

**UPC Ph.D. Course on  
Parallel Computer Architecture  
Scalable Multiprocessors (Chapter 7)**

**Copyright 2003 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),  
and J. P. Singh (Princeton). Thanks!

---

---

---

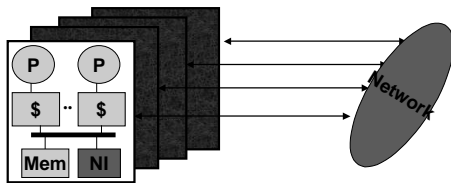
---

---

---

---

**Scalable Systems**



- **Must use non-bus network**
  - Issues with shipping bits across the network?
  - Issues with locating the network interface?
  - Issues with supporting a programming model?

(C) 2003 Mark D. Hill from Adve, Falsafi, Lebeck, Reinhardt, & Singh UPC Parallel Computer Architecture

2

---

---

---

---

---

---

---

**Outline**

- **Issues**
- **Case Studies**

(C) 2003 Mark D. Hill from Adve, Falsafi, Lebeck, Reinhardt, & Singh UPC Parallel Computer Architecture

3

---

---

---

---

---

---

---

### Network Transaction Primitive

The diagram illustrates a network transaction primitive. It shows a 'Source Node' on the left and a 'Destination Node' on the right, connected by a 'Communication Network'. The Source Node contains an 'output buffer' with an upward arrow. The Destination Node contains an 'input buffer' with a downward arrow. A 'serialized msg' is shown being sent from the Source Node's output buffer to the Destination Node's input buffer via the Communication Network. Ellipses (...) are shown between the two nodes, indicating a network path.

- One-way transfer of information from source to destination
- causes some action at the destination
  - process info and/or deposit in buffer
  - state change (e.g., set flag, interrupt program)
  - maybe initiate reply (separate network transaction)

(C) 2003 Mark D. Hill from Adve, Falsafi, Lebeck, Reinhardt, & Singh      UPC Parallel Computer Architecture      4

---

---

---

---

---

---

---

---

### Network Transaction Issues

- All architectures use point-to-point messaging
- How is this different from a bus?
- Protection
- Format
- Output buffering
- Input buffering
- Media arbitration & flow control
- Destination name & routing
- Action
- Completion detection
- Transaction ordering
- Deadlock avoidance
- Delivery guarantees

(C) 2003 Mark D. Hill from Adve, Falsafi, Lebeck, Reinhardt, & Singh      UPC Parallel Computer Architecture      5

---

---

---

---

---

---

---

---

### Node Communication Architecture

The diagram shows the internal architecture of a node. At the top is a 'Processor' (P) in an oval, connected to a 'Network Interface' (NI) in a rectangle via a 'Cache bus'. Below the Processor is a '\$' symbol in a rectangle, representing cache. This is connected to a horizontal 'Memory bus'. The Memory bus is connected to a 'Memory' block in the center and another 'NI' block on the right. On the left, the Memory bus is connected to an 'I/O bus', which is then connected to a third 'NI' block.

- Network Interface (NI) = Communication Assist (CA)
- I/O NI closer to "off-the-shelf"
- Processor NI from experimental machines
- Memory NI is compromise

(C) 2003 Mark D. Hill from Adve, Falsafi, Lebeck, Reinhardt, & Singh      UPC Parallel Computer Architecture      6

---

---

---

---

---

---

---

---

Outline

- Issues
- Case Studies

(C) 2003 Mark D. Hill from Adve, Falsafi, Lebeck, Reinhardt, & Singh
 

UPC Parallel Computer Architecture

7

---

---

---

---

---

---

---

---

Spectrum of Designs

Increasing HW Support, Specialization, Intrusiveness, Performance (??)

- None: Physical bit stream
  - physical DMA
- User Messaging
  - User-level port
  - User-level handler
- Remote virtual address
  - Processing, translation
  - Reflective memory (?)
- Global physical address
  - Proc + Memory controller
- Cache-to-cache (later)
  - Cache controller

nCUBE, iPSC, . . .  
  
 CM-5, \*T  
 J-Machine, Monsoon, . . .  
  
 Paragon, Meiko CS-2, Myrinet  
 Memory Channel, SHRIMP  
  
 RP3, BBN, T3D, T3E  
  
 Dash, KSR, Flash

(C) 2003 Mark D. Hill from Adve, Falsafi, Lebeck, Reinhardt, & Singh
 

UPC Parallel Computer Architecture

8

---

---

---

---

---

---

---

---

Network of Workstations

```

graph TD
    Processor -- interrupts --> CoreChipSet[Core Chip Set]
    CoreChipSet --- Cache
    CoreChipSet --- MainMemory[Main Memory]
    CoreChipSet --- IOBus[I/O Bus]
    IOBus --- DiskController[Disk Controller]
    IOBus --- GraphicsController[Graphics Controller]
    IOBus --- NetworkInterface[Network Interface]
    DiskController --- Disk1[Disk]
    DiskController --- Disk2[Disk]
    GraphicsController --- Graphics[Graphics]
    NetworkInterface --- Network[Network]
  
```

- Network interface on I/O bus
- Standards (e.g., PCI) => longer life, faster to market
- Slow (microseconds) to access network interface
- “System Area Network” (SAN): between LAN & MPP

(C) 2003 Mark D. Hill from Adve, Falsafi, Lebeck, Reinhardt, & Singh
 

UPC Parallel Computer Architecture

9

---

---

---

---

---

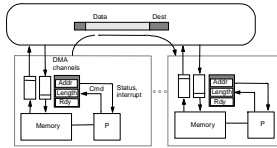
---

---

---

Page 3

## Net Transactions: Physical DMA

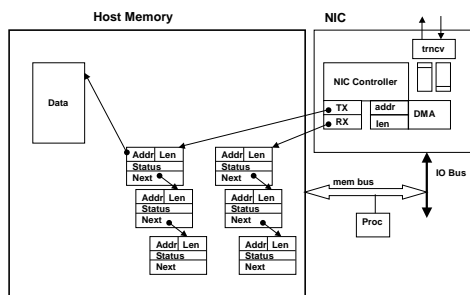


- **Physical addresses: OS must initiate transfers**
  - system call per message on both ends: ouch
- **Sending OS copies data to kernel buffer w/ header/trailer**
  - can avoid copy if interface does scatter/gather
- **Receiver copies packet into OS buffer, then interprets**
  - user message then copied (or mapped) into user space

(C) 2003 Mark D. Hill from Advv, Falsafi, Lebeck, Reinhardt, & Singh UPv Parallel Computer Architecture

10

## Conventional LAN Network Interface



(C) 2003 Mark D. Hill from Advv, Falsafi, Lebeck, Reinhardt, & Singh UPv Parallel Computer Architecture

11

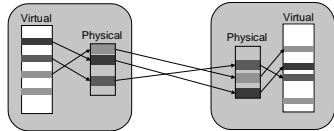
## Myricom Myrinet (Berkeley NOW)

- **Programmable network interface on I/O Bus (Sun SBUS or PCI)**
  - embedded custom CPU ("Lanai", ~40 MHz RISC CPU)
  - 256KB SRAM
  - 3 DMA engines: to network, from network, to/from host memory
- **Downloadable firmware executes in kernel mode**
  - includes source-based routing protocol
- **SRAM pages can be mapped into user space**
  - separate pages for separate processes
  - firmware can define status words, queues, etc.
    - » data for short messages or pointers for long ones
    - » firmware can do address translation too... w/OS help
  - poll to check for sends from user
- **Bottom line: I/O bus still bottleneck, CPU could be faster**

(C) 2003 Mark D. Hill from Advv, Falsafi, Lebeck, Reinhardt, & Singh UPv Parallel Computer Architecture

12

### DEC Memory Channel (Princeton SHRIMP)



- **Reflective Memory**
- **Writes on Sender appear in Receiver's memory**
  - send & receive regions
  - page control table
- **Receive region is pinned in memory**
- **Requires duplicate writes, really just message buffers**

(C) 2003 Mark D. Hill from Advv, Falsafi, Lebeck, Reinhardt, & Singh      UPC Parallel Computer Architecture      13

---

---

---

---

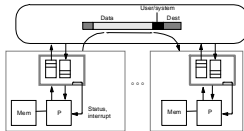
---

---

---

---

### User Level Ports



- **map network hardware into user's address space**
  - talk directly to network via loads & stores
- **user-to-user communication without OS intervention: low latency**
- **protection: user/user & user/system**
- **DMA hard... CPU involvement (copying) becomes bottleneck**

(C) 2003 Mark D. Hill from Advv, Falsafi, Lebeck, Reinhardt, & Singh      UPC Parallel Computer Architecture      14

---

---

---

---

---

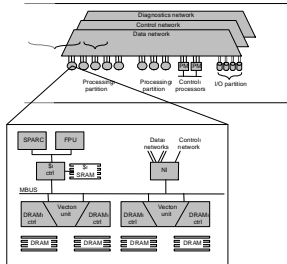
---

---

---

### Example: CM-5

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



(C) 2003 Mark D. Hill from Advv, Falsafi, Lebeck, Reinhardt, & Singh      UPC Parallel Computer Architecture      15

---

---

---

---

---

---

---

---

### User Level Handlers

- **Hardware support to vector to address specified in message**
  - message ports in registers
  - alternate register set for handler?
- **Examples: J-Machine, Monsoon, \*T (MIT), iWARP (CMU)**

(C) 2003 Mark D. Hill from Adve, Falsafi, Lebeck, Reinhardt, & Singh      UPC Parallel Computer Architecture      16

---

---

---

---

---

---

---

---

### Active Messages

- **User-level analog of network transaction**
  - invoke handler function at receiver to extract packet from network
  - grew out of attempts to do dataflow programming on msg-passing machines
  - handler may send reply, but no other messages
- **Event notification: interrupts, polling, events?**
- **May also perform memory-to-memory transfer**

(C) 2003 Mark D. Hill from Adve, Falsafi, Lebeck, Reinhardt, & Singh      UPC Parallel Computer Architecture      17

---

---

---

---

---

---

---

---

### J-Machine

- **Each node a small message-driven processor**
- **HW support to queue msgs and dispatch to msg handler task**

(C) 2003 Mark D. Hill from Adve, Falsafi, Lebeck, Reinhardt, & Singh      UPC Parallel Computer Architecture      18

---

---

---

---

---

---

---

---

### Dedicated Message Processing Without Specialized Hardware

- Microprocessor performs arbitrary output processing (at system level)
- Microprocessor interprets incoming network transactions (in system)
- User Processor ↔ Msg Processor share memory
- Msg Processor ↔ Msg Processor via system network transaction

(C) 2003 Mark D. Hill from Adve, Falsafi, Lebeck, Reinhardt, & Singh      UPC Parallel Computer Architecture      19

---

---

---

---

---

---

---

---

### Example: Cray T3D

- **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

(C) 2003 Mark D. Hill from Adve, Falsafi, Lebeck, Reinhardt, & Singh      UPC Parallel Computer Architecture      20

---

---

---

---

---

---

---

---

### 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-regs and index for centrifuge
- **Centrifuge**
  - supports data distributions used in data-parallel languages (HPF)
  - 4 E-regs for global memory operation: mask, base, two arguments
- **Get & Put Operations**

(C) 2003 Mark D. Hill from Adve, Falsafi, Lebeck, Reinhardt, & Singh      UPC Parallel Computer Architecture      21

---

---

---

---

---

---

---

---

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-regs
  - send like put, but dest must be message control word
  - processor is responsible for queue space (buffer management)
- Barrier and Eureka synchronization

(C) 2003 Mark D. Hill from Adve, Falsafi, Lebeck, Reinhardt, & SinghUPC Parallel Computer Architecture22

---

---

---

---

---

---

---

---

Outline

- Issues
- Case Studies

(C) 2003 Mark D. Hill from Adve, Falsafi, Lebeck, Reinhardt, & SinghUPC Parallel Computer Architecture23

---

---

---

---

---

---

---

---