Chapter 5

Message Passing Architecture
5.1 Introduction To Message Passing

- A message passing architecture is used to communicate data among a set of processors without the need for a global memory.
- Each processor has its own local memory and communicates with other processors using messages.
5.1 Introduction To Message Passing

– Process Granularity: is the parameter describing the size of a process in a message passing system.
– Process granularity = computation time/communication time
– Types of granularity:
  • Coarse: each process holds a large number of sequential instructions and takes a substantial amount of time to execute.
  • Medium: a middle ground where communication overhead is reduced.
  • Fine: each process contains a few sequential instructions.
– Message passing multiprocessors use mostly medium or coarse granularity.
5.2 Routing in Message Passing Networks

- Routing for Broadcasting and Multicasting
  - 2 types of communication operations:
    - One-to-one (unicast): message is communicated to a single destination node.
    - Collective: a number of routing operations are under this category.
      - Broadcast: one-to-all
      - Multicast: one-to-many
5.2 Routing in Message Passing Networks

• Routing Potential Problems
  – Deadlock:
    • When 2 messages, each is holding the resources required by the other in order to move, both messages will be blocked (cyclic dependency for resources).
    • Straightforward solution (but inefficient) is rerouting.
    • Another solution is avoidance of occurrence of deadlock using a strict monotonic order of network resources.
    • Channel dependency graph (CDG) is a technique for developing a deadlock-free routing algorithm.
5.2 Routing in Message Passing Networks

A 4-node network and its CDGs

(a) A 4-node network

(b) Channel dependency graph (CDG)

(c) CDG for a deadlock-free version of the network
5.2 Routing in Message Passing Networks

• Routing Potential Problems
  – Livelock:
    • A message goes around the network and never reaches its destination.
    • It results from using adaptive routing algorithms with dynamic injection, where nodes inject their messages in the network at arbitrary times.
    • Policies to avoid livelock are based on assigning a priority to a message injected to the network:
      – Messages are routed according to their priorities
      – Once a message is injected, only a finite number of messages will be injected with higher or equal priority.
5.2 Routing in Message Passing Networks

• Routing Potential Problems
  – Starvation:
    • A node suffers from starvation if it has a message to inject into the network but is never allowed to do so.
    • The simplest policy to avoid starvation is to allow each node to have an injection queue that competes with the queues of the incoming links to the same node.
      – The main disadvantage is that a node with a high message injection rate can slow down all the other nodes in the network.
### 5.3 Switching Mechanisms in Message Passing

<table>
<thead>
<tr>
<th>Switching Mechanism</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| Circuit switching     | 1. Suitable for long messages  
2. Deadlock-free                                                              | Wasting of bandwidth                               |
| Store-and-forward     | 1. Simple  
2. Suitable for interactive traffic  
3. Bandwidth on demand                                                   | 1. Buffer for every packet  
2. Potential long latency  
3. Potential deadlock                                                  |
| Virtual cut-through   | 1. Good for long messages  
2. Possible deadlock avoidance  
3. Elimination of data-link protocol                                     | 1. Need for multiple message buffers  
2. Wasting of bandwidth  
3. Mainly used with profitable routing                                   |
| Wormhole              | 1. Good for long messages  
2. Reduced need for buffering  
3. Reduced effect of path length                                         | 1. Possibility for deadlock  
2. Inability to support backtracking                                     |
5.3 Switching Mechanisms in Message Passing

• Wormhole Routing in Mesh Networks
  – Dimension-ordered (X-Y) routing
    • Each packet is routed in one dimension at a time, arriving at the proper coordinate in each dimension before proceeding to the next dimension.
    • By enforcing a strict monotonic order of dimensions traversed, Deadlock-free routing is guaranteed.
5.3 Switching Mechanisms in Message Passing

• Wormhole Routing in Mesh Networks
  – Dimension-ordered (X-Y) routing
    • Let \((s_x, s_y)\) and \((d_x, d_y)\) be coordinates of a source and a destination, \((g_x, g_y) = (d_x-s_x, d_y-s_y)\).

    • X-Y routing can be implemented by placing \(g_x\) and \(g_y\) in the first 2 flits of the message.

    – When the first flit arrives at a node, it is decremented or incremented:
      » If the result is different than 0, the message is forwarded along the same direction in which it arrived.
      » If the result = 0 and the message arrived on the Y-dimension, then the message is delivered to the local node.
      » If the result = 0 and the message arrived on the X-dimension, the flit is discarded and the next flit is examined.
      » If the flit is 0, the packet is delivered to local node
      » Otherwise, the packet is forwarded in the Y-dimension.
5.3 Switching Mechanisms in Message Passing

- Wormhole Routing in Mesh Networks

Dimension ordered (X-Y) routing in an 8×8 mesh network

- Destination node
- Source node
5.3 Switching Mechanisms in Message Passing

- **Virtual Channels**
  - A principle introduced to allow the design of deadlock-free routing algorithms.
  - Inexpensive method to increase the number of logical channels without adding more wires.
  - A number of adaptive routing algorithms are based on the use of virtual channels.
  - Adding virtual channels to an interconnection network is analogous to adding lanes to a street network (blocked messages are allowed to pass).
  - Virtual channels provide an additional degree of freedom in allocating resources to messages in a network.
5.3 Switching Mechanisms in Message Passing

- Virtual Channels
  - The paths X-A-B-Z and Y-A-B-W share the common link AB.
  - Therefore AB is multiplexed between the 2 paths.
5.4 Message Passing Programming Models

• A message passing architecture uses a set of primitives allowing processes to communicate with each other.

• The \textit{send} primitive takes a memory buffer and sends it to a destination node.

• The \textit{receive} primitive accepts a message from a source node and stores it in a specified memory buffer.
5.4 Message Passing Programming Models

- Implementation of the send/receive among processes requires a 3-way protocol:
  - No buffering required at source or destination.
  - Blocking of both sender and receiver for at least a full round-trip time (network bandwidth not fully utilized).
5.5 Processor Support For Message Passing

• The following features are needed to support message passing:
  – Port (communication channel) where 2 operations can be performed: send and receive.
  – Messages are used as communication among objects (fixed header, variable size of message body).
  – A task can hold multiple access rights on ports. In port set, a task can have either all or none of the access rights to a group of ports.
5.6 Example Message Passing Architectures

- Caltech Hypercube
- Inmos Transputer Systems
- Meiko CS-2
- Cosmic Cube
- nCUBE/2
- iPSC/2
- iPSC/860
- CM-5
- IBM POWERparallel 3
5.6 Example Message Passing Architectures

- IBM Scalable POWERparallel 3
  - Is the most recent IBM supercomputer series.
  - Consists of 6000 processors nodes.
  - Each node has a private memory and copy of the AIX operating system.
  - POWER3 processor is an eight-stage pipeline processor.
5.6 Example Message Passing Architectures

- IBM Scalable POWERparallel 3
5.6 Example Message Passing Architectures

- **IBM Scalable POWERparallel 3**
  - Nodes are connected by a high-performance scalable packet-switched network in a distributed memory and message passing.
  - The network’s building block is a 2-stage 16x16 switch board made up of 4x4 bidirectional crossbar switching elements.
  - Each link is bidirectional and has a 40 MB/s bandwidth in each direction.
  - The switch uses buffered cut-through wormhole routing allowing all processors to send messages simultaneously.
5.6 Example Message Passing Architectures

• IBM Scalable POWERparallel 3
  – Message passing programming style is the preferred style for performance on SP 3.
  – The main message passing operations are the send and receive.
  – Send can be either synchronous or asynchronous, blocking or nonblocking.
  – SP 3 programming environment offers 3 message passing libraries:
    • PVM
    • MPL
    • MPI
5.7 Message Passing vs. Shared Memory Architectures

• Shared Memory
  – Communications using implicit loads and stores to a global address space.
  – Communication and synchronization are distinct.
  – The programmer isn’t concerned with the details of the interprocessor communication.
  – This model is a polling interface (drawback as far as synchronization is concerned).
  – One-way communication of data isn’t possible.
5.7 Message Passing vs. Shared Memory Architectures

- **Message Passing**
  - Explicit communication model.
  - Messages include both data and synchronization in a single unit.
  - This model lends itself to applications having large synchronization components.
  - This model suffers from the need for marshaling cost.
5.8 Summary

- Shared memory systems may be easier to program, but are difficult to scale up to a large number of processors.
- If scalability to larger systems was to continue, systems had to use message passing techniques.
- It is apparent that message passing systems are the only way to efficiently increase the number of processors managed by a multiprocessor system.
- We discussed the architecture and network models of message passing systems.
- We shed some light on routing and network switching techniques.