CS4379: Parallel and Concurrent Programming
CS5379: Parallel Processing

Lecture 20

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/Lecture20.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

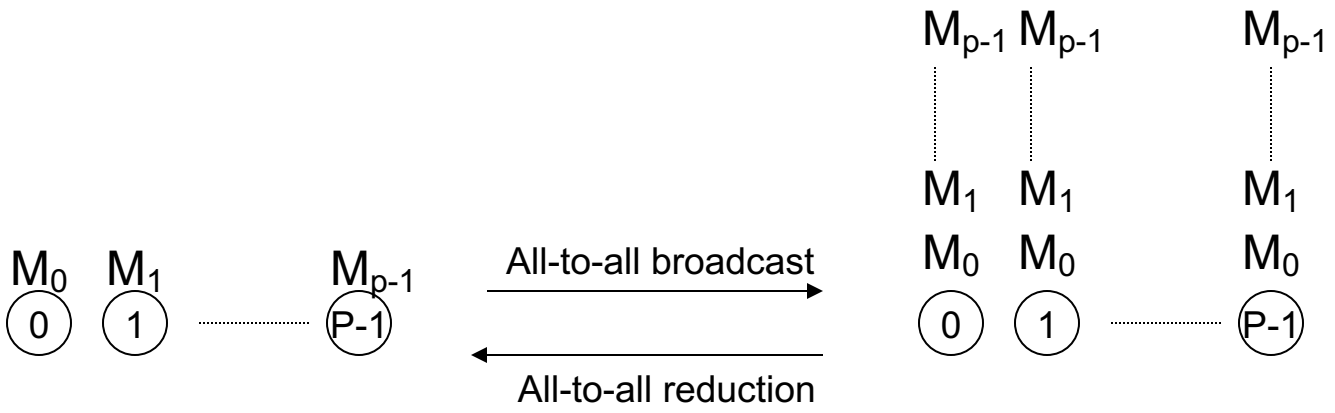
- Questions?

- All-to-all broadcast and All-to-all reduction
- All-reduce operation
- One-to-all scatter/All-to-one gather

All-to-all Broadcast/Reduction

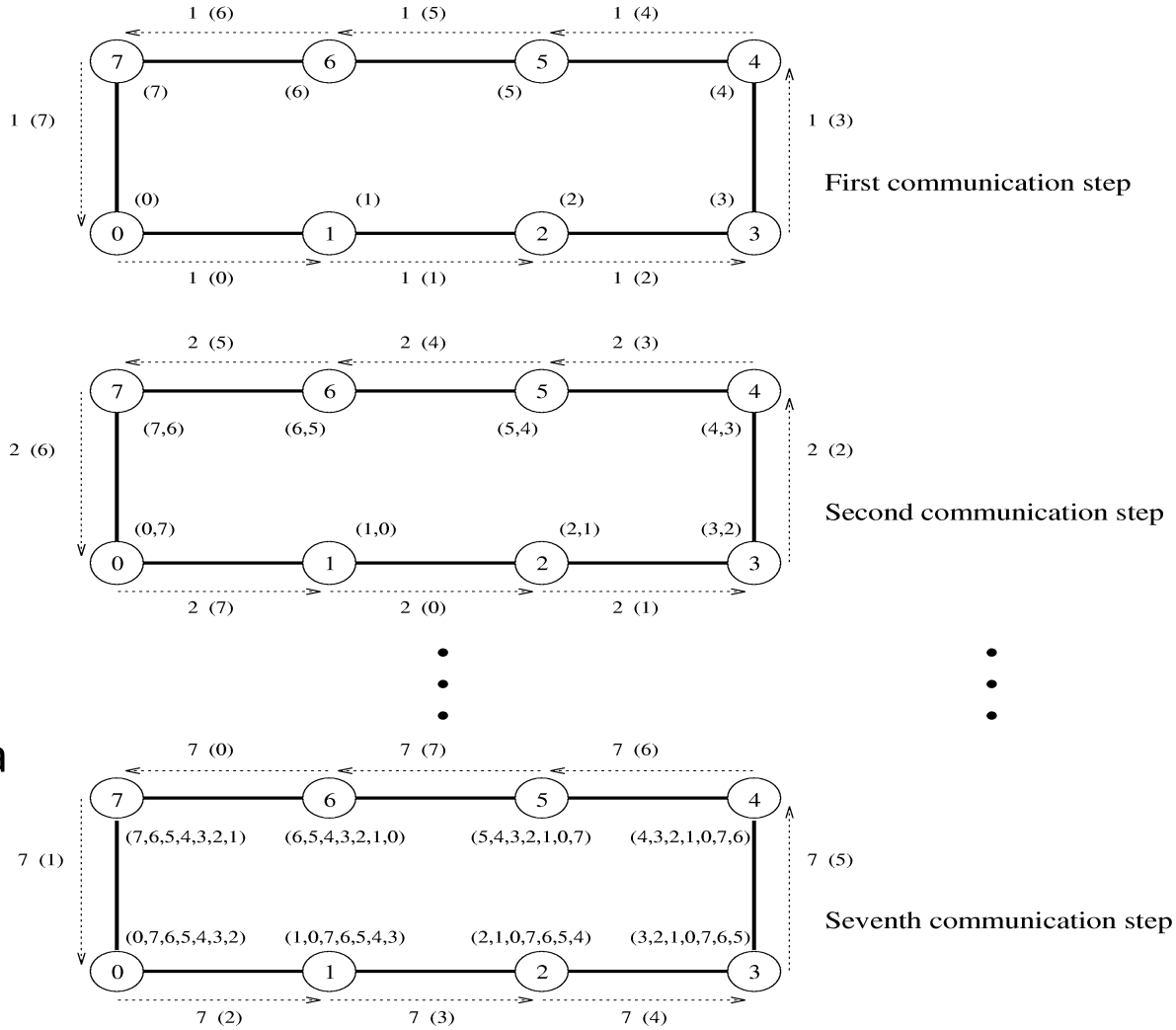
- Algorithms often require each processor to send different data to all other processors. This operation is called all-to-all broadcast or a multinode broadcast
- At the start of a multinode broadcast, each processor has m words of data; at the end each processor has a copy of the m words that originated at each of the other processors
- The dual of this operation is an all-to-all reduction or a multinode reduction
- Generalization of broadcast in which each processor is the source as well as destination
- Naïve multinode broadcast or reduction using *p* times of single-node broadcasts/reductions

All-to-all Broadcast/Reduction



All-to-all Broadcast/Reduction: CT Routing on Ring

- Every processor sends its message to the next processor on the ring in the first step
- In every subsequent step, all processors receive a message from the previous processor and send it to the next processor after retaining a copy for themselves



All-to-all Broadcast/Reduction: CT Routing on Ring

- Steps?

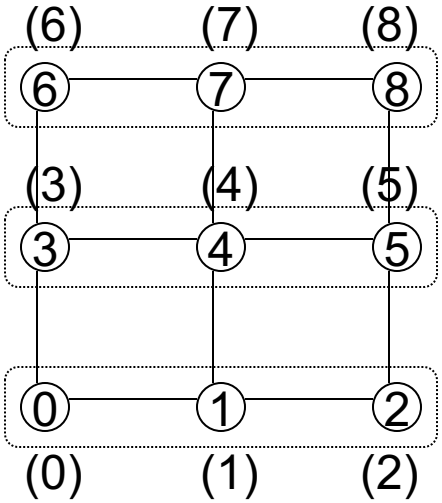
$$p-1$$

- Cost?

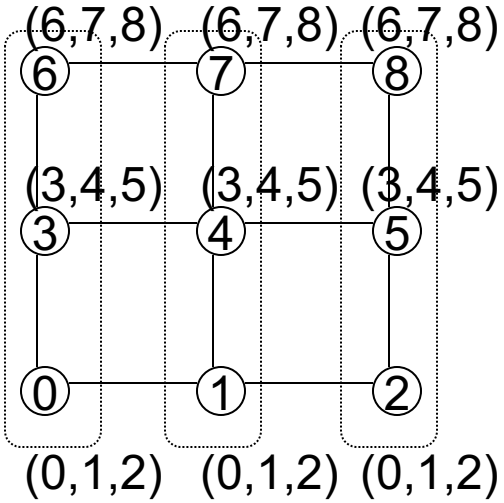
$$(t_s + t_w m + t_h)(p-1)$$

All-to-all Broadcast/Reduction: CT Routing on 2-D Torus

- Use ring method for each row of the torus
- Compose all $\sqrt{p}$ messages received into a single message and use the ring method for every column



(a) Initial data distribution



(b) Data distribution after rowwise broadcast

All-to-all Broadcast/Reduction: CT Routing on 2-D Torus

- Steps?

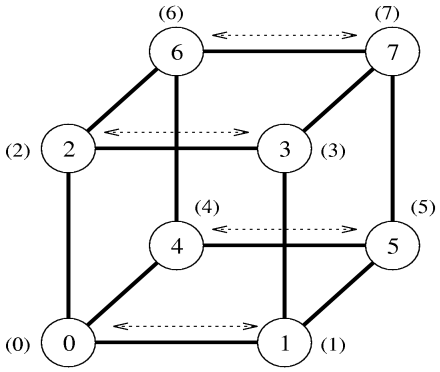
$$2(\sqrt{p}-1)$$

- Cost?

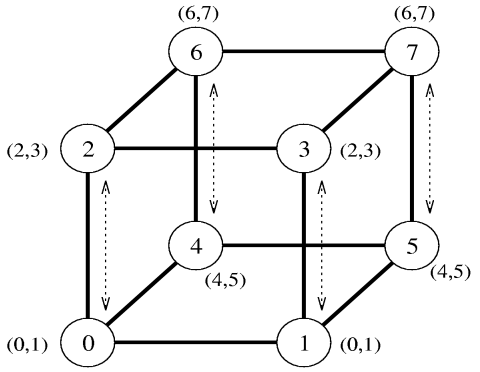
$$2t_s(\sqrt{p}-1)+t_w m(p-1)+2t_h(\sqrt{p}-1)$$

All-to-all Broadcast/Reduction: CT Routing on Hypercube

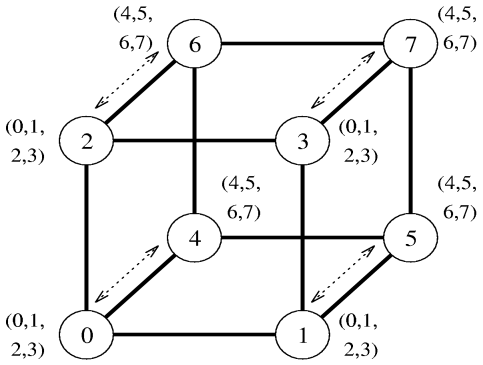
- Takes $\log(p)$ steps for a p-processor hypercube
- In the ith step, every processor exchanges messages with the neighboring processor that differs in the ith most significant bit
- Message size doubles at each of the *log p* steps



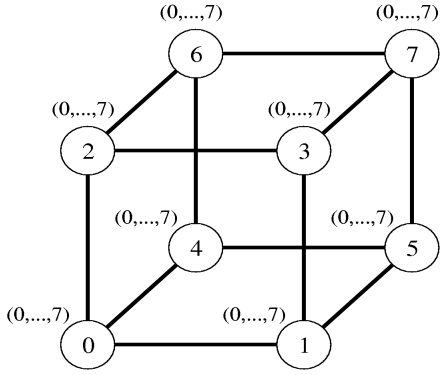
(a) Initial distribution of messages



(b) Distribution before the second step



(c) Distribution before the third step



(d) Final distribution of messages

All-to-all Broadcast/Reduction: CT Routing on Hypercube

- Cost?

$$\sum_{i=1}^{\log p} (t_s + 2^{i-1} t_w m + t_h)$$

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

All-to-all Broadcast/Reduction: SF Routing

- Store-and-forward routing performs similarly
- Cut-through routing does not provide benefits over store-and-forward for all-to-all broadcasts/reductions

Outline

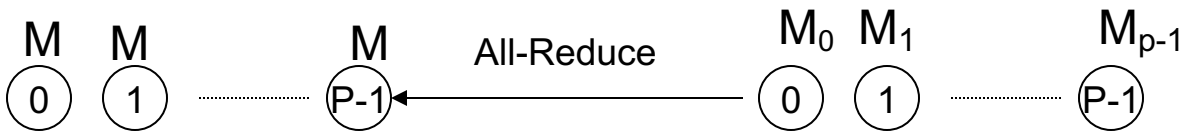
- Questions?

- All-to-all broadcast and All-to-all reduction
- All-reduce operation
- One-to-all scatter/All-to-one gather

All-Reduce Operations

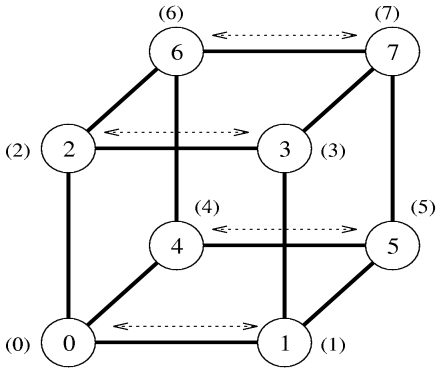
- In all-reduce, each node starts with a buffer of size *m* and the final results of the operation are identical buffers of size *m* on each node that are formed by combining the original *p* buffers using an associative operator

- Identical to all-to-one reduction followed by a one-to-all broadcast
  - Not the most efficient

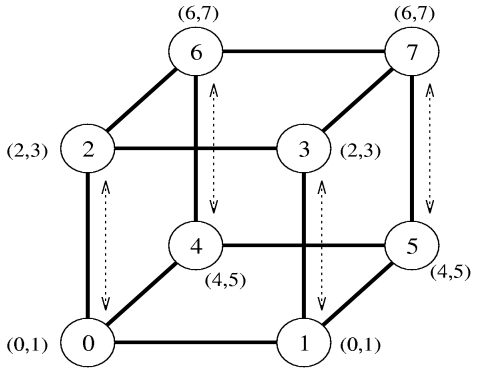


All-Reduce Operations

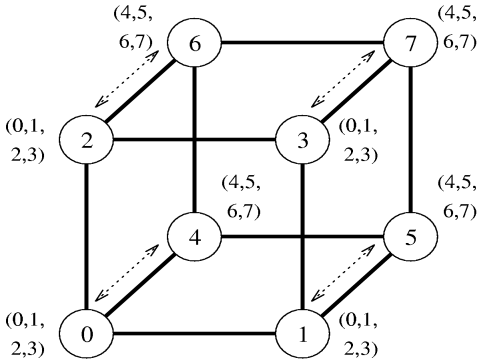
- Uses the pattern of all-to-all broadcast, instead
- The only difference is that message size does not increase here
  - Why?



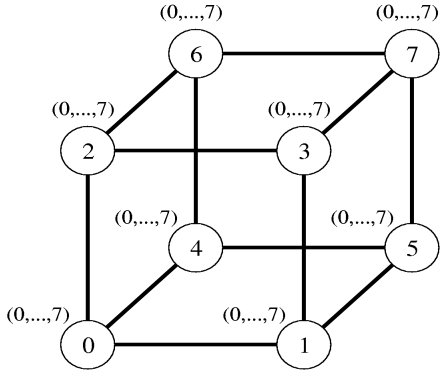
(a) Initial distribution of messages



(b) Distribution before the second step



(c) Distribution before the third step



(d) Final distribution of messages

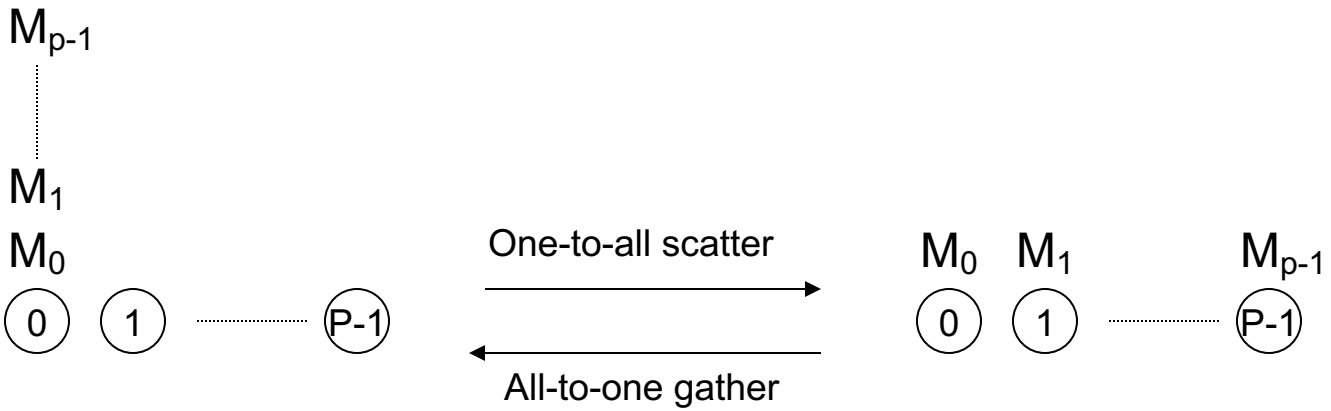
All-Reduce Operations

- Cost for CT/SF routing
  - $(t_s + t_w m + t_h) \log p$
- Message size remains the same

One-to-all Scatter

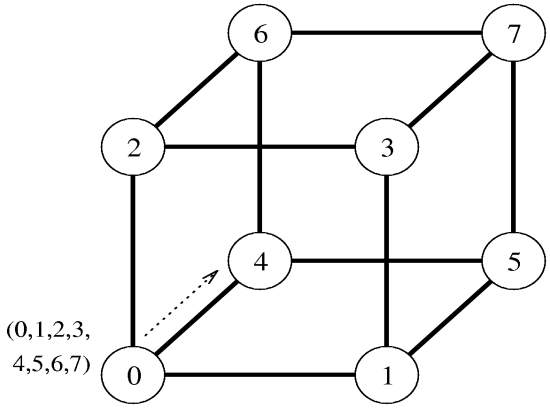
- Algorithms often require a processor to send different data to each of the other processors, this is called a one-to-all personalized communication or singlenode scatter (one-to-all scatter)
- At the start of a singlenode scatter, the source processor has *p* messages of *m* words, each of which needs to be sent to each of the other processors; at the end each of them has *m* words
- The dual of this operation is an all-to-one personalized communication or singlenode gather (all-to-one gather)
- At the start of a singlenode gather each processor has m bytes of data, the gather combines all the data from processors to produce mp words at the receiver
- Naïve singlenode scatter or gather using (*p*-1) steps

One-to-all Scatter/All-to-one Gather

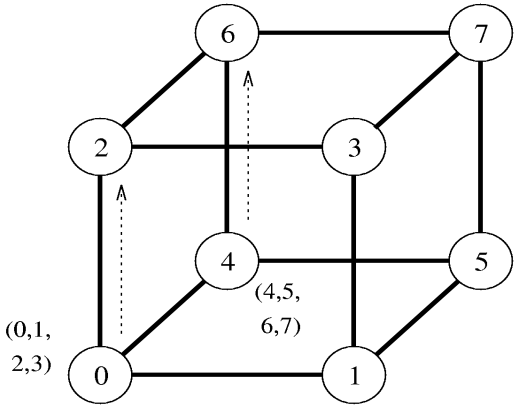


- Semantically different, but communication pattern similarly to one-to-all broadcast/all-to-one reduction
- Can follow broadcast/reduction for an efficient implementation

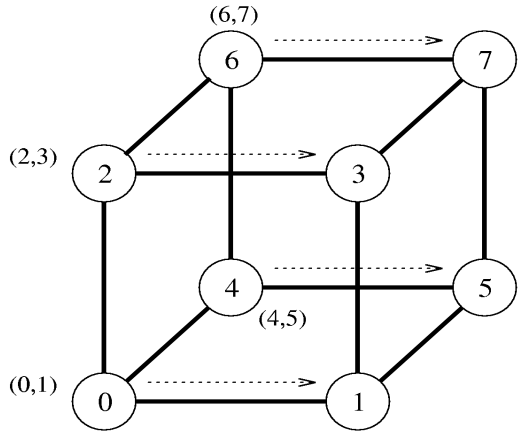
SF Routing on Hypercube



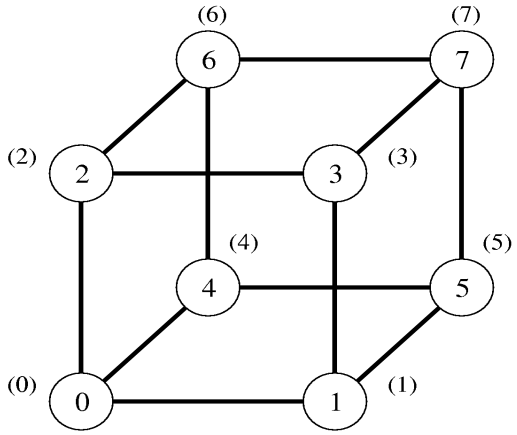
(a) Initial distribution of messages



(b) Distribution before the second step



(c) Distribution before the third step



(d) Final distribution of messages

SF Routing on Hypercube

- Takes $\log(p)$ steps for a *p*-processor hypercube
- In the *i*-th step, all processors that have messages transmit half of them to the neighboring processor that differs in the ith most significant bit
- Cost

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

CT Routing on Hypercube

- Cut-through routing performs similarly
- Does not provide benefits over store-and-forward for one-to-all scatter

Readings

- Reference book ITPC – Chapter 4, 4.2-4.4
- Reference book has algorithm descriptions too

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].