CS4379: Parallel and Concurrent Programming
CS5379: Parallel Processing

Lecture 19

Dr. Yong Chen
Associate Professor
Computer Science Department
Texas Tech University

Lecture Video

- Please view the lecture video either from Teams or from the below link:

- https://texastechuniversity.sharepoint.com/sites/CS4379-CS5379/Shared%20Documents/General/Lecture19.mp4

Course Info

- **Lecture Time**: TR, 12:30-1:50
- **Lecture Location**: ECE 217
- **Sessions**: CS4379-001, CS4379-002, CS5379-001, CS5379-D01
- **Instructor:** Yong Chen, Ph.D., Associate Professor
- **Email:** yong.chen@ttu.edu
- **Phone:** 806-834-0284
- **Office**: Engineering Center 315
- **Office Hours**: 2-4 p.m. on Wed., or by appointment
- **TA:** Mr. Ghazanfar Ali, Ghazanfar.Ali@ttu.edu
- **TA Office hours:** Tue. and Fri., 2-3 p.m., or by appointment
- **TA Office:** EC 201 A
- **More info:**
  - http://www.myweb.ttu.edu/yonchen
  - http://discl.cs.ttu.edu; http://cac.ttu.edu/; http://nsfcac.org

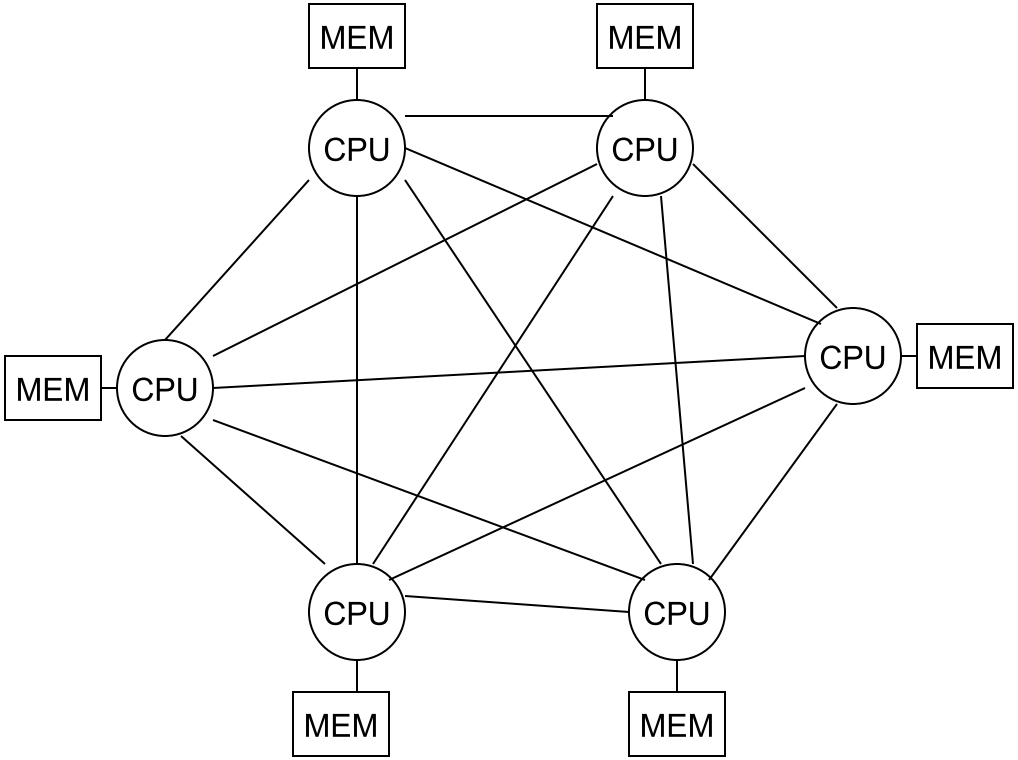
Outline

- Questions?

- Distributed-address-space architectures and message passing cost
- Basic communication operations
- One-to-all broadcast and All-to-One Reduction

Distributed-Address-Space Parallel Computers

- Each processor has its own local address space ("shared nothing", no global/shared address space)
- Easy to scale up, most large-scale parallel computers, clusters
- Processors share/exchange data via explicit message passing



Message Passing Costs

- Communication is a major overhead in programming distributed-address-space machines

- The cost of communication is dependent on a variety of features
  - Programming model semantics
  - Network topology (e.g. 2-D mesh/torus, 3-D mesh/torus, hypercube, etc.; sometimes customized for supercomputers)
  - Data handling and routing
  - Associated software protocols

Message Passing Costs

- The total time to transfer a message over a network comprises of the following:
  - *Startup time (t_s)*: Time spent at sending and receiving nodes (adding header, trailer, executing the routing algorithm, etc.).
  - *Per-hop time (t_h)*: This time is a function of number of hops and includes factors such as switch latencies, network delays, etc.
  - *Per-word transfer time (t_w)*: This time includes all overheads that are determined by the length (or size) of the message. This includes bandwidth of links, buffering overheads, etc.

Store-and-Forward Routing

- A message traversing multiple hops is completely received at an intermediate hop before being forwarded to the next hop.

- The total communication cost for a message of size *m* words to traverse *l* communication links is

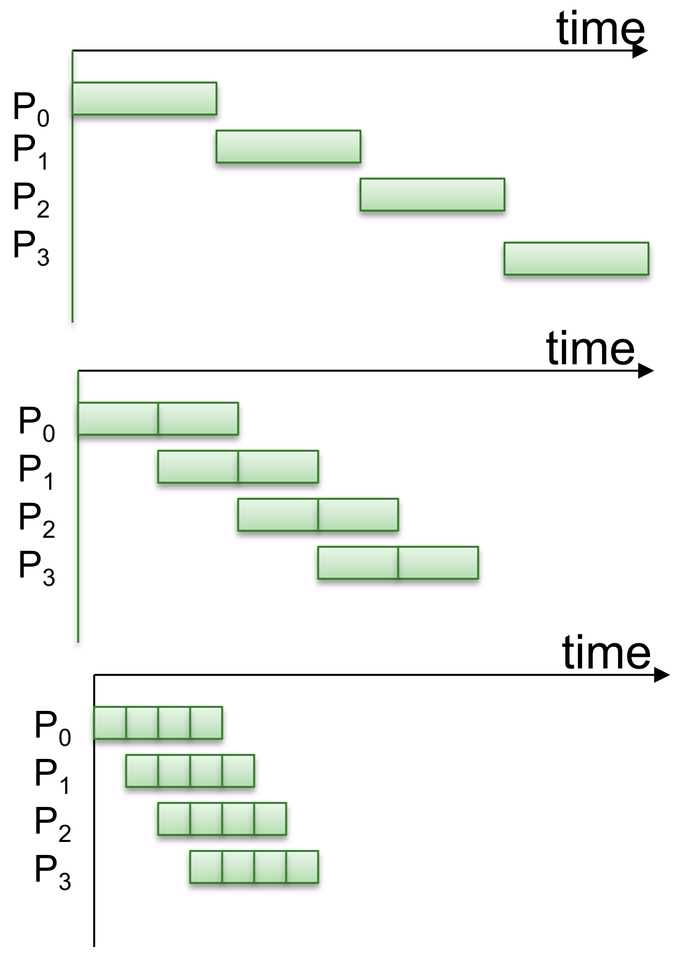
$$t_{comm} = t_s + (mt_w + t_h)l.$$

- In most platforms, t_h is small and the above expression can be approximated by

$$t_{comm} = t_s + mlt_w.$$

Packet Routing

- Store-and-forward makes poor use of communication resources.
- Packet routing breaks messages into packets and pipelines them through the network.



(1) A single message sent over a store-and-forward network

(2) The same message broken into two parts and send over the network

(3) The same message broken into four parts and sent over the network

Cut-Through Routing

- Takes the concept of packet routing to an extreme by further dividing messages into basic units called flits (flow control digits).

- The total communication time for cut-through routing is approximated by:

$$t_{comm} = t_s + t_h l + t_w m.$$

Implications of Message Passing Cost Model

- To optimize the cost of message transfers
- Communicate in bulk
  - Instead of sending small messages and paying a startup cost t_s for each, we want to aggregate small messages into a single large message and amortize the startup latency across a larger message
  - This is because on typical platforms such as clusters and message-passing machines, the value of t_s is much larger than those of t_h or t_w.
- Minimize the volume of data
  - To reduce the overhead paid in terms of per-word transfer time t_w, it is desirable to reduce the volume of data transferred as much as possible.
- Minimize the distance of data transfer
  - Minimize the number of hops l that a message must traverse

Outline

- Questions?

- Distributed address space architectures and message passing cost
- Basic communication operations
- One-to-all broadcast and All-to-One Reduction

Basic Communication Operations: Introduction

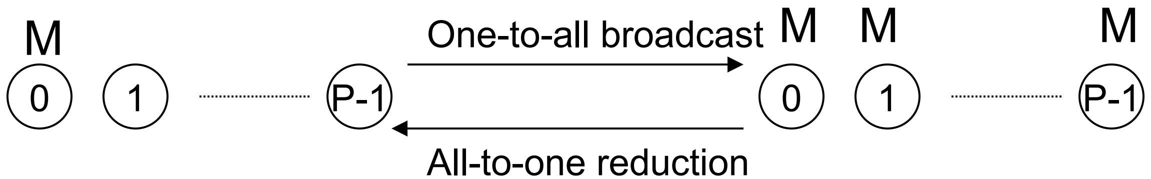
- Many communications in distributed-address-space machines occur in well-defined patterns involving a group of processes
- Efficient implementations of these operations can improve performance, reduce development effort and cost, and improve software quality
- Efficient implementations must leverage underlying architecture. For this reason, we refer to specific architectures here.
- We select a descriptive set of architectures to illustrate the process of algorithm design

Basic Communication Operations: Introduction

- Group communication operations can be built using point-to-point messaging primitives
- Will use message passing cost model to analyze the cost
  - Store-and-Forward Routing $t_{comm} = t_s + (mt_w + t_h)l.$
  - Cut-Through Routing $t_{comm} = t_s + t_h l + t_w m.$

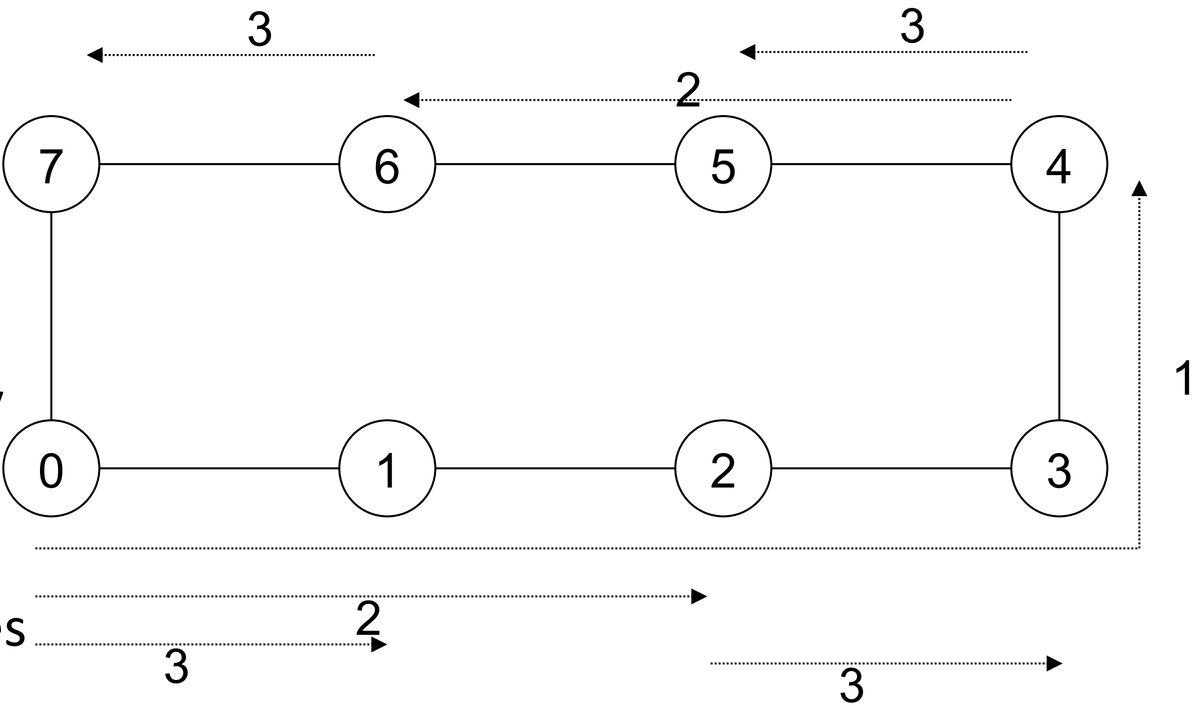
One-to-all Broadcast/All-to-one Reduction

- Algorithms often require a processor to send identical data to all other processors
- Called a one-to-all broadcast or singlenode broadcast
- At the start of a singlenode broadcast, one processor has m words of data that needs to be sent, at the end there are p copies of this data, one on each processor
- All-to-one reduction or singlenode reduction (dual): at the start of singlenode reduction each process has m words of data, the reduction combines all the data from processors using an associative operator to produce m words at the receiver
- Naïve singlenode broadcast or reduction using p-1 steps



One-to-all Broadcast: CT Routing on Ring

- Use recursive doubling: source sends a msg to a selected process, and both processes can continue simultaneously send msg
- Continue till all processes receive data



One-to-all Broadcast: CT Routing on Ring

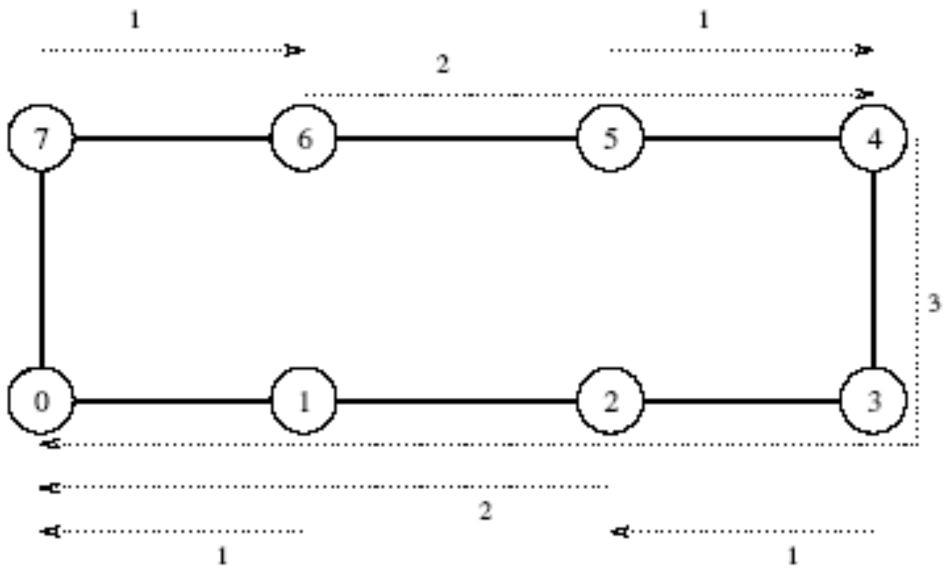
- Steps needed?
  - in *logp* steps

- In step i, message is sent to processor at a distance $\frac{p}{2^i}$
- All messages flow in the same direction
- Cost?

$$t_s \log(p) + t_w m \log(p) + t_h (p-1)$$

All-to-One Reduction: CT Routing on Ring

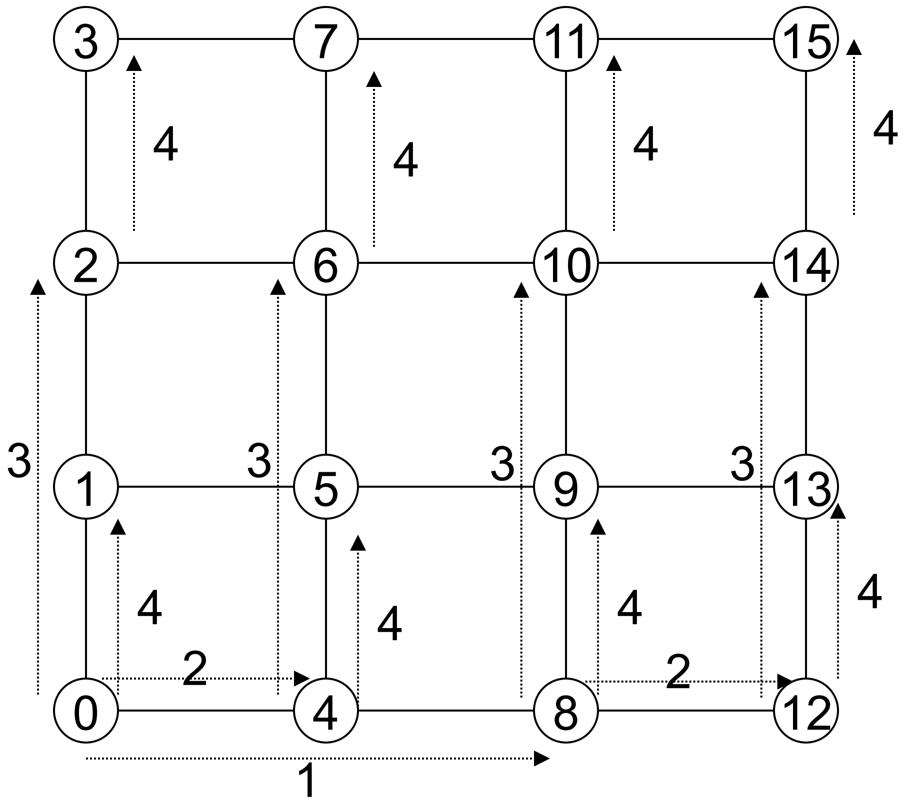
- Reduction can be performed in an identical fashion by inverting the process



Reduction on an eight-node ring with node 0 as the destination of the reduction.

One-to-all Broadcast: CT Routing on 2d Torus

- Apply ring algorithm for the processor row of sender
- Now use ring algorithm for all processor column



One-to-all Broadcast: CT Routing on 2d Torus

- Steps?

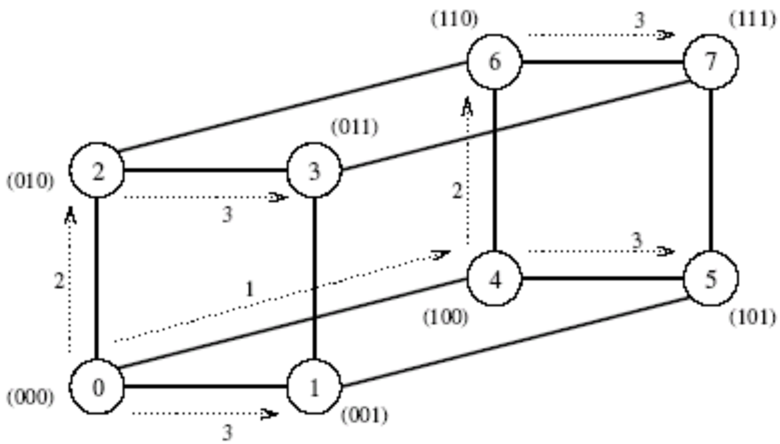
$$2\log(\sqrt{p}) = \log p$$

- Cost:?

$$(t_s + t_w m)\log(p) + 2t_h(\sqrt{p} - 1)$$

Broadcast and Reduction on a Hypercube

- A hypercube with 2^d nodes can be regarded as a d-dimensional mesh with two nodes in each dimension.
- The mesh algorithm can be generalized to a hypercube and the operation is carried out in d ($= \log p$) steps.



One-to-all broadcast on a three-dimensional hypercube. The binary representations of node labels are shown in parentheses.

Broadcast and Reduction on a Hypercube

- Steps?

$$\log p$$

- Cost:?

$$(t_s + t_w m + t_h)\log p$$

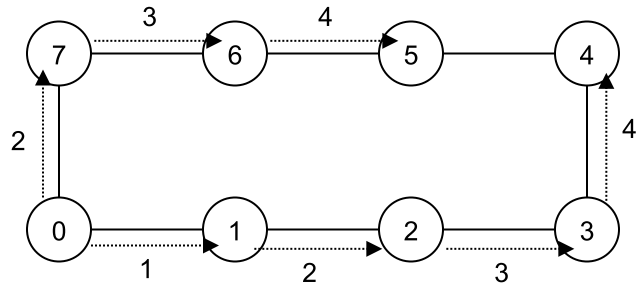
Broadcast and Reduction: SF Routing on Ring

- Steps?

$$\left\lceil \frac{p}{2} \right\rceil$$

- Cost?

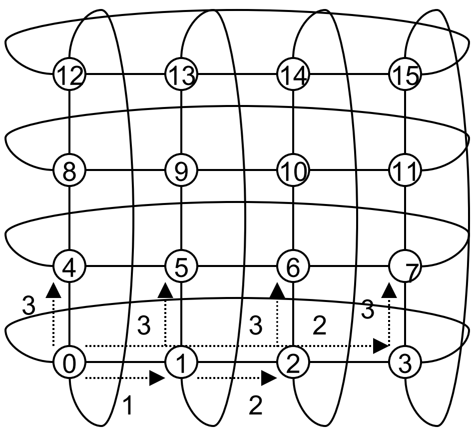
$$(t_s + mt_w + t_h)\left\lceil \frac{p}{2} \right\rceil \quad \text{Or} \quad (t_s + t_w m)\left\lceil \frac{p}{2} \right\rceil$$ if we ignore t_h



Broadcast and Reduction: SF Routing on 2d Torus

- Each row or column of the torus can be regarded as a ring
- Using ring method for the row to which the sending processor belongs; then use ring method for every column
- Steps? $2\left\lceil \frac{\sqrt{p}}{2} \right\rceil$
- Cost?

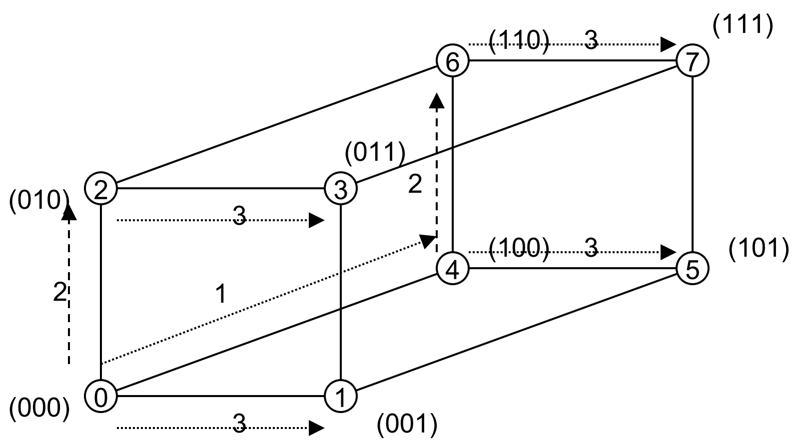
$$2(t_s + mt_w + t_h)\left\lceil \frac{\sqrt{p}}{2} \right\rceil$$



Broadcast and Reduction: SF Routing on Hypercube

- Takes log(p) steps for a p processor hypercube
- In the ith step, all processors that have the message transmit it to the neighboring processor that differs in the ith most significant bit
- Cost?

$$(t_s + mt_w + t_h)\log(p)$$



Reference book has pseudo-code/algorithm description

Readings

- Reference book ITPC – Chapter 2, 2.5.1; Chapter 4, 4.1
- Reference book has algorithm descriptions
  - One-to-all broadcast/All-to-one reduction on various architectures

Questions?

Questions/Suggestions/Comments are always welcome!

Write me: yong.chen@ttu.edu
Call me: 806-834-0284
See me: ENGCTR 315

If you write me an email for this class, please start the email subject with [CS4379] or [CS5379].