Node-to-Network Interface in Scalable Multiprocessors

CS 258, Spring 99
David E. Culler
Computer Science Division
U.C. Berkeley

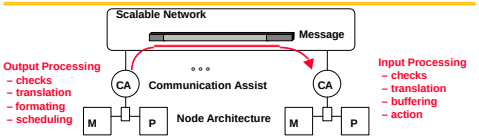
Racap: Common Challenges

- Input buffer overflow
  - N-1 queue over-commitment => must slow sources
  - reserve space per source (credit)
    - » when available for reuse?
      - Ack or Higher level
  - Refuse input when full
    - » backpressure in reliable network
    - » tree saturation
    - » deadlock free
    - » what happens to traffic not bound for congested dest?
  - Reserve ack back channel
  - drop packets
  - Utilize higher-level semantics of programming model

Racap: Challenges (cont)

- Fetch Deadlock
  - For network to remain deadlock free, nodes must continue accepting messages, even when cannot source msgs
  - what if incoming transaction is a request?
    - » Each may generate a response, which cannot be sent!
    - » What happens when internal buffering is full?
- logically independent request/reply networks
  - physical networks
  - virtual channels with separate input/output queues
- bound requests and reserve input buffer space
  - K(P-1) requests + K responses per node
  - service discipline to avoid fetch deadlock?
- NACK on input buffer full
  - NACK delivery?

Network Transaction Processing



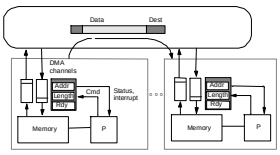
- Key Design Issue:
- How much interpretation of the message?
- How much dedicated processing in the Comm. Assist?

Spectrum of Designs

- None: Physical bit stream
  - blind, physical DMA — nCUBE, iPSC, . . .
- User/System
  - User-level port — CM-5, *T
  - User-level handler — J-Machine, Monsoon, . . .
- Remote virtual address
  - Processing, translation — Paragon, Meiko CS-2
- Global physical address
  - Proc + Memory controller — RP3, BBN, T3D
- Cache-to-cache
  - Cache controller — Dash, KSR, Flash

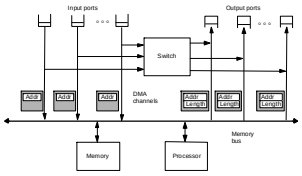
Increasing HW Support, Specialization, Intrusiveness, Performance (???)

Net Transactions: Physical DMA



- DMA controlled by regs, generates interrupts
- Physical => OS initiates transfers
- Send-side
  - construct system "envelope" around user data in kernel area
- Receive
  - must receive into system buffer, since no interpretation inCA

nCUBE Network Interface

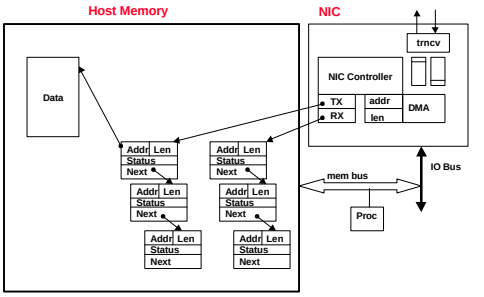


- **independent DMA channel per link direction**
  - **leave input buffers always open**
  - **segmented messages**
- **routing interprets envelope**
  - **dimension-order routing on hypercube**
  - **bit-serial with 36 bit cut-through**

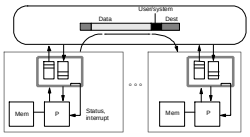
Os	16 ins	260 cy	13 us
Or	18	200 cy	15 us

- includes interrupt

Conventional LAN NI

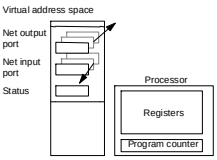


User Level Ports



- **initiate transaction at user level**
- **deliver to user without OS intervention**
- **network port in user space**
- **User/system flag in envelope**
  - **protection check, translation, routing, media access in src CA**
  - **user/sys check in dest CA, interrupt on system**

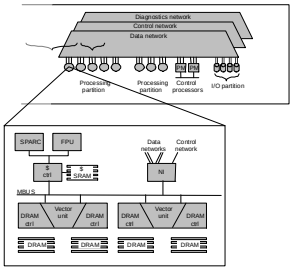
User Level Network ports



- **Appears to user as logical message queues plus status**
- **What happens if no user pop?**

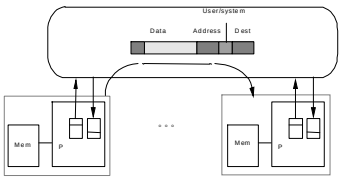
Example: CM-5

- **Input and output FIFO for each network**
- **2 data networks**
- **tag per message**
  - **index NI mapping table**
- **context switching?**
- ***T integrated NI on chip**
- **iWARP also**



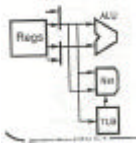
Os	50 cy	1.5 us	
Or	53 cy	1.6 us	
interrupt		10us	

User Level Handlers



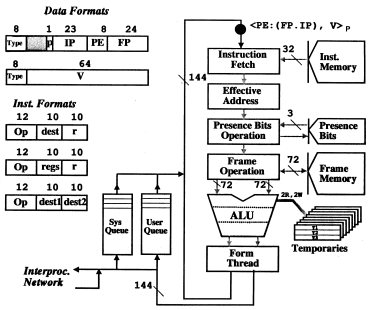
- **Hardware support to vector to address specified in message**
  - **message ports in registers**

J-Machine: Msg-Driven Processor

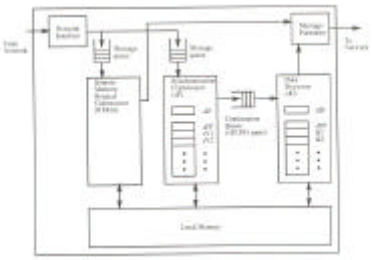


- Each node a small msg driven processor
- HW support to queue msgs and dispatch to msg handler task

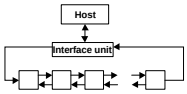
Monsoon Explicit Token-Store



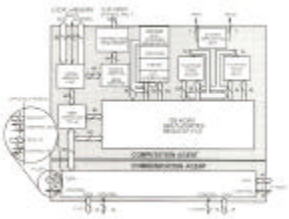
*T: Network Co-Processor



iWARP: Systolic Computation

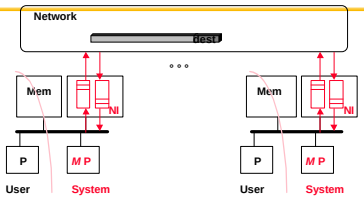


- Nodes integrate communication wi computation on systolic basis
- Msg data direct to register
- Stream into memo



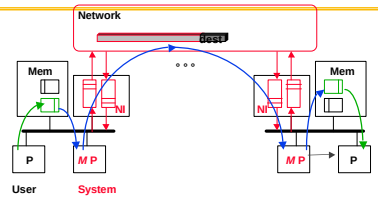
Dedicated processing without dedicated hardware design

Dedicated Message Processor



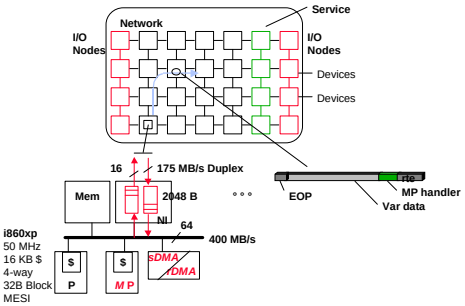
- General Purpose processor performs arbitrary output processing (at system level)
- General Purpose processor interprets incoming network transactions (at system level)
- User Processor <–> Msg Processor share memory
- Msg Processor <–> Msg Processor via system network transaction

Levels of Network Transaction

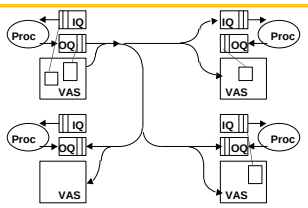


- User Processor stores cmd / msg / data into shared output queue
  - must still check for output queue full (or make elastic)
- Communication assists make transaction happen
  - checking, translation, scheduling, transport, interpretation
- Effect observed on destination address space and/or events
- Protocol divided between two layers

Example: Intel Paragon

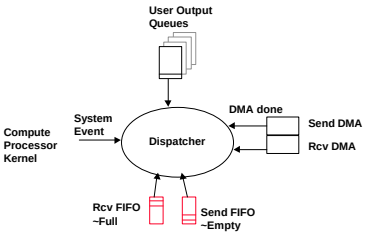


User Level Abstraction (Lok Liu)

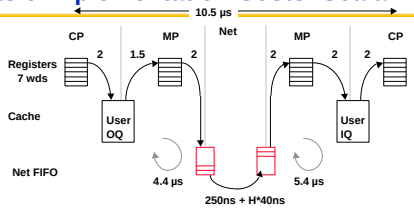


- Any user process can post a transaction for any other in protection domain
  - communication layer moves OQ_{src} –> IQ_{dest}
  - may involve indirection: VAS_{src} –> VAS_{dest}

Msg Processor Events

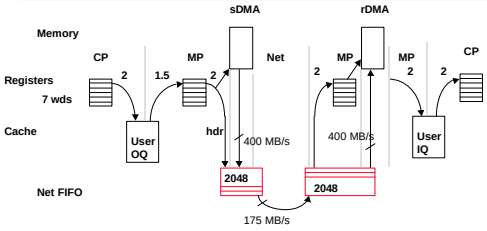


Basic Implementation Costs: Scalar



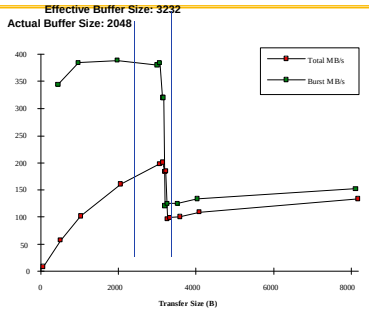
- Cache-to-cache transfer (two 32B lines, quad word ops)
  - **producer:** read(miss,S), chk, write(S,WT), write(I,WT),write(S,WT)
  - **consumer:** read(miss,S), chk, read(H), read(miss,S), read(H),write(S,WT)
- to NI FIFO: read status, chk, write, . . .
- from NI FIFO: read status, chk, dispatch, read, read, . . .

Virtual DMA -> Virtual DMA

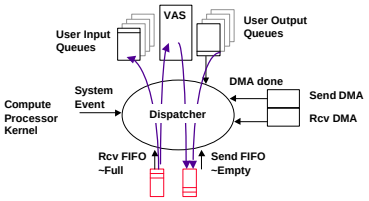


- Send MP segments into 8K pages and does VA –> PA
- Recv MP reassembles, does dispatch and VA –> PA per page

Single Page Transfer Rate

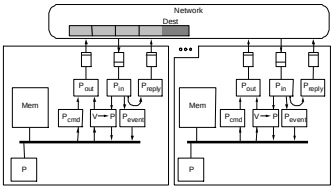


Msg Processor Assessment



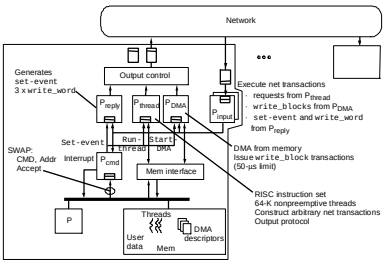
- Concurrency Intensive
  - Need to keep inbound flows moving while outbound flows stalled
  - Large transfers segmented
- Reduces overhead but adds latency

Case Study: Meiko CS2 Concept

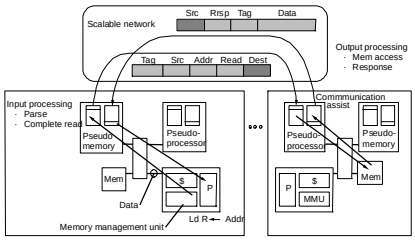


- Circuit-switched Network Transaction
  - source-dest circuit held open for request response
  - limited cmd set executed directly on NI
- Dedicated communication processor for each step in flow

Case Study: Meiko CS2 Organization

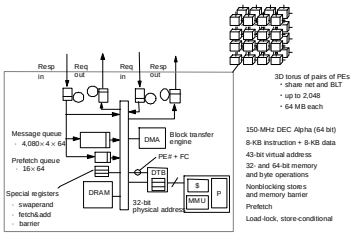


Shared Physical Address Space



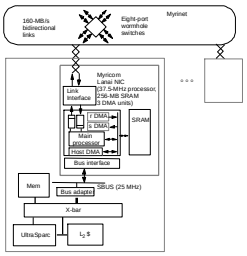
- NI emulates memory controller at source
- NI emulates processor at dest
  - must be deadlock free

Case Study: Cray T3D



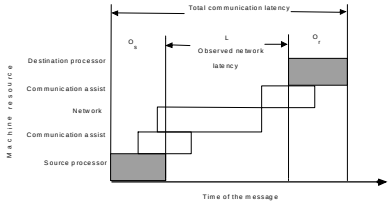
- Build up info in 'shell'
- Remote memory operations encoded in address

Case Study: NOW



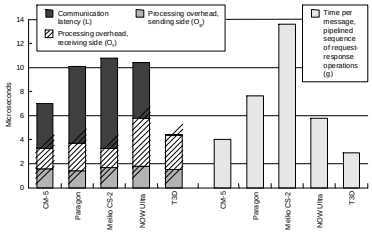
- **General purpose processor embedded in NIC**

Message Time Breakdown

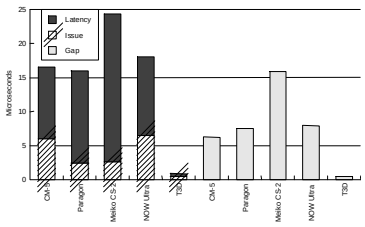


- **Communication pipeline**

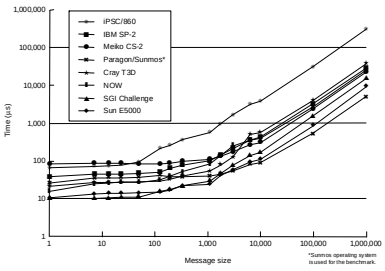
Message Time Comparison



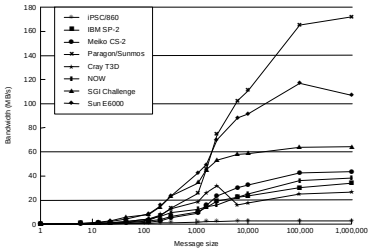
SAS Time Comparison



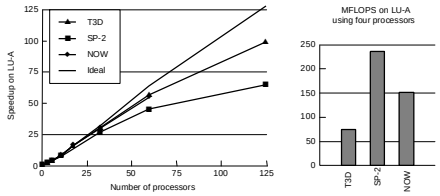
Message-Passing Time vs Size



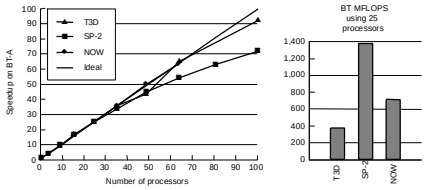
Message-Passing Bandwidth vs Size



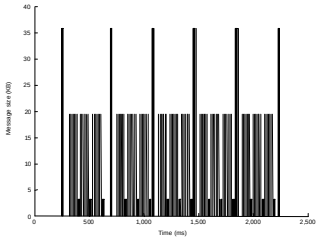
Application Performance on LU



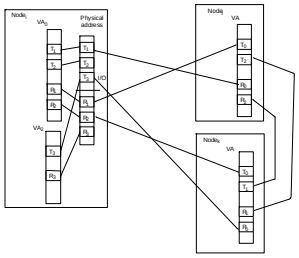
Application Performance on BT



Message Profile on BT

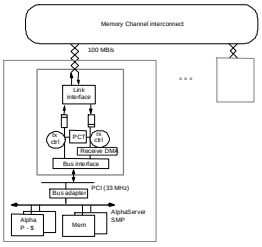


Reflective Memory



- **Writes to local region reflected to remote**

Case Study: DEC Memory Channel



- **See also Shrimp**