CS/ECE 757: Advanced Computer Architecture II
(Parallel Computer Architecture)

Clusters & Scalable Systems

Copyright 2005 Mark D. Hill
University of Wisconsin-Madison

Slides are derived from work by
Sarita Adve (Illinois), Babak Falsafi (CMU),
Alvy Lebeck (Duke), Steve Reinhardt (Michigan),
J. P. Singh (Princeton), and Jim Smith (Wisconsin).
Thanks!

Outline

• A Model

• Clusters with Ethernet

• Clusters with Special Networks

• Network Interfaces on Memory Bus

• Cray T3E Case Study
A Model

- What's a node?
- What's the network?
- How interface (hardware & software)?

How Interface (Hardware)?

- NI-1 with Ethernet (Cluster)
- NI-1 with special network (IBM SP-2)
- NI-2 with special network (Cray T3E)
- NI-3 with special network (MIT J-Machine) (skip)
**Clusters**

- PC nodes connected by Ethernet
- Usually programmed with MPI (see next slide)
- MPI Library uses standard TCP/IP

**Advantages**
- Cheap computation cycles
- Impressive peak rate (e.g., with no communication)
- Use it if it works for you

**Disadvantages**
- Medium bandwidth
- Terrible latency
- Shared-memory not too practical

**Successful! (too successful?)**

**Not much (hardware) architecture to discuss**

---

**Implementing MPI**

- Asynchronous messaging often uses high-latency handshake

```
send X, dest
send request
Tag Check
If not, wait
Go back to computation
receive
ready
data transfer
```

---

(C) 2005 Mark D. Hill from Adve, Falsafi, Lebeck, Reinhardt, Singh & Smith
Clusters with Special Networks

- PC nodes connected by System Area Network attached to I/O card (e.g., Myrinet, Infiniband)
- Usually programmed with MPI
- MPI Library uses SAN software

- More closely coupled than Cluster with Ethernet
- But proprietary solutions add cost & schedule delay
- Broad success elusive, despite technical advantages

-- Example: IBM SP-2

- Made out of essentially complete RS6000 workstations
- Network interface integrated in I/O bus (bw limited by I/O bus)
Network Interface (NI) on Memory “Bus”

- Nodes are microprocessor with special chip set
- Programmed with MPI or shared memory
- Consider proprietary software (libraries & OS)

- More closely coupled than Cluster with SAN
- But proprietary solutions add cost & schedule delay
- Broad success elusive, despite technical advantages
- Issues with vicious cycle
  - Costs more due to propriety hardware & software
  - Few user to amortize cost across
  - Need government support? (like bombers)

Intel Paragon (No Shared Memory)

Sandia’s Intel Paragon XPS-based Supercomputer

2D grid network with processing node attached to every switch

Intel Paragon node

Memory bus (64-bit, 50 MHz)

DMA

8 bits, 175 MHz, bidirectional

4-way interleaved DRAM

Driver

880

L1 $ 

i860

L1 $ 

22
Thinking Machines CM-5 (Data Parallel)

- Input and output FIFO for each network
- Two data networks
- Save/restore network buffers on context switch

Cray T3E (Shared Memory w/o Coherence)

- Scale up to 1024 processors, 480MB/s links
- Memory controller generates comm. request for nonlocal references
- No hardware mechanism for coherence (SGI Origin etc. provide this)
**SGI Origin2000 (Shared Memory w/ Coherence)**

- Upto 512 nodes (1024 processors)
- Directory state in same or separate DRAMs, accessed in parallel

**Cray T3D (T3E Predecessor)**

- Up to 2,048 Alpha 21064s
  - no off-chip L2 to avoid inherent latency
- In addition to remote mem ops, includes:
  - prefetch buffer (hide remote latency)
  - DMA engine (requires OS trap)
  - synchronization operations (swap, fetch&inc, global AND/OR)
  - message queue (requires OS trap on receiver)
- Big problem: physical address space
  - 21064 supports only 32 bits
  - 2K-node machine limited to 2M per node
  - external “DTB annex” provides segment-like registers for extended addressing, but management is expensive & ugly
Cray T3E

- Similar to T3D, uses Alpha 21164 instead of 21064 (on-chip L2)
  - still has physical address space problems
- E-registers for remote communication and synchronization
  - 512 user, 128 system; 64 bits each
  - replace/unify DTB Annex, prefetch queue, block transfer engine, and remote load / store, message queue
  - Address specifies source or destination E-register and command
  - Data contains pointer to block of 4 E-reg and index for centrifuge
  - Centrifuge
    - supports data distributions used in data-parallel languages (HPF)
    - 4 E-reg for global memory operation: mask, base, two arguments
- Get & Put Operations
  - Atomic Memory operations
    - E-registers & centrifuge used
    - F&I, F&Add, Compare&Swap, Masked_Swap
  - Messaging
    - arbitrary number of queues (user or system)
    - 64-byte messages
    - create msg queue by storing message control word to memory location
  - Msg Send
    - construct data in aligned block of 8 E-reg
    - send like put, but dest must be message control word
    - processor is responsible for queue space (buffer management)
  - Barrier and Eureka synchronization

T3E (continued)

- Atomic Memory operations
  - E-registers & centrifuge used
    - F&I, F&Add, Compare&Swap, Masked_Swap
  - Messaging
    - arbitrary number of queues (user or system)
    - 64-byte messages
    - create msg queue by storing message control word to memory location
  - Msg Send
    - construct data in aligned block of 8 E-reg
    - send like put, but dest must be message control word
    - processor is responsible for queue space (buffer management)
  - Barrier and Eureka synchronization