Sun’s E6000 Server Family

Erik Hagersten
Uppsala University
Sweden

What Approach to Shared Memory

(a) Shared Address Space
(b) Bus-based shared memory
(c) Dancehall
(d) Distributed memory

Looks like a NUMA but drives like a UMA

- Memory bandwidth scales with the processor count
- One “interconnect load” per (2xCPU + 2xMem)
- Optimize for the dancehall case (no memory shortcut)

SUN Enterprise Overview

- 16 slots with either CPUs or I/O
- Up to 30 UltraSPARC processors (peak 9 GFLOPs)
- Gigaplane™ bus has peak bw 2.67 GB/s; up to 30GB memory
- 16 bus slots, for processing or I/O boards

Enterprize Server E6000

An E6000 Proc Board

16 boards

288 signals = 256 data + ECC

80 signals = addr, uid, arb, ...

Snoop tags

Snoop tags

Data

Address Controller

Mem

$ S P

$ S P

$ S P

$ S P

$ S P

$ S P

$ S P

$ S P

$ S P

$ S P
An I/O Board

- 80 signals = addr, uid, arb, ...
- 288 signals = 256 data + ECC

Split-Transaction Bus

- Split bus transaction into request and response sub-transactions
- Separate arbitration for each phase
- Other transactions may intervene
  - Improves bandwidth dramatically
  - Response is matched to request
  - Buffering between bus and cache controllers

Gigaplane Bus Timing

- At most 16 electrical loads per signal
- 8 boards from each side (ex. 15 CPU + 1 I/O)
- 20.5 inches "centerplane"
- Well controlled impedance
- ~350-400 signals
- Runs at 90/100 MHz

Address Controller

Dual State Tags

Timing of a single read trans
Board 1 reading from mem 2

Foreign and own transactions queue in IQ
State Change on Address Packet

Protocol tuned for timing
SRAM lookup

Foreign and own transactions queue in IQ
State Change on Address Packet

Foreign and own transactions queue in IQ
State Change on Address Packet
A cascade of "write requests"

- Initially resides in CPU7's cache
- CPU1: RTW, ID=a
- CPU2: RTW, ID=b
- ...
- CPU5: RTW, ID=f

IQ1 = <mRTWid, fRTWid>
IQ2 = <mRTWid, fRTWid>
...
IQ5 = <mRTWid>
IQ7 = <fRTWid>

L2 cache = 8MB, snoop tags on-chip
CPU 1+GHz UltraSPARC III
Mem= 4+GB/CPU
FirePlane, 24 CPUs

ID = <CPU#, Uid>

Data Repeater

Scalable Shared-Memory Implementations
Erik Hagersten
Uppsala University
Sweden

Directory-based snooping: NUMA.
Per-cachline info in the home node

Three options
COMA cache-only (@SICS)
NUMA non-uniform
UMA uniform (a.k.a. SMP)
"Upgrade" in dir-based

Cache-to-cache in dir-based

Fully mapped directory

Reducing the Memory Overhead: SCI

Cache Invalidation Patterns

Overflow Schemes for Limited Pointers
cc-NUMA issues

- Memory placement is key!
- Gotta’ migrate data to where it’s being used
- Gotta’ have cache affinity
  - Long time between process switches in the OS
  - Reschedule processor on the CPU it ran last
- Origin 2000’s migration always turned off

Sun’s WildFire System

- Runs unmodified SMP apps in a more scalable way than E6000
- Minot modifications to E6000 snooping required
- CPUs generate local address OR global address
- Global address --> no replication (NUMA)
- Coherent Memory Replication (~Simple COMA@ SICS)
- Hardware support for detecting migration/replication pages
- Directory cache + address translation cache backed by memory
- Deterministic directory implementation (easy to verify)

WildFire:
One Solaris spanning four nodes

- A page may have space allocated in many nodes
- HW maintains memory coherence per cache line
- Replication under SW control --> simple HW (S-COMA)
- Adaptive replication algorithm in OS (R-NUMA)
- Coherent Memory Replication (CMR)
- Hierarchical affinity scheduler (HAS)
- Few large nodes --> simple interconnect and coherence protocol
A WildFire Node

- 16 slots with either CPUs, IO or...
- WildFire extension board
  - Up to 28 UltraSPARC processors
  - Gigaplane™ bus has peak bw 2.67 GB/s
  - Local access time of 330ns (lmbench)

Sun WildFire Interface Board

Sun WildFire Interface Board

WildFire as a vanilla "NUMA"

NUMA -- local memory access

NUMA -- remote memory access

SRAM overhead = 10/512 = 2% (lower bound 2/512 = 0.4%)
Global Cache Coherence Prot.

- Mod dir entry
- Reply(Data)
- Dir$
- Mtag
- Mem
- I/F
- Cache
- Proc

Access right changes

NUMA -- local memory access

- Interconnect
- Access right
- OK?
- NO!!
- Dir$
- Mtag
- Mem
- I/F
- Cache
- Proc

Gigaplane Bus Timing

- Arbitration
- Address
- State
- Tag
- Status
- Data

WildFire Bus Extensions

- Asserted by WildFire
- Resent by WildFire

Ignore transaction squashes an ongoing transaction => not put in IQ
WildFire eventually reissues the same transaction
RTSF -- a new transaction sends data to CPU and memory

WildFire Directory -- only 4 nodes!!

- k nodes (with one or more pros)
- With each cache-block in memory: k presence-bits, 1 dirty-bit
- With each cache-block in cache: 1 valid bit, and 1 dirty (owner) bit

ReadRequest from main memory by processor i:
- If dirty-bit OFF then { read from main memory; turn p[i] ON}
- if dirty-bit ON then { recall line from dirty proc (cache state to shared); update memory; turn dirty-bit OFF; turn p[i] ON; supply recalled data to i}

NUMA "detecting excess misses"

- I thought you had the data!!
- Dir$
- Mtag
- Mem
- I/F
- Cache
- Proc
- Interconnection Network

- Data
- Access right
- OK?
Deterministic Directory

- MOSI protocol, fully mapped directory (one bit/node)
- Directory blocking: one outstanding trans/cache line
- Directory blocks new requests until completion received
- The directory state and cache state always in agreement (except for silent replacement...)

![Diagram showing directory blocking and state agreement](image)

Directory blocking: one outstanding trans/cache line
- Directory blocks new requests until completion received
- The directory state and cache state always in agreement (except for silent replacement...)

![Diagram showing directory blocking and state agreement](image)

Advantages of Multiprocessor Nodes

Pros:
- amortization of fixed node costs over multiple processors
- can use commodity SMPs
- fewer nodes to keep track of in the directory
- much communication may stay within node (NUCA)
- can share “node caches” (WildFire: Coherent Memory Replication)

Cons:
- bandwidth shared among processors and interface
- bus may increases latency to local memory
- snoopy bus at remote node increases delays there too

![Diagram showing advantages of multiprocessor nodes](image)

Memory cost of replication

Example: Replicate 10% of data in all nodes
- 50 nodes, each with 2 CPUs
  => 490% overhead
- 4 nodes, each with 25 CPUs
  => 30% overhead

![Diagram showing memory cost of replication](image)

WildFire’s Technology Limits

- DIR $ reach >> sum(cache size)
- Slow interconnect
- Hard to make busses faster
- SRAM size = DRAMsize/256
- Snoop frequency

![Diagram showing WildFire’s technology limits](image)
Sun’s SunFire 15k/25k

Erik Hagersten
Uppsala University
Sweden

StarCat
Sun Fire 15k/25k
(used at Lab2)

StarCat, 72 CPUs

StarCat Coherence Mechanism

Active Backplane
18x18 addr X-bar
18x18 addr X-bar

Expander board
CPU board

Data Repeater
Addr Rep.

Glob-coh

MTAG

Check!

DATA+
MTAG+
ECC =576 bits

DATA+
MTAG+
ECC =576 bits

Remote request

Addr (snoop)
Allocate Dir$ entry only for write requests. Speculate on clean data on Dir$ miss.

WildCat coherence w/o CMR & w/ faster interconnect.