Content

- Message passing cost
  - Principle parameters

- Routing Techniques
  - Store-and-forward routing
    - Packet Routing
  - Cut-through routing

- Mapping Techniques
Communication Cost

• Communication cost depends on
  – Network topology
  – Data handling and routing
  – Programming model semantics
  – Associated software protocols

Message Passing Cost

• The time taken to communicate a message between two nodes
  – Time to prepare a massage for transmission + time taken by the message to traverse the network to its destination

• Principle parameters
  – Startup time ($t_s$): time required to handle a message at the sending and receiving nodes. It includes
    – Prepare the message
    – Execute the routing algorithm
    – Establish an interface between the local node and the router
    • This delay is incurred only once
  – Per-hop time ($t_h$): time taken by the header of a message to travel between two directly-connected nodes in the network - also called node latency
  – Per-word transfer time ($t_w$):
    • If the channel bandwidth is $r$ words per second, then $t_w = 1/r$
Routing Techniques

- Routing techniques:
  - Store-and-forward routing
  - Packet Routing
    - Suited to networks with highly dynamic states and higher error rates such as local and wide area networks (LAN and WAN)
  - Cut-through routing

Store-And-Forward Routing

- Store-and-forward routing
  - When a message is traversing a path with multiple links, each intermediate node on the path forward the message to the next node after it has received and stored the entire message

- Example
  - A message with size $m$. It traverses $l$ links
  - At each link it incurs a cost $t_h$ for the header.
    - $t_h$ is quite small in current parallel computers and can be ignored
  - A cost $t_w m$ for the rest of the message to traverse the link
  - Total communication cost is

\[
t_{\text{comm}} = t_s + (mt_w + t_h)l
\]
Cost of Packet Routing

- Consider m word messages
  - Assumption: all packets traverse the same path
  - Size of the packet: r + s,
    - where r is the original message and s is the additional information carried in the packet
  - The time for packetizing the message is proportional to its length: mt_w1
  - Communicating one word every t_w2 seconds
  - Delay of t_h on each hop. If the first packet traverses l hops, then it takes time t_h + t_w2(r+s) to reach the destination. After this time, the destination node receives an additional packet every t_w2(r+s) seconds.
  - There are (m/r-1) additional packets, the total communication time

\[
t_{\text{comm}} = t_s + t_{w1}m + t_hl + t_{w2}(r+s) + (m/r-1)t_{w2}(r+s)
\]

\[
= t_s + t_hl + t_wm
\]

\[
where \ t_w = t_{w1} + t_{w2}(1 + \frac{s}{r}).
\]

Overhead?
Cut-Through Routing

• Cut-through routing
  – A message is broken into fixed size units called flow control digits (flits)
    • It can be smaller than a packet (four bits to 32 bytes)
    • determined by network parameters
  – A tracer is first sent from the source to the destination node to establish a connection
  – The flits are sent one after the other, following the same path
  – As soon as a flit is received at an intermediate node, the flit is passed on to the next node
    – No need to have a buffer at intermediate node to store the entire message
• It needs less memory and is faster
• Most current parallel computers and many LANs support cut-through routing

Communication cost example

– The message with \( m \) words traverses \( l \) links
– \( t_h \) is the per-hop time
– The header of the message takes \( l t_h \) to reach the destination
– Then the entire message arrives in time \( t_w m \) after the arrival of the header of the message
– The total communication time is

\[
t_{comm} = t_s + t_h l + mt_w
\]

– Message constants \( t_s, t_w \) and \( t_h \) are determined by hardware, software and messaging semantics
Cost Optimization

\[ t_{\text{comm}} = t_s + t_h l + m t_w \]

- How to optimize the cost of message transfer?
  - Communication in bulk
    - Aggregate small messages into a single large message
    - Need less total startup time
      - The value of \( t_s \) is much larger than those of \( t_w \) or \( t_h \) for clusters
  - Minimize the volume of data
  - Minimize the distance of data transfer
    - Smaller hops \( l \)

Routing Mechanism Classification

- A routing mechanism is critical to the performance
- Classify routing mechanism according to path
  - Minimal
    - Select one of the shortest path
  - Non-minimal
    - Routing the message along a longer path
      - To avoid network congestion
- Classify routing mechanism according to the state of the network
  - Deterministic routing
    - Determine a unique path for a message
    - May result in uneven use of communication resources in a network
  - Adaptive routing
    - Determine the path of the message using the information of network state
    - Detect congestion
**Dimension-Ordered Routing**

- **Dimension-ordered routing**
  - Assigns successive channels based on a numbering scheme determined by the dimension of the channel
  - XY-routing for a 2D mesh
    - A message is sent along the X dimension until it reaches the column of the destination node
    - Along the Y dimension until it reaches the destination
  - E-cube routing for a hypercube

**E-Cube Routing**

- Using bitwise exclusive-OR operation $P_i \oplus P_d$ to obtain 101
- Forward a message along the dimension corresponding to the least significant nonzero bit of 101 to node 011
- Repeat the above steps to reach the destination

---

**Figure 2.28** Routing a message from node $P_i$ (010) to node $P_d$ (111) in a three-dimensional hypercube using E-cube routing.
Mapping Techniques for Graph

- Given two graphs, $G(V, E)$ and $G'(V', E')$
  - $V$: tasks; $E$: task interaction
  - $V'$: Processors; $E'$: physical links
- Three parameters in mapping
  - Congestion
    - Maximum number of edges mapped onto any edge in $E'$
  - Dilation
    - Maximum number of links in $E'$ that any edge in $E$ is mapped onto
  - Expansion
    - The ratio of the number of nodes in the set $V'$ to that in set $V$
Embedding a Linear Array into a Hypercube

- A linear array (or a ring) composed of $2^d$ nodes can be embedded into a $d$-dimensional hypercube
  - Map node $i$ of the linear array onto node $G(i,d)$ of the hypercube
- Function $G(i,d)$ is called binary reflected gray code (RGC)
  - the $i$th entry in the sequence of Gray codes of $d$ bits

$$G(0,1) = 0$$
$$G(1,1) = 1$$
$$G(i, x + 1) = \begin{cases} 
G(i, x), & i < 2^x \\
2^x + G(2^x - 1 - i, x), & i \geq 2^x 
\end{cases}$$

Example

Figure 2.30: (a) A three-bit reflected Gray code; and (b) its embedding into a three-dimensional hypercube.
Embedding a Mesh into a Hypercube

• Embed a $2^r \times 2^s$ wraparound mesh into a $2^{r+s}$-node hypercube
  - Map node $(i, j)$ of the mesh onto node $G(i, r) || G(j, s)$ of the hypercube
  - $||$ denotes concatenation of the two Gray code

• Map a $2 \times 4$ mesh into an eight-node hypercube
  - The values of $r$ and $s$ are 1 and 2, respectively
  - For node $(0,0)$, $G(i,r) = G(0,1) = 0$, $G(0,s)=G(0,2) = 00$
  - $(0,0)$ is mapped to node 000 of the hypercube
  - $(0,1)$ is mapped to node $G(0,1)||G(1,2)$ which is node 001 of the hypercube

All nodes in the same row of the mesh are mapped to hypercube nodes whose labels have $r$ identical most significant bits
Embedding a Mesh into a Linear Array

![Image](a) Mapping a linear array into a 2D mesh (congestion 1).

![Image](b) Inverting the mapping – mapping a 2D mesh into a linear array (congestion 5).

Figure 2.32  (a) Embedding a 16 node linear array into a 2-D mesh; and (b) the inverse of the mapping. Solid lines correspond to links in the linear array and normal lines to links in the mesh.

Characteristics of Parallel Platforms and Paradigms

- **Mapping**
  - The programmer has little control on the mapping of processes onto physical processors
  - The mapping of processes to nodes might destroy the well-defined topology of tasks

- **Routing**
  - Many architectures rely on randomized routing
  - Randomized routing alleviate contention on network

- **Communication cost**
  - The per-hop time can be ignored
  - Dominated by startup latency for small message and by per-word component for large message

\[ t_{comm} = t_s + mt_w \]
Summary

- Communication cost depends on
  - Programming model semantics
  - Network topology
  - Data handling and routing
  - Associated software protocols
- Message passing cost
  - Principle parameters
- Routing Techniques
  - Store-and-forward routing
  - Packet Routing
  - Cut-through routing
- Mapping Techniques

Questions?