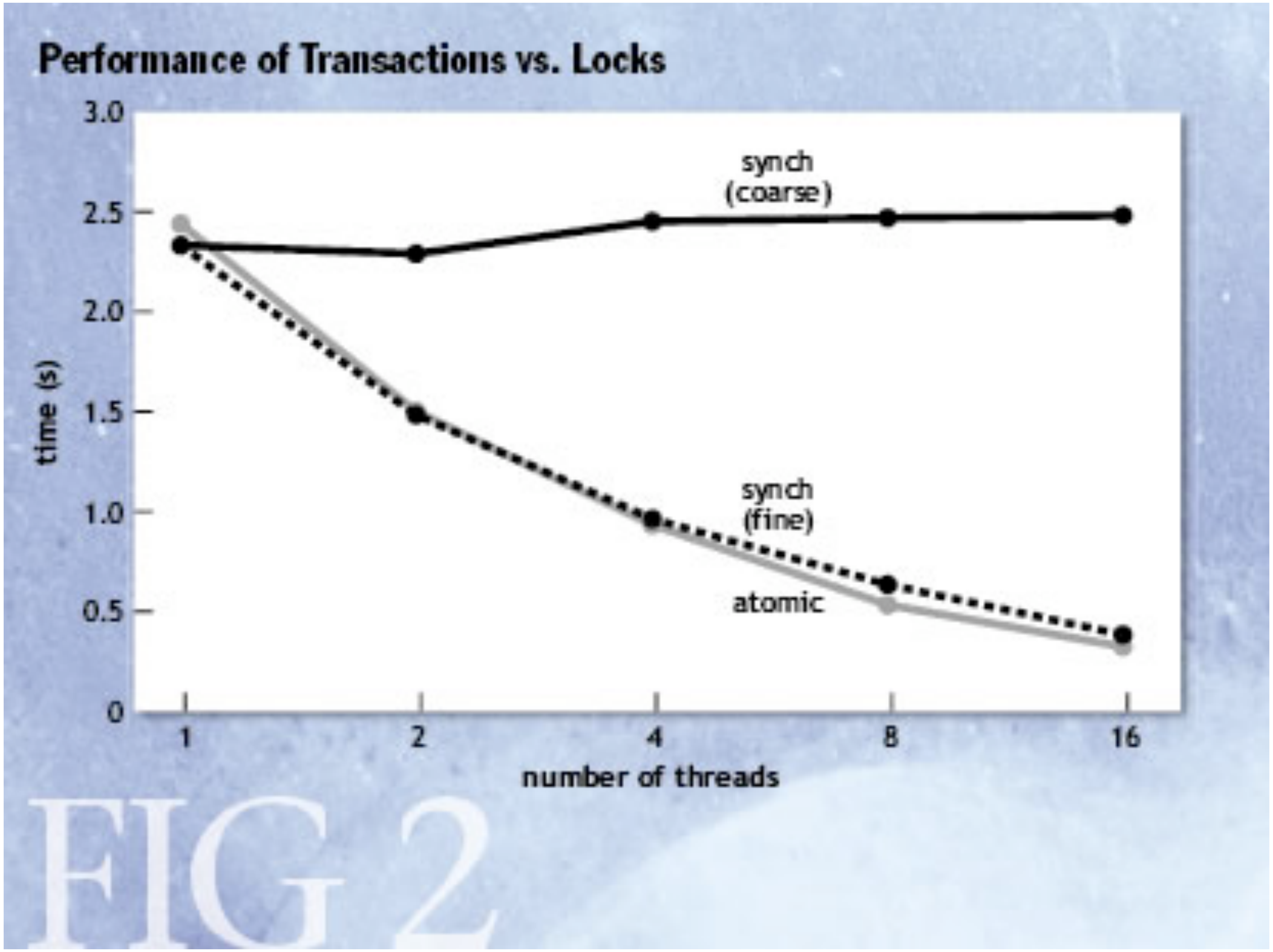
- Future for many computer architectures
- Eliminates mandatory locking
  - (hopefully) reduces the complexity of coding
- Retries blocks of code if a conflict occurs
  - Detects cache invalidation by another thread
  - Backs up execution to a checkpoint
  - Retry or use alternate means
- Basis for possible compiler generated “speculative parallelization”
  - Even if a proof can not be generated, the code may be parallelized

Transaction Logic

```
public static <T> T doIt(Callable<T> xaction) {
  T result = null;
  while (!Thread.stop) {
    beginTransaction ();
    try {
      result = xaction . call ();
    } catch (AbortedException d) {
    } catch (Exception e) {
      throw new PanicException("Unhandled␣exception␣" + e);
    }
    if (commitTransaction()) {
      return result ;
    }
  }
}
```

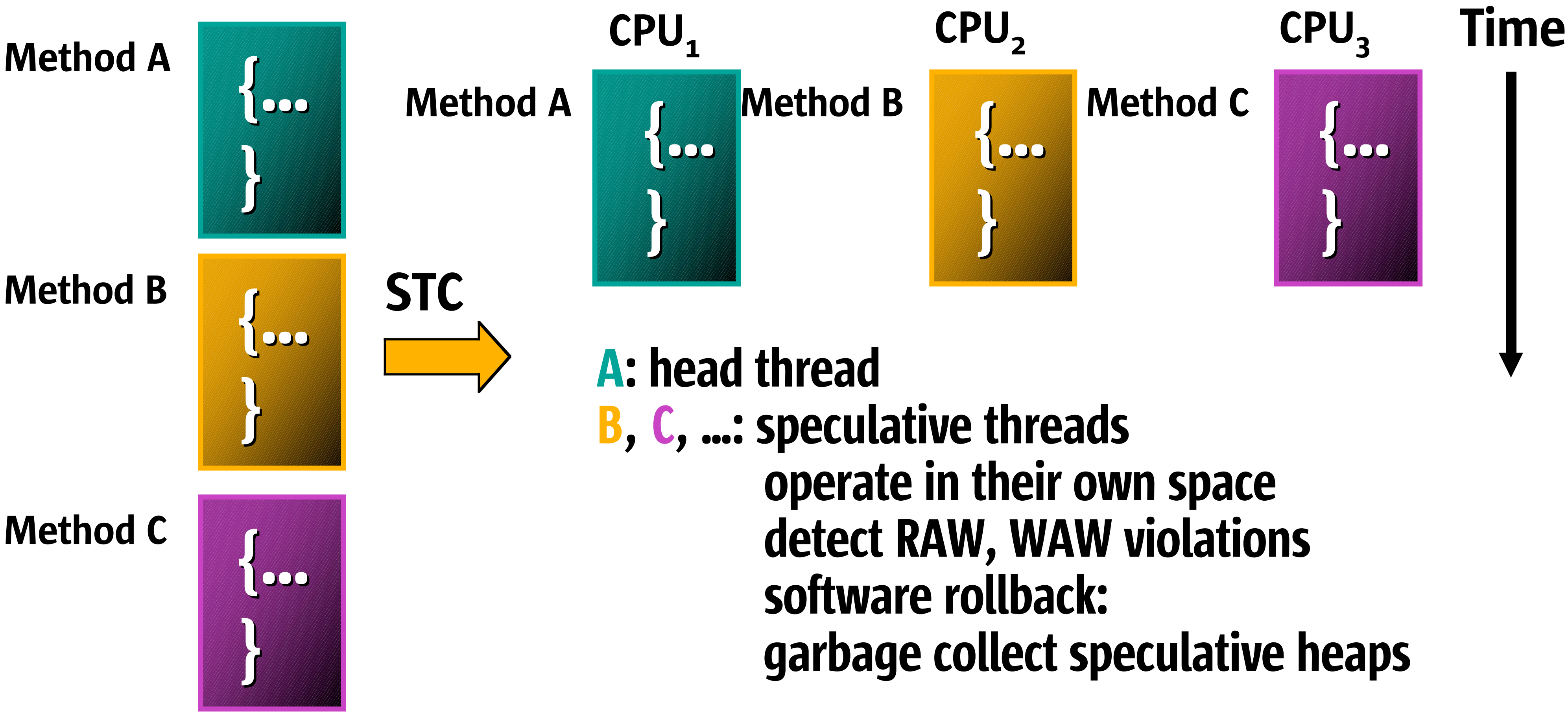
Figure 3. The transaction retry loop

Transactional memory gives high performance with less effort



- Unlocking Concurrency, From Computer Architecture, Vol. 4, No. 10 - December 2006 / January 2007 by Ali-Reza Adl-Tabatabai, Intel, Christos Kozyrakis, Stanford University, Bratin Saha, Intel

Space-Time Computing in MAJC



Joins: collapsing of dimensions

- heap merging
- new head thread

Conclusion

- Highly Threaded Commodity Processor
  - > Commercially available
- With a fully featured OS
  - > Solaris, OpenSolaris
- Support
  - > Free and pay
- Open Source
  - > Hardware and software
- Community
  - > Eager to help
  - > http://www.opensolaris.org

Free access to Niagara machines

- Proposals are being accepted for a University program to receive a four socket Victoria Falls Machine
  - Open Source project
  - Cryptanalysis
  - Highly threaded
  - Solaris or OpenSolaris
- Other proposals may get remote access to a Victoria Falls Machine
- Niagara-1 machine is available online (free) at
  - http://en.unix-center.net/
    - Located in China
    - Register for free account, validate email address
    - ssh t1000.unix-center.net

What will you open?

Thank You

James Hughes
james.hughes@sun.com