

General Purpose Node-to-Network Interface in Scalable Multiprocessors

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

iWARP: Systolic Computation

• Nodes integrate communication with computation on systolic basis
• Msg data direct to register
• Stream into memo

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

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} \rightarrow OQ_{dest}
  - may involve indirection: VAS_{src} \rightarrow VAS_{dest}

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 \rightarrow PA
- Recv MP reassembles, does dispatch and VA \rightarrow PA per page

Single Page Transfer Rate

- 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

Network

Shared Physical Address Space

- NI emulates memory controller at source
- NI emulates processor at dest

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
Application Performance on LU

Message-Passing Time vs Size

Message-Passing Bandwidth vs Size

Working Sets Change with P
Application Performance on BT

NAS Communication Scaling

NAS Communication Scaling: Volume

Communication Characteristics: BT

Beware Average BW analysis

Reflective Memory

- Writes to local region reflected to remote

NOW Handout Page 5
**Case Study: DEC Memory Channel**

- See also Shrimp

**Scalable Synchronization Operations**

- Messages: point-to-point synchronization
- Build all-to-all as trees
- Recall: sophisticated locks reduced contention by spinning on separate locations
  - caching brought them local
  - test&test&set, ticket-lock, array lock
  - O(p) space
- Problem: with array lock location determined by arrival order => not likely to be local
- Solution: queue-lock
  - build distributed linked-list, each spins on local node

**Queue Locks**

- Head holds lock; Each points to next waiter
- Shared pointer to tail
- Acquire
  - swap (fetch&store) tail with node address, chain in prev
- Release
  - signal next
  - compare&swap plus check to reset tail

**Parallel Prefix: Upward Sweep**

- generalization of barrier (reduce-broadcast)
- compute \( S_i = X_0 + X_1 + \ldots + X_{i-1} \) for \( i = 0, 1, \ldots \)
- combine children, store least significant

**Downward Sweep of parallel Prefix**

- Least branch send to most sig child
- when receive from above
  - send to least significant
  - combine with stored and send result to most sign