

Portland State University
ECE 588/688

SGI Origin & The Cosmic Cube

© Copyright by Alaa Alameldeen and Haitham Akkary 2009

The SGI Origin

- Cache coherent non-uniform memory access
- Up to 512 nodes
- Scalable Cray link network (hypercube)
- 1 or 2 R10000 MIPS processors per node
- Up to 4G bytes per node
- Node connects to a portion of the IO subsystem
- No snooping within node

Portland State University – ECE 588/688 – Fall 2009 2

Key Goals

- Scale to large number of processors (up to 1024)
- Provide higher performance per processor
- Maintain cache-coherent globally addressable memory model
 - ◆ For ease of programming
- System cost lower than a high performance SMP

Portland State University – ECE 588/688 – Fall 2009 3

Origin Architecture

- Distributed shared memory (DSM)
- Directory-based cache coherence
- Designed to minimize latency difference between local and remote memory
- Hardware and software provided to insure most memory references are local
- Origin block diagram: paper figure 1
- Cache coherence does not require in-order message delivery
- I/O subsystem is also distributed and globally addressable
- I/O can DMA to and from all memory in the system
- Cluster bus is multiplexed but is not a snoopy bus
 - ◆ Reduce local and remote memory latency
 - Fewer processors on the bus
 - Remote request does not need to wait for snoop response

Portland State University – ECE 588/688 – Fall 2009 4

Origin Architecture (Cont.)

- Non-snoopy node bus tradeoff
 - ◆ Disadvantage: remote bandwidth needs to match local bandwidth, unlike in SMP node systems
 - ◆ Advantage: easier migration path for existing SMP software
- Page migration and replication insures most references are local
 - ◆ Memory reference hardware counters
 - ◆ Copy engine to copy at near peak memory bandwidth
- Rich synchronization primitives
- Fetch and op primitives are not cached and performed at memory
 - ◆ Useful in highly contended locks
 - ◆ HUB implements 4-way full cross bar between processors, memory and I/O-network
 - ◆ RAS features
 - ECC in external cache and memory
 - Faulty packets automatic retries
 - Modular design provides highly available hardware

Portland State University – ECE 588/688 – Fall 2009 5

Network

- Six ported router chip
- Fat-hypercube (paper figures 3 and 4)
- Low latency wormhole routing
- Four virtual channels per physical channel
- Congestion control to allow messages to adaptively switch between two virtual channels
- Support for 256 levels of message priority
- Increased priority via packet aging
- Automatic packet retries
- Software programmable routing tables

Portland State University – ECE 588/688 – Fall 2009 6

Cache Coherence Protocol

- Similar to DASH protocol but with significant improvements
 - ◆ MESI protocol is fully supported
 - Single fetch from memory for read-modify-writes
 - Permits processor to replace E block in cache without informing directory
 - Requests from processors that had replaced E blocks can be immediately satisfied from memory
 - ◆ Support of upgrade requests from S to E without data transfer
- Sequences of coherence transactions (see paper)

Configuration and Performance

- CPU Configuration
 - ◆ MIPS R10000
 - ◆ 195 MHz
 - ◆ 4-way out-of-order
 - ◆ 4 M byte L2 cache
 - ◆ Bus connected to the HUB chip
- Latency variation (paper table 4)
- High memory bandwidth (paper figures 11 & 12)
- Synchronization (paper figure 13)

The Cosmic Cube

- 64 small computers (8086/8087 processor)
- Point-to-point communication network
- Binary 6-cube
- Hardware Simulation of future VLSI implementation with single-chip nodes
- Suggests scalability into 1000s of nodes
- A message passing machine
 - ◆ Compare to cc-NUMA: Pros and Cons

N-cube Architecture

- Also called hypercube (paper figure 1)
- Internode communication scales well to large number of nodes (paper figure 2)
- High aggregate bandwidth
- High bisection bandwidth

Process Programming

- Hardware structure of Cosmic Cube is difficult to target for programming
- Resident operating system a more flexible machine-independent environment for concurrent computations
- Process model of computation is quite similar to the hardware structure but is usefully abstracted from it
- Programmer formulates problems in terms of processes and virtual channels between them
- Each process has a unique global ID
- Messages have headers containing src and dest IDs and message info (e.g. type, length)
- A node can have one process or multiple processes

Programming Model

- Message passing: communication and synchronization through messages
 - ◆ Explicitly seen by the programmer
- Programming model is reflected in the hardware and operating system
- Operating system kernel in each node
 - ◆ schedule processes within node
 - ◆ Provides system calls for processes to send and receive
- Single-program, multiple-data (SPMD)
 - ◆ N-body problem (Paper figures 3 & 4)

Reading Assignment

- Erik Hagersten et al., "DDM - A Cache-Only Memory Architecture," IEEE Computer, 1992 (Review)
- Babak Falsafi and David Wood, "Reactive NUMA: A Design for Unifying S-COMA and CC-NUMA," ISCA 1997 (Skim)
- Homework 2 due Monday Oct 26
- Midterm: Monday Nov 2