EE/CSCI 451: Parallel and Distributed Computation

Lecture #10

9/17/2020

Viktor Prasanna

prasanna@usc.edu

ceng.usc.edu/~prasanna

University of Southern California

Announcement

- Midterm 1 date: 9/25
  - 3:30 – 5:30 PM
- HW3 due 9/17 (Today)
- PHW3 due 9/24
- HW4 released (due 9/24)
- HW2 grades are out

HW2 Statistics	
Average	79.4
Median	96.0
Standard Deviation	35.5

Announcement: Midterm1 Logistics

- **Online Proctored** Exam:
  - Time: Week 6 discussion session – 2 hours: 3:30-5:30PM (Los Angeles time)
  - Format: Open-book, open-notes **[Attendence is required, no make-up given]**
  - Proctoring: 2 proctors watching different subgroups of students in separate Zoom meetings, links will be sent to students in advance
    - Require **camera-enabled** device
- Receiving and returning your exam:
  - Exam will be released on Piazza under resource page at/around 3:26 PM
  - You will submit the completed exam on Blackboard (a submission portal will be created in advance)
- Completing your exam:
  - Option 1 - Download the assignment pages (exam pages) as pdf files on to your tablet and annotate it with your answers. Only hand-annotated pdf files are acceptable.
    - Require a **writable tablet** device
  - Option 2 – Download and print the exam pages, print it out and write on paper. Scan the paper and save into PDF format
- Coverage: Week 1-Week 5 contents (Week 6 contents - analytical modeling & communication primitives - not covered)
- Special note 1: Important - discussion **attendance is required** on Week 5 (Sept 18)!
  - 10-min midterm trial run to make sure all students are prepared for and comfortable with the exam process
- Special note 2:
  - We have created a Piazza poll to collect info regarding your available resources/capabilities to complete the exam with writable tablet. Everyone is **required to participate in the poll**

Outline

- From last class
  - Communication Cost in Parallel Machines
  - Message Passing Machines
    - Routing mechanisms
    - Cut through routing
  - LogP Model
  - Routing in Interconnection Networks
- Today
  - Program and Data Mapping
    - Graph embedding problem
    - Metrics
    - Network model
    - Simulations

Program and Data Mapping (1)

- Parallel Program = Collection of Processes
  +
  Interaction among Processes

- Each process = Computation + Data access

- Classic problem:
  - Given a parallel program, embed (map) processes and data onto a given parallel architecture such that
    - communication cost is minimized
    - overall execution time is minimized

Program and Data Mapping (2)

A simple abstraction

Parallel Program $\xrightarrow{\textbf{Map}}$ Parallel Architecture

$G\ (V, E)$

Node - process

Edge - communication

$G'\ (V', E')$

Node - processor

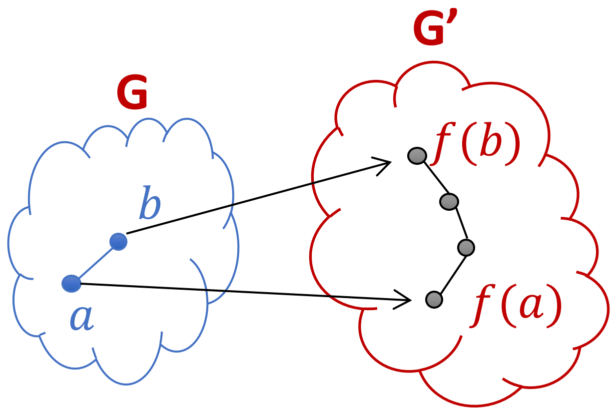
Edge - interconnection

Graph embedding problem

Program and Data Mapping (3)

Graph embedding problem

Given parallel program and parallel architecture



Each edge in E specifies a path in G'

$G(V,E)$ $G'(V',E')$ undirected

Function $f: V \rightarrow V'$

Function $g: E \rightarrow$ paths in G'

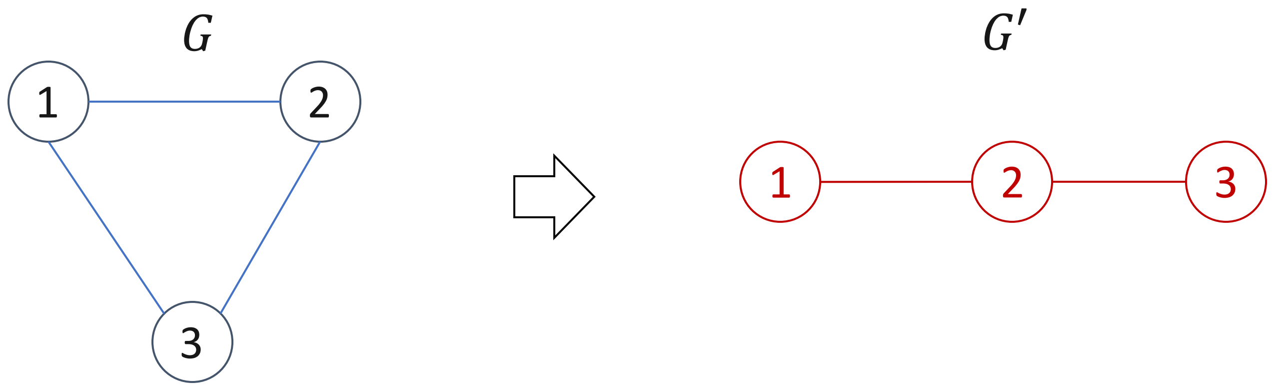
For every edge (a,b) in E, g specifies a path in G', such that the end points in G' are $f(a)$ and $f(b)$

Note: 1. $|V'|$ can be larger than $|V|$

2. A vertex in V' may correspond to more than one vertex in V

Program and Data Mapping (4)

- Example:



Mapping functions:

- $f: i \rightarrow i$
- $g: (1,2) \rightarrow (1,2)$
  $(2,3) \rightarrow (2,3)$
  $(1,3) \rightarrow (1,2), (2,3)$

Program and Data Mapping (5)

Metrics

Congestion: (on edges) Max number of edges in E mapped to an edge in E'

Dilation: Edge in E → Path in E'
Max path length in E' // Max over all edges in E

Expansion: $|V'|/|V|$

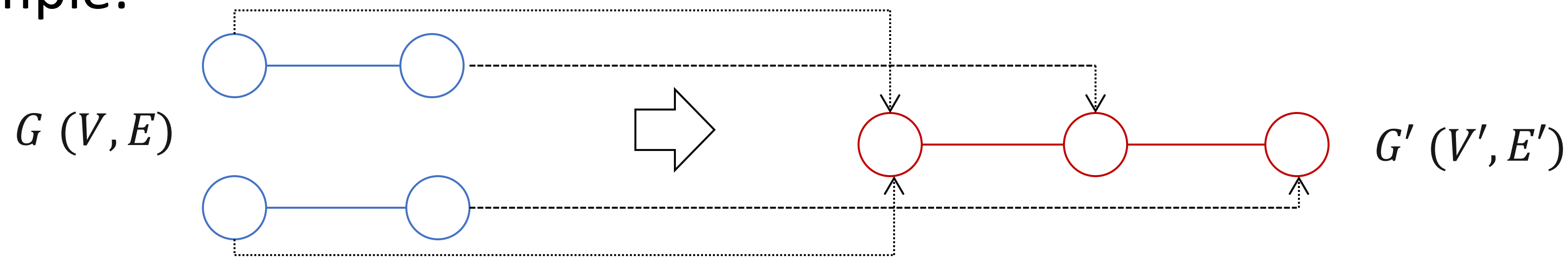
It is also possible $|V'| < |V|$
for example: virtualization

Program and Data Mapping (6)

Congestion: Max number of edges in E (path length in E')
mapped to an edge in E'

$$\text{Congestion} = \max_{e' \in G'} \{\#\text{ of edges} \in G \text{ mapped to } e'\}$$

Example:

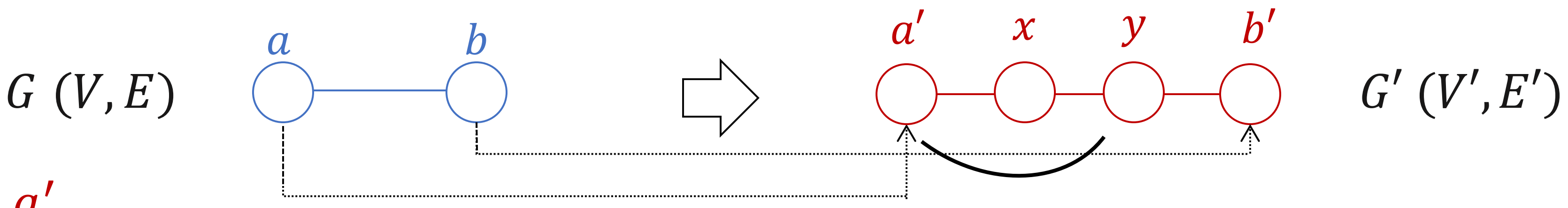


Congestion = 2

Program and Data Mapping (7)

Dilation: maximum number of edges in E' that any edge in E is mapped onto

$$\text{Dilation} = \max_{(a,b)\in E} \{\text{length of path } g(a,b) \text{ in } G'\}$$



$f: a \rightarrow a'$

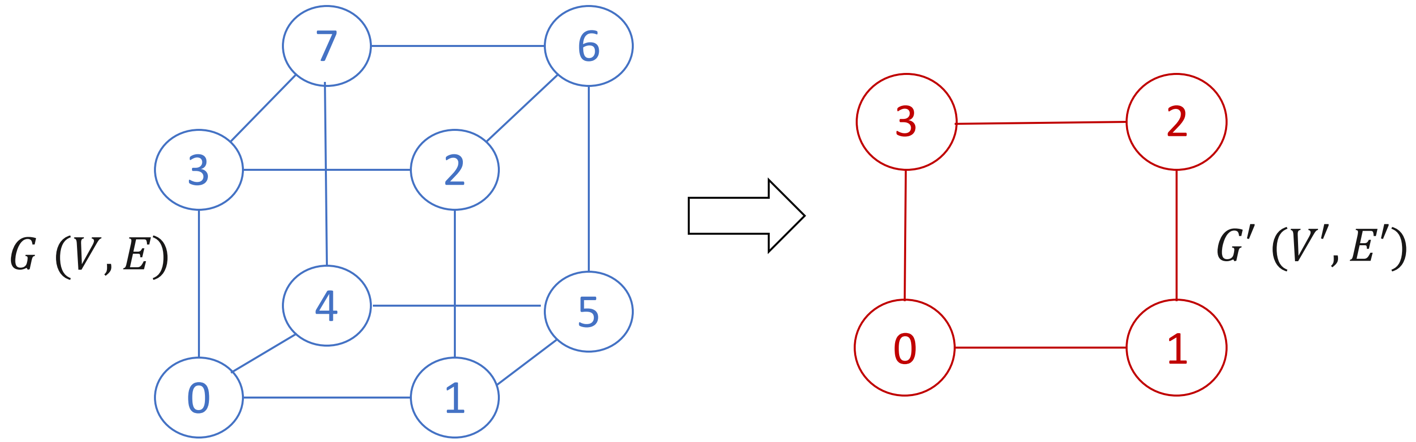
$\quad b \rightarrow b'$

If $g: (a,b) \rightarrow (a',x),(x,y),(y,b')$ ⇨ **Dilation = 3**

If $g: (a,b) \rightarrow (a',y),(y,b')$ ⇨ **Dilation = 2**

Program and Data Mapping (8)

- Example: Collapsed Hypercube



- f: vertex $i, i+4$ in G $\rightarrow$ vertex i in G', $0 \leq i \leq 3$
- g: $(0,1) \rightarrow (0,1)$ $\quad (1,2) \rightarrow (1,2)$
  $(2,3) \rightarrow (2,3)$ $\quad (3,0) \rightarrow (3,0)$
  $(4,5) \rightarrow (0,1)$ $\quad (5,6) \rightarrow (1,2)$
  $(6,7) \rightarrow (2,3)$ $\quad (7,4) \rightarrow (3,0)$

Congestion = 2
Dilation = 1
Expansion = 0.5

Program and Data Mapping (9)

Embedding a Linear Array in Hypercube

A linear array (with wrap around) of size $n = 2^k$ can be embedded in hypercube of size n such that

- Expansion = 1
- Dilation = 1
- Congestion = 1

Program and Data Mapping (10)

Note: Hypercube with $n = 2^k$ nodes

Each node represented using k-bit number

Distance 2^l links $-$ all edges(i, j) $0 \leq i, j < n$, s.t. node j is obtained from node i by complementing bit l, $0 \leq l < k$

Total number of distance 2^l links = $\frac{n}{2}$, $0 \leq l < k$ {also called dimension l link}

Program and Data Mapping (11)

A three bit reflected Gray code ring

1-bit Gray code

0
1

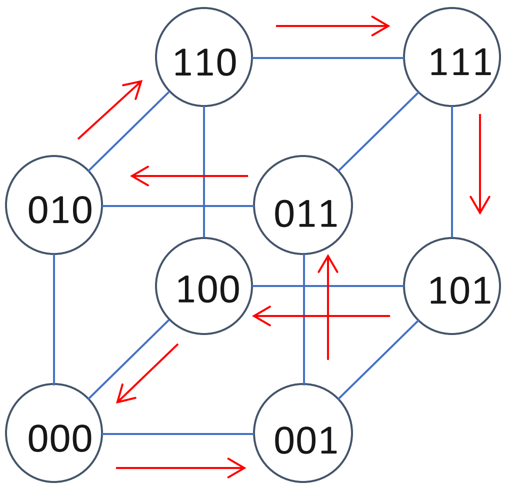
2-bit Gray code

0	0
0	1
1	1
1	0

Reflect along this line

3-bit Gray code	3-D bit hypercube	8-processor ring
0 0 0	0	0
0 0 1	1	1
0 1 1	3	2
0 1 0	2	3
1 1 0	6	4
1 1 1	7	5
1 0 1	5	6
1 0 0	4	7

f



Embedding a 8-node ring into a three-dimensional hypercube

Program and Data Mapping (12)

Embedding a 2^k-node ring into a n-dimensional hypercube

Mapping function

$f: i \rightarrow$ Gray code of i

Let $b_{k-1}, b_{k-2}, \dots, b_1, b_0$ be the binary representation of i,

the Gray code of i is $b_{k-1}, (b_{k-1} \oplus b_{k-2}), \dots, (b_2 \oplus b_1), (b_1 \oplus b_0)$

XOR

g: edge connecting the two end points in hypercube

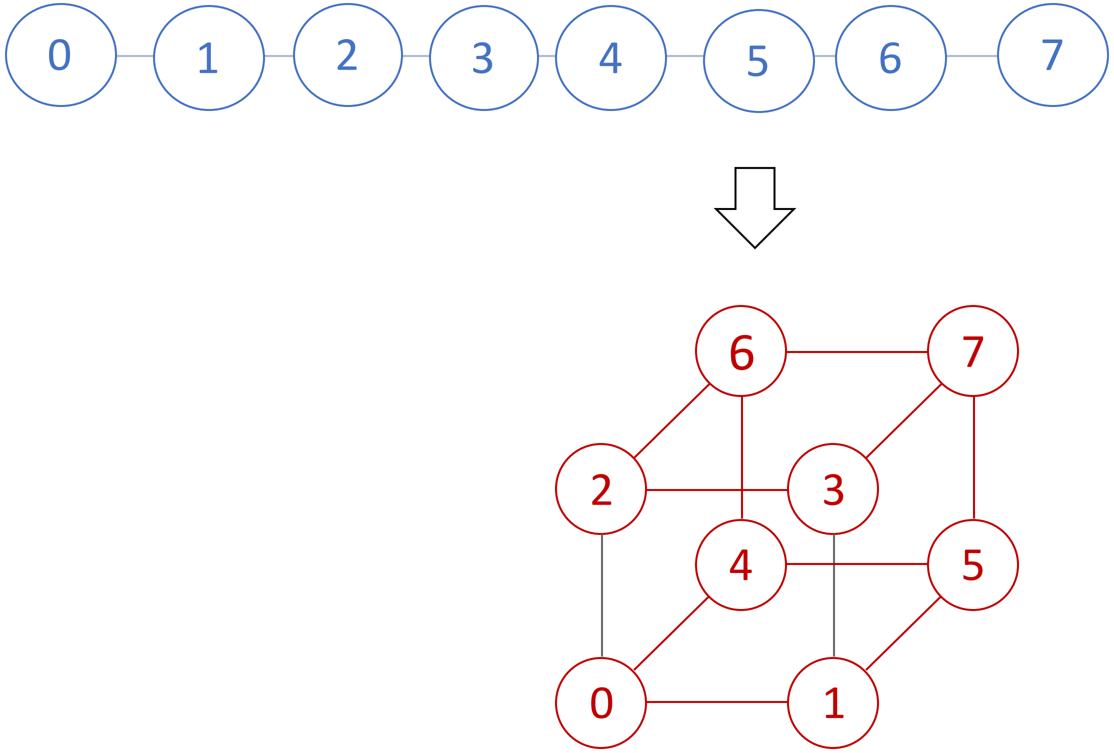
Note: In the Gray code sequence, any two adjacent elements in the sequence are connected by an edge in the hypercube

only one bit changes

Program and Data Mapping (13)

Example for $n = 8$

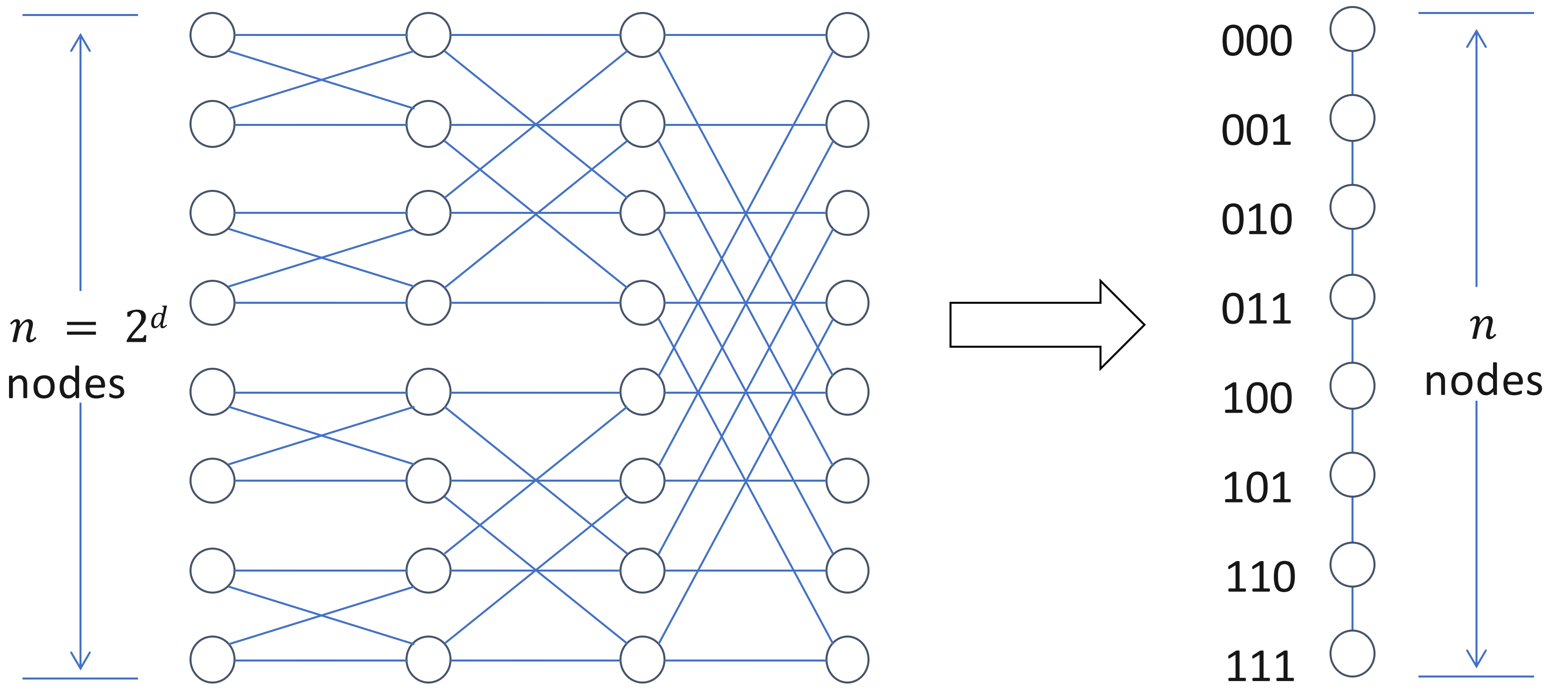
Mapping function



f	g
$0 \rightarrow 0$	$(0,1) \rightarrow (0,1)$
$1 \rightarrow 1$	$(1,2) \rightarrow (1,3)$
$2 \rightarrow 3$	$(2,3) \rightarrow (3,2)$
$3 \rightarrow 2$	$(3,4) \rightarrow (2,6)$
$4 \rightarrow 6$	$(4,5) \rightarrow (6,7)$
$5 \rightarrow 7$	$(5,6) \rightarrow (7,5)$
$6 \rightarrow 5$	$(6,7) \rightarrow (5,4)$
$7 \rightarrow 4$	$(7,0) \rightarrow (4,0)$

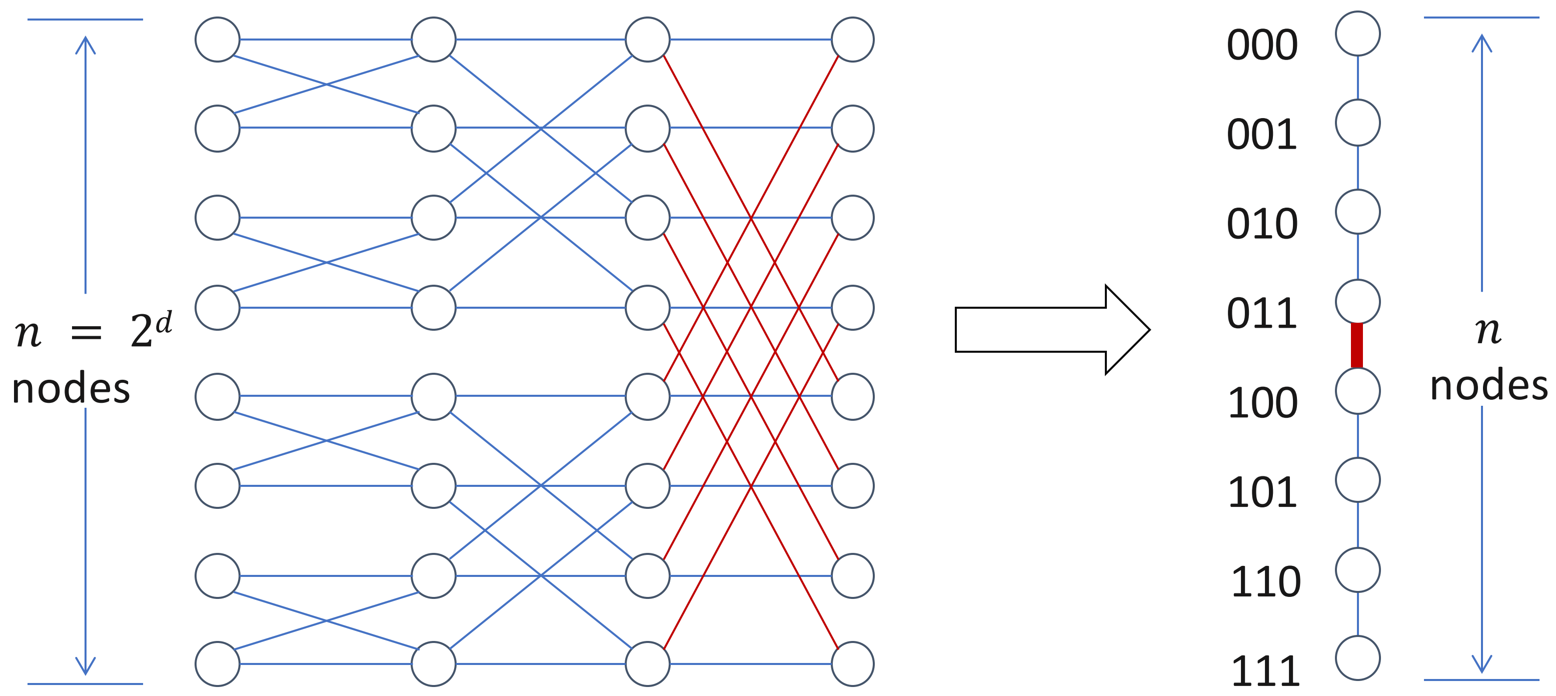
Program and Data Mapping (14)

Example: Embedding butterfly network (FFT computation graph) in a linear array



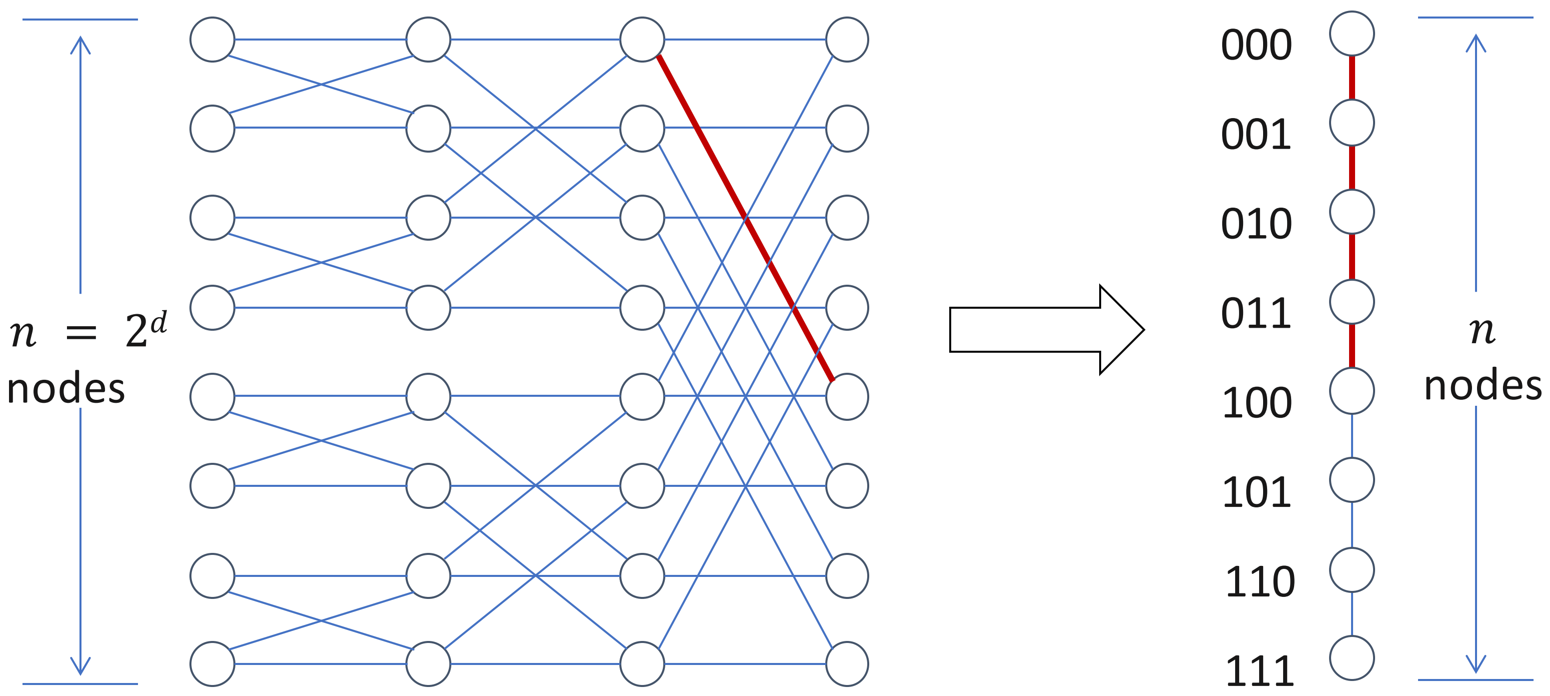
Program and Data Mapping (15)

Congestion $\propto n$



Program and Data Mapping (16)

Dilation = $\frac{n}{2}$

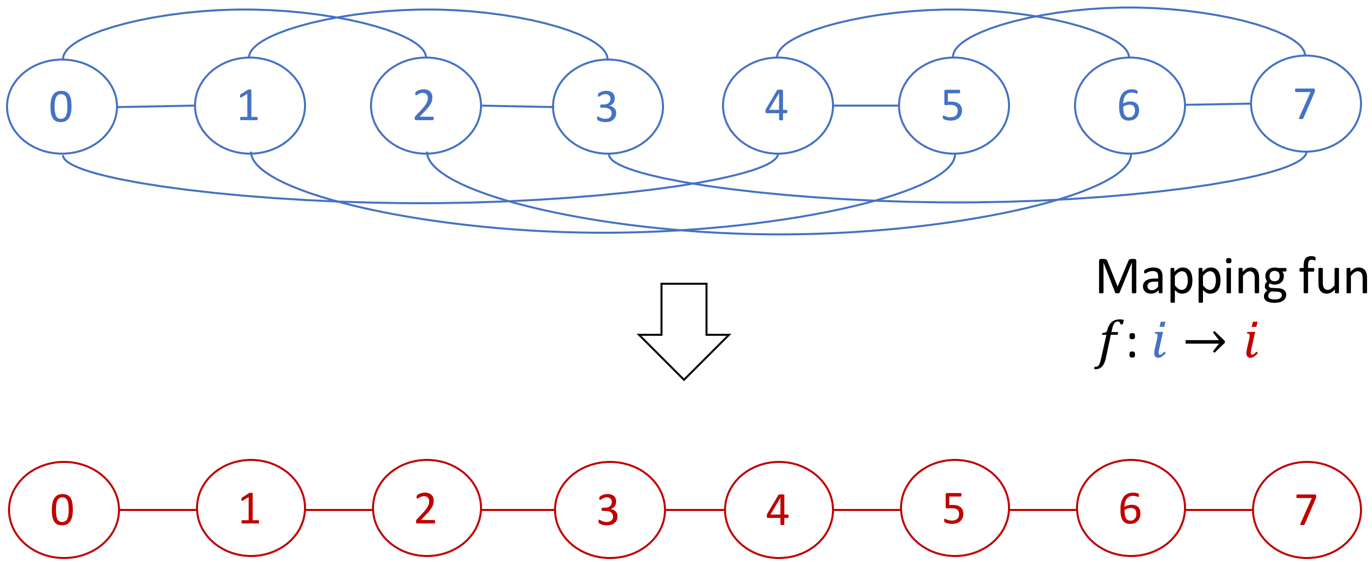


Program and Data Mapping (17)

- n input butterfly network can be embedded in n node 1-D array
  - Congestion $\propto n$
  - Dilation = $\frac{n}{2}$
  - Expansion = $\frac{1}{1+\log_2 n}$

Program and Data Mapping (18)

- Embedding Hypercube in Linear Array



Mapping function
$f: i \rightarrow i$

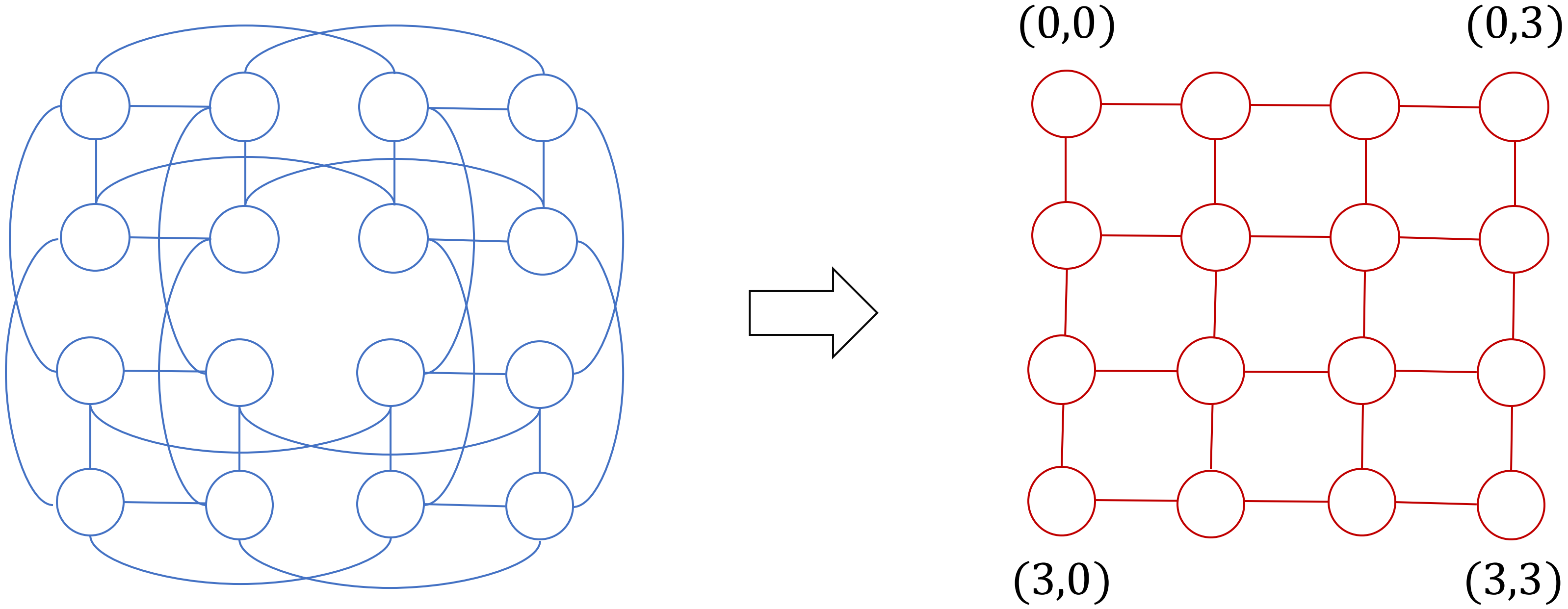
Congestion $\propto \frac{n}{2}$
(edge (3,4))

Dilation = $\frac{n}{2}$

Expansion = 1

Embedding Hypercube on 2-D Mesh (1)

- Embedding Hypercube of size n on a 2-D mesh of size n



Embedding Hypercube on 2-D Mesh (2)

Hypercube of size n $\rightarrow$ 2-D mesh of size n

Congestion $\propto \frac{\sqrt{n}}{2}$

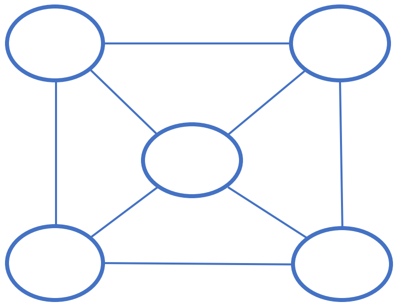
Dilation = $\frac{\sqrt{n}}{2}$

Expansion = 1

Network Model (1)

Shared nothing model

Synchronous execution



p processors + interconnections (static network)

Each step:

In each processor

Compute using data available in the processor

Communicate 1 unit of message along an incident edge

Note: Some models allow communication along all incident edge

{Compute – Communicate} Repeat

Network Model (2)

Computation and communication time

Step: communicate only (no computation)

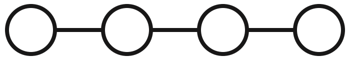
compute time can be < communication time

Network Model (3)

Communication Cost in Network Model

Ex. 1-D Mesh

$i \to i+1,\ \ i-1$



0 1 2 3

Synchronous model

Unit of time:

- Local operation
- Unit data communication using a link

Note: In p processor network model, in each step we can perform p operations in parallel and p communications in parallel.

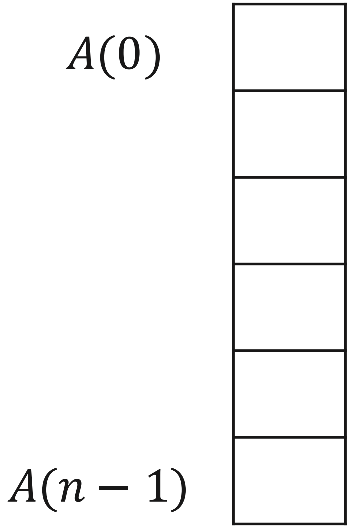
Network Model (4)

Adding n numbers using n processors on hypercube network model

Input: $A(0), \ldots A(n-1)$

$A(i)$ is stored in $P_i, 0 \leq i < n$

Output: $\sum_{i=0}^{n-1} A(i)$ in P_0



Network Model (5)

Adding n numbers using n processors on hypercube network model

- Parallel algorithm

Program in $P_i, 0 \leq i \leq n-1$

Do $j = 0$ to $\log_2 n - 1$

If $i = k \cdot 2^{j+1}$, for some $k \in N$

Receive $A(i + 2^j)$ from P_{i+2^j}

$A(i) \leftarrow A(i) + A(i + 2^j)$

End if

End

Communication

Local computation

$T_{comunication} = O(\log n)$ $T_{computation} = O(\log n)$

Network Model (6)

$n \times n$ Matrix Multiplication using $n \times n$ 2-D mesh (with wrap around)

Systolic Array

$O(n)$ time solution

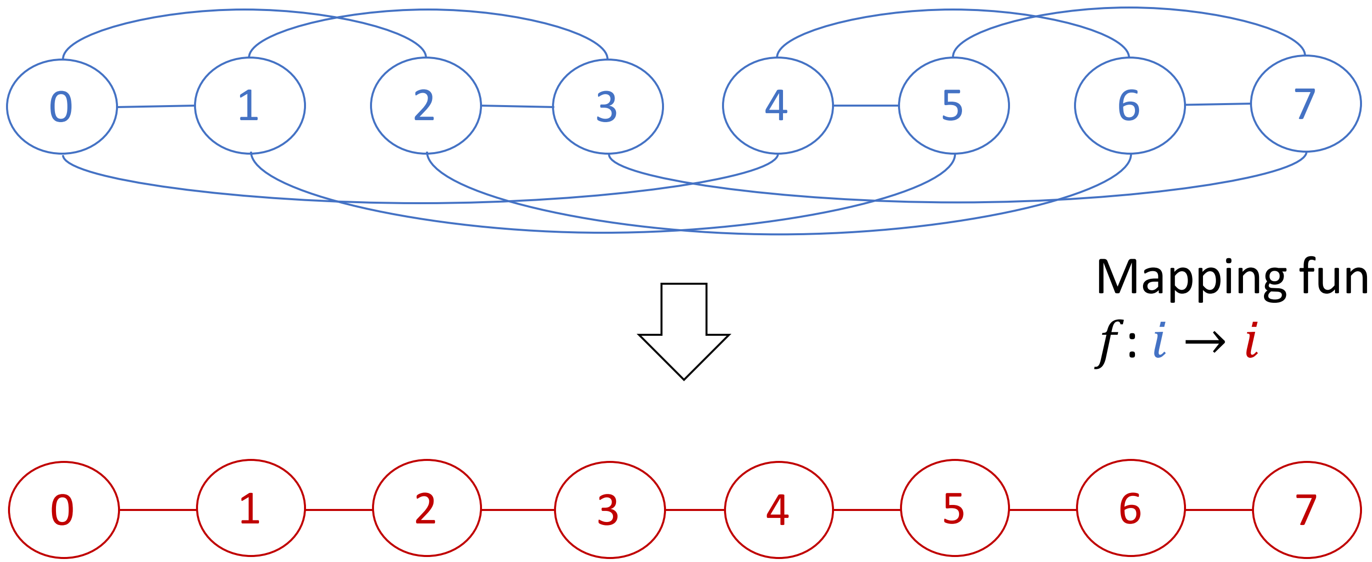
Accelerators

Type of network	# of network layers	# of weights	% of deployed
MLP0	5	20M	61%
MLP1	4	5M	
LSTM0	58	52M	29%
LSTM1	56	34M	
CNN0	16	8M	5%
CNN1	89	100M	

Google TPU

Simulation of Hypercube on 1-D Mesh (1)

Embedding hypercube in 1-D mesh



Mapping function
$f: i \rightarrow i$

Connections with 2^l distance in the hypercube → $O(2^l)$ time on 1-D mesh

Simulation of Hypercube on 1-D Mesh (2)

Does embedding lead to simulating one network on another?

Eg.

Hypercube algorithm $\rightarrow$ 1-D mesh

$T(n)$ Runs in time $T(n) * $ dilation ?

Algorithm running in the network model with hypercube connection

Effect of congestion?

Simulation of Hypercube on 1-D Mesh (3)

Network Model

Hypercube

size $n = 2^k$

$\longrightarrow$

1-D mesh

size n

$\frac{n}{2}$ links along dimension l,
$0 \leq l < k$

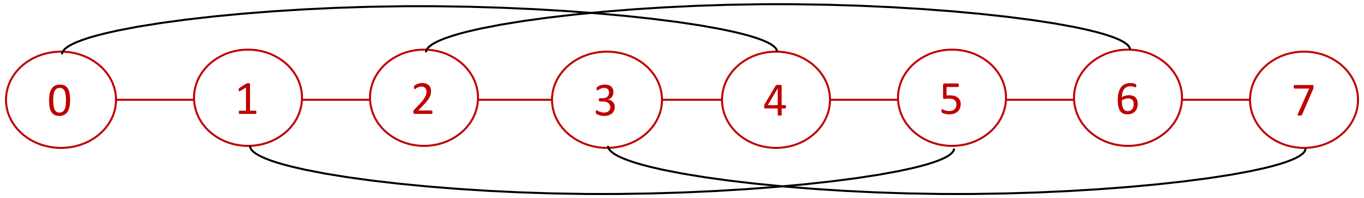
Mapping function
$f: i \rightarrow i, 0 \leq i < n$
$g: (i, j) \rightarrow$ unique path between $f(i)$ and $f(j)$ in 1-D mesh

Simulation of Hypercube on 1-D Mesh (4)

Theorem

In a $n = 2^k$ node hypercube mapped to 1-D mesh, **all** communications in dimension l can be performed in time $O(2^l)$ on the 1-D mesh, $0 \leq l < k$.

Example: $k = 3$, suppose $l = k - 1 = 2$



Pipeline the communication

Note: Total volume of data communicated = $\frac{n}{2}$

Simulation of Hypercube on 1-D Mesh (5)

On n-node hypercube (network model), let,

Computation time = $T^H(n)$

# of communication steps along dimension l = $T_l^H(n)$

Theorem

In the network model, any algorithm on n-node hypercube, n=2^k can be mapped to n-node 1-D mesh such that

Time on hypercube = $T^H(n)$ + $\sum_{l=0}^{k-1} T_l^H(n)$

Time on 1-D mesh = $T^H(n)$ + $\sum_{l=0}^{k-1} T_l^H(n) \cdot 2^l$

Summary

- Program and Data Mapping
  - Graph embedding problem
  - Metrics
    - Congestion
    - Dilation
    - Expansion
  - Network model
  - Simulation
  - Hypercube on 1-D mesh