Cost/Performance Efficient
Interconnection Networks
for Supercomputers

Low performance
High cost
High performance at low cost

Yamin Li, Graduate School of CIS, Hosei University, yamin@hosei.ac.jp
Outline

- Top 10 supercomputers
- Multiprocessor shared-memory parallel systems
- Multicomputer distributed systems
- Problems of interconnection network
- Low-degree and short-diameter topologies
  - Dual-Cube
  - Metacube and KMS-cube
  - RDN — Recursive Dual-Net
- Dynamic interconnection network — MiKANT
- Summary and exercise
## Top 10 Supercomputers

### TOP500 Supercomputer Sites:

[https://www.top500.org/](https://www.top500.org/)

### TOP 10 Supercomputers in June 2023

<table>
<thead>
<tr>
<th>Rank</th>
<th>System</th>
<th>Country</th>
<th>Maker</th>
<th>Cores</th>
<th>TFlop/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Frontier</td>
<td>USA</td>
<td>HPE</td>
<td>8,699,904</td>
<td>1,194.00</td>
</tr>
<tr>
<td>2</td>
<td>Fugaku</td>
<td>Japan</td>
<td>Fujitsu</td>
<td>7,630,848</td>
<td>442.01</td>
</tr>
<tr>
<td>3</td>
<td>LUMI</td>
<td>Finland</td>
<td>HPE</td>
<td>2,220,288</td>
<td>309.10</td>
</tr>
<tr>
<td>4</td>
<td>Leonardo</td>
<td>Italy</td>
<td>Atos</td>
<td>1,824,768</td>
<td>238.70</td>
</tr>
<tr>
<td>5</td>
<td>Summit</td>
<td>USA</td>
<td>IBM</td>
<td>2,414,592</td>
<td>148.60</td>
</tr>
<tr>
<td>6</td>
<td>Sierra</td>
<td>USA</td>
<td>IBM</td>
<td>1,572,480</td>
<td>94.64</td>
</tr>
<tr>
<td>7</td>
<td>Sunway TaihuLight</td>
<td>China</td>
<td>NRCPC</td>
<td>10,649,600</td>
<td>93.01</td>
</tr>
<tr>
<td>8</td>
<td>Perlmutter</td>
<td>USA</td>
<td>HPE</td>
<td>761,856</td>
<td>70.87</td>
</tr>
<tr>
<td>9</td>
<td>Selene</td>
<td>USA</td>
<td>Nvidia</td>
<td>555,520</td>
<td>63.46</td>
</tr>
<tr>
<td>10</td>
<td>Tianhe-2A</td>
<td>China</td>
<td>NUDT</td>
<td>4,981,760</td>
<td>61.44</td>
</tr>
</tbody>
</table>
In general, **parallel systems** are referred to as shared-memory multiprocessors.

- **SMP** — Symmetric multiprocessors
  - Uniform memory access (UMA)
  - Example: *Servers*

- **DSM** — Distributed shared memory
  - Non-uniform memory access (NUMA)
  - Example: *Supercomputers*

In contrast, **distributed systems** are referred to as message-passing multicomputers.

- Cannot access other computer’s memory directly
- Example: Infrastructure of Cloud Computing
Symmetric Multiprocessors (Servers)

- CPUs
- Caches
- Common bus
- Bus arbiter
- Main memory
- Inputs/outputs
Distributed Shared Memory (Supercomputers)

Interconnection network

CPU
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
CPU
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
CPU
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
CPU
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
CPU
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
CPU
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
CPU
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
CPU
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
CPU
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
CPU
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
Cache
Memory
CPU
Interconnection Networks

Ports are connected by links based on a certain topology.
An Implementation of Switch

Data (5 bits) -> 5-32 decoder -> LE, Reset
Address (5 bits) -> LE, Reset
Load
Config, CS
Reset

Inputs

32 load registers
32 configuration registers
32×32 switch matrix (32 32-to-1 multiplexers)

Outputs

Crossbar
A Switch with Self-Routing Function

It can support **wormhole routing**.
Properties of Interconnection Networks

- **Degree**
  - Two nodes are **neighbors** if there is a link connecting them.
  - The **degree** of a node is defined to be the number of its neighbors.
  - Affect hardware cost.

- **Diameter**
  - The **diameter** of a network is defined as the maximum of the shortest distances between any two nodes.
  - Affect communication time.
Popular Topologies

- **Fat-tree**
  - Almost all systems

- **Torus**
  - Fujitsu
  - IBM
  - HPE/Cray

- **Dragonfly**
  - HPE/Cray

- **Hypercubic**
  - SGI
  - Intel
Ring and Completely Connected Networks

(a) Ring  
(b) Completely connected

Symmetric
Mesh Interconnection Networks

(a) 2D mesh

(b) 3D mesh

Not symmetric
Torus Interconnection Networks

(a) 2D torus

(b) 3D torus

Symmetric
Hypercube Interconnection Networks

(a) 0-cube
(b) 1-cube
(c) 2-cube
(d) 3-cube
(e) 4-cube
(f) 5-cube

Symmetric
Tree Interconnection Networks

(a) Tree

(b) Fat-tree

(c) Fat-tree
K-ary N-tree Interconnection Networks

$k = 4$, $n = 3$  
Not symmetric
Groups are fully connected. Within a group: switches are fully connected.
Cost Ratio (CR)

\[
CR(G) = \frac{d(G) \times w_1 + D(G) \times w_2}{\log_2 |(G)|}
\]

d(G): the node degree of G (the number of links of a node)
D(G): the diameter of G
\(|(G)|\): the total number of nodes in G
w_1 and w_2: weights, w_1 + w_2 = 100%

<table>
<thead>
<tr>
<th>INs</th>
<th># Nodes</th>
<th>Degree</th>
<th>Diameter</th>
<th>CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D-Torus (10)</td>
<td>1,000</td>
<td>6</td>
<td>15</td>
<td>1.05</td>
</tr>
<tr>
<td>10-cube</td>
<td>1,024</td>
<td>10</td>
<td>10</td>
<td>1.00</td>
</tr>
<tr>
<td>3D-Torus (128)</td>
<td>2,097,152</td>
<td>6</td>
<td>192</td>
<td>4.72</td>
</tr>
<tr>
<td>21-cube</td>
<td>2,097,152</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
</tr>
</tbody>
</table>

w_1 = w_2 = 0.5
Problems of Interconnection Networks

Problems of Hypercube, 3D Torus, and Tree

Hypercube problem
- The number of links increases logarithmically as the number of nodes increases: \( n = \log_2 N \)

3D Torus problem
- Large (long) diameter:
  \[ D = \left\lceil \frac{X}{2} \right\rceil + \left\lceil \frac{Y}{2} \right\rceil + \left\lceil \frac{Z}{2} \right\rceil \]

Tree problem
- Not symmetric
Suppose a supercomputer has $2^{21} = 2,097,152$ nodes

By Hypercube
- Diameter = 21
- Node-degree = 21 (high cost)

By $128 \times 128 \times 128$ Torus
- Diameter = $\lceil 128/2 \rceil \times 3 = 192$ (too large)
- Node-degree = 6
A router has six links. Two links connect to CPU boards.
An Origin2000 router has six links
Two links connect to two CPU boards
A board contains two CPUs
Cray Router in 5D Origin2000

Yamin Li, CIS, Hosei University

Interconnection Networks – 24 / 85
Low node degree
and
Short diameter

These are conflicting!
Cube-Connected Cycles

Node degree === 3
Hierarchical Cubic Network

Each node \((X, Y)\) is adjacent to

1. \((X, Y^{(k)})\) for all \(1 \leq k \leq n\), where \(Y^{(k)}\) differs from \(Y\) at the \(k\)th bit position,
2. \((Y, X)\) if \(X \neq Y\), and
3. \((\overline{X}, \overline{Y})\) if \(X = Y\), where \(\overline{X}\) and \(\overline{Y}\) are the bitwise complements of \(X\) and \(Y\), respectively.
Dual-Cube
Dual-Cube Interconnection Network

- Node degree: \( m + 1 \)
- Can connect \( N = 2^{2m+1} \) nodes
- Keeps the main properties of hypercube
- Simple routing algorithm
- Is Hamiltonian
- Performs collective communications efficiently
- Low communication cost for matrix multiplication
- Easy to build disjoint paths
- Maximum length of fault-free cycle embedding
- Efficient fault-tolerant routing
Node Address Format of Dual-Cube

Each node has \((m + 1)\) links
- The \(m\) links in Node ID builds a cluster (m-cube)
- One link in Class ID connects to a node in a cluster of the other class
- No links in Cluster ID
- A DC(m) can connect \(2^{2m+1}\) nodes
A Dual-Cube DC(2)
A Dual-Cube DC(3)
It can connect $16 \times 8 \times 2 \times 2 = 512$ CPUs
Metacube
Node Address Format of Metacube

- **class_id**: $c$
- **node_id**: $m_c$
- **cluster_id**: $m_{2^{k-1}}, ..., m_{c+1}, m_{c-1}, ..., m_0$

- An MC($k, m$) can connect $2^{m2^k+k}$ nodes
- Each node has $(m + k)$ links
- Links in $c$ field form high-level $k$-cubes
- Links in $m_c$ field form low-level $m$-cubes
A Metacube MC(2,2)
An Alternative Address Format

Address of k-Cube Oriented Metacube
A k-Cube Oriented Metacube MC(2,1)
An MC(k, m) can connect $2^{m2^k+k}$ nodes. Degree: $m + k$

<table>
<thead>
<tr>
<th>Links/node</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hypercube</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
<td>265</td>
</tr>
<tr>
<td>MC(1, m)</td>
<td>32</td>
<td>128</td>
<td>512</td>
<td>2048</td>
<td>8192</td>
<td>32768</td>
</tr>
<tr>
<td>MC(2, m)</td>
<td>64</td>
<td>1024</td>
<td>16384</td>
<td>$2^{18}$</td>
<td>$2^{22}$</td>
<td>$2^{26}$</td>
</tr>
<tr>
<td>MC(3, m)</td>
<td>–</td>
<td>2048</td>
<td>$2^{19}$</td>
<td>$2^{27}$</td>
<td>$2^{35}$</td>
<td>$2^{43}$</td>
</tr>
<tr>
<td>MC(4, m)</td>
<td>–</td>
<td>–</td>
<td>$2^{20}$</td>
<td>$2^{36}$</td>
<td>$2^{52}$</td>
<td>$2^{68}$</td>
</tr>
</tbody>
</table>

$2^{27} = 134,217,728$
16,384 Nodes: 14-Cube vs MC(2,3)

- **Hypercube**
  - Links: \(14\) links per node
  - \(2^{14} \times 14/2 = 114,688\) links in total
  - Diameters: \(14\)

- **Metacube**
  - Links: \(5\) links per node
  - \(2^{14} \times (3 + 2)/2 = 40,960\) links in total
  - Diameters: \(16\)

- The reduction in the total number of links for this example is \(73,728\) links or about \(64\%\)
- Diameters: only \(2\) more than that of hypercube
MC(2,4) vs 18-Cube vs 3D Torus

- **Metacube(2,4) k = 2, m = 4**
  - The number of nodes: \(2^{18} = 262,144\)
  - Node degree: 6
  - Diameter: 20

- **Hypercube (18-Cube)**
  - The number of nodes: \(2^{18} = 262,144\)
  - Node degree: 18
  - Diameter: 18

- **3D Torus (64 \times 64 \times 64)**
  - The number of nodes: \(2^{18} = 262,144\)
  - Node degree: 6
  - Diameter: \(32 + 32 + 32 = 96\)
Building Origin2000 with MC(2,2)

- An Origin2000 router has six links
- Two links connect to two CPU boards
- A board contains two CPUs
- Four links are used to build an MC(2,2)
  - \( k = 2 \) and \( m = 2 \) (\( k + m = 4 \))
  - There are \( 2^m 2^{k+k} = 2^{10} = 1024 \) nodes
- It can connect \( 1024 \times 2 \times 2 = 4096 \) CPUs
- Does not use the Cray Router anymore
- In contrast, the Origin2000 using Cray Router can only connect \( 32 \times 2 \times 2 = 128 \) CPUs
KMS-Cube
### Node Address of KMS-Cube

The diagram illustrates the address structure of the KMS-Cube. The address is divided into several parts, each labeled with a different variable:

- $b_{h-1,3} \ldots b_{0,3}$
- $b_{h-1,2} \ldots b_{0,2}$
- $b_{h-1,1} \ldots b_{0,1}$
- $c_{k-1} \ldots c_{0}$

Where $h$ and $k$ are the dimensions of the cube, and $s$ is a parameter. The address is structured as follows:

- $h = 2^{k-s}$
- $n = m2^{k-s} + k$

The diagram shows the bits distribution for different values of $m$:

- $m = 0$: $s$ bits
- $m = 1$: $k - s$ bits
- $m = 2$: $k$ bits
- $m = 3$: $h$ bits

The diagram visually represents how the bits are distributed across the different parts of the address.
Links of KMS-Cube

\begin{array}{cccc}
\text{n-1} & b_{h-1,3} \ldots b_{0,3} & b_{h-1,2} \ldots b_{0,2} & b_{h-1,1} \ldots b_{0,1} & c_{k-1} \ldots c_{0} \\
\text{k links} & \text{1 link} & \text{1 link} & \text{1 link} & \text{k links} \\
\text{m links} & b_{c,3} & b_{c,2} & b_{c,1} & c = c_{k-s-1} \ldots c_{0} \\
\end{array}
KMS-Cube with $K = 2, M = 1, \text{ and } S = 1$
KMS-Cube with K = 2, M = 1, and S = 1
KMS-Cube with $K = 2$, $M = 2$, and $S = 1$
Degree Comparison

Node degree vs. Number of nodes in the system for different network architectures:
- Hypercube
- KMS-Cube
- Metacube
- k-ary 2-cube
- k-ary 3-cube
- k-ary 5-cube

Graph shows the relationship between node degree and the number of nodes in the system for various architectures, providing insights into their scalability and connectivity properties.
Diameter Comparison

Number of nodes in the system

Diameter

Hypercube
KMS-Cube
Metacube
k-ary 2-cube
k-ary 3-cube
k-ary 5-cube
Cost Ratio Comparison

Cost ratio ($w_1 = w_2 = 0.5$) vs. Number of nodes in the system

- Hypercube
- KMS-Cube
- Metacube
- k-ary 2-cube
- k-ary 3-cube
- k-ary 5-cube
RDN — Recursive Dual-Net
Recursive Construction of RDN

\[ n_i = n_{i-1} \times n_{i-1} \times 2 = 2n_{i-1}^2 \]

\[ N = \frac{(2m)^{2^k}}{2} \]

where \( m \) is the number of nodes in a base network; \( k \): level
A Recursive Dual-Net RDN(4,1)

RDN($m, k$)

$m$: the number of nodes in the base network; $k$: level

The base network can be an any symmetric network.
A Recursive Dual-Net RDN(4,2)

Cluster 0 of Type 0

Cluster 0 of Type 1

Cluster 31 of Type 0

Cluster 31 of Type 1

RDN(4,1)
RDN(27,2)

- Base network: 3D Torus \((3 \times 3 \times 3)\), \(k = 2\)
  - \(m = 27\) (nodes)
  - \(d_0 = 6\) (degree)
  - \(D_0 = 1 + 1 + 1 = 3\) (diameter)

- The number of nodes
  - \(k = 1\): \(N_1 = 27 \times 27 \times 2 = 1,458\)
  - \(k = 2\): \(N_2 = 1,458 \times 1,458 \times 2 = 4,251,528\)

- \(d = d_0 + k = 6 + 2 = 8\) (degree) \((k = 2)\)
- \(D = 2(2D_0 + 2) + 2 = 18\) (diameter) \((k = 2)\)
- 3D Torus \((162 \times 162 \times 162)\): 4,251,528 nodes; \(d = 6\) (degree)
- 3D Torus \((162 \times 162 \times 162)\): \(D = 81 + 81 + 81 = 243\) (diameter)
RDN(5,3)

- Base network: 5-node-Ring, $k = 3$
  - $m = 5$ (nodes)
  - $d_0 = 2$ (degree)
  - $D_0 = 2$ (diameter)

- The number of nodes
  - $k = 1$: $N_1 = 5 \times 5 \times 2 = 50$
  - $k = 2$: $N_2 = 50 \times 50 \times 2 = 5,000$
  - $k = 3$: $N_3 = 5,000 \times 5,000 \times 2 = 50,000,000$

- $d = d_0 + k = 2 + 3 = 5$ (degree) ($k = 3$)
- $D = 2[2(2D_0 + 2) + 2] + 2 = 30$ (diameter) ($k = 3$)
- 3D Torus ($500 \times 500 \times 200$): 50,000,000 nodes; $d = 6$ (degree)
- 3D Torus ($500 \times 500 \times 200$): $D = 250 + 250 + 100 = 600$ (diameter)
Cost Ratio (CR)

\[
CR(G) = \frac{d(G) \times w_1 + D(G) \times w_2}{\log_2 |(G)|}
\]

<table>
<thead>
<tr>
<th>INs</th>
<th># Nodes</th>
<th>Degree</th>
<th>Diameter</th>
<th>CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D-Torus (10)</td>
<td>1,000</td>
<td>6</td>
<td>15</td>
<td>1.05</td>
</tr>
<tr>
<td>10-cube</td>
<td>1,024</td>
<td>10</td>
<td>10</td>
<td>1.00</td>
</tr>
<tr>
<td>RDN(5^2, 1)</td>
<td>1,250</td>
<td>5</td>
<td>10</td>
<td>0.73</td>
</tr>
<tr>
<td>RDN(3^3, 1)</td>
<td>1,458</td>
<td>7</td>
<td>8</td>
<td>0.71</td>
</tr>
<tr>
<td>3D-Torus (128)</td>
<td>2,097,152</td>
<td>6</td>
<td>192</td>
<td>4.72</td>
</tr>
<tr>
<td>21-cube</td>
<td>2,097,152</td>
<td>21</td>
<td>21</td>
<td>1.00</td>
</tr>
<tr>
<td>DC(10)</td>
<td>2,097,152</td>
<td>11</td>
<td>22</td>
<td>0.79</td>
</tr>
<tr>
<td>RDN(5^2, 2)</td>
<td>3,125,000</td>
<td>6</td>
<td>22</td>
<td>0.65</td>
</tr>
<tr>
<td>RDN(3^3, 2)</td>
<td>4,251,528</td>
<td>8</td>
<td>18</td>
<td>0.59</td>
</tr>
<tr>
<td>RDN(5, 3)</td>
<td>50,000,000</td>
<td>5</td>
<td>30</td>
<td>0.68</td>
</tr>
<tr>
<td>Network</td>
<td>Nodes</td>
<td>Degree</td>
<td>Diameter</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------</td>
<td>--------</td>
<td>-------------------------------</td>
<td></td>
</tr>
<tr>
<td>3D Torus</td>
<td>$n^3$</td>
<td>6</td>
<td>$3\lceil n/2 \rceil$</td>
<td></td>
</tr>
<tr>
<td>n-cube</td>
<td>$2^n$</td>
<td>$n$</td>
<td>$n$</td>
<td></td>
</tr>
<tr>
<td>CCC(n)</td>
<td>$n2^n$</td>
<td>3</td>
<td>$2n + \lceil n/2 \rceil - 2$</td>
<td></td>
</tr>
<tr>
<td>HCN(n)</td>
<td>$2^{2n}$</td>
<td>$n+1$</td>
<td>$n + \lceil (n+1)/3 \rceil + 1$</td>
<td></td>
</tr>
<tr>
<td>Dual-Cube(m)</td>
<td>$2^{2m+1}$</td>
<td>$m+1$</td>
<td>$2(m+1)$</td>
<td></td>
</tr>
<tr>
<td>Metacube(k, m)</td>
<td>$2^{m2^k+k}$</td>
<td>$m+k$</td>
<td>$(m+1)2^k$</td>
<td></td>
</tr>
<tr>
<td>RDN(m, k)</td>
<td>$(2m)^{2^k}/2$</td>
<td>$d_0 + k$</td>
<td>$2^kD_0 + 2^{k+1} - 2$</td>
<td></td>
</tr>
</tbody>
</table>
Degree Comparison

Number of nodes in system vs. Node degree for various network topologies:
- hypercube
- dual-cube
- quad-cube
- oct-cube
- hex-cube
- 3D-torus
- rdn(torus, 1)
- rdn(torus, 2)
- rdn(cube, 2)
Diameter Comparison

<table>
<thead>
<tr>
<th>System Type</th>
<th>Diameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hypercube</td>
<td></td>
</tr>
<tr>
<td>Dual-cube</td>
<td></td>
</tr>
<tr>
<td>Quad-cube</td>
<td></td>
</tr>
<tr>
<td>Oct-cube</td>
<td></td>
</tr>
<tr>
<td>Hex-cube</td>
<td></td>
</tr>
<tr>
<td>3d-torus</td>
<td></td>
</tr>
<tr>
<td>rdn(torus,1)</td>
<td></td>
</tr>
<tr>
<td>rdn(torus,2)</td>
<td></td>
</tr>
<tr>
<td>rdn(cube,2)</td>
<td></td>
</tr>
</tbody>
</table>

Number of nodes in system

Yamin Li, CIS, Hosei University
Interconnection Networks – 62 / 85
Cost Ratio Comparison

- hypercube
- dual-cube
- quad-cube
- oct-cube
- hex-cube
- 3d-torus
- rdn(torus,1)
- rdn(torus,2)
- rdn(cube,1)
- rdn(cube,2)

Number of nodes in system vs. Cost ratio

- $2^4$, $2^8$, $2^{12}$, $2^{16}$, $2^{20}$, $2^{24}$, $2^{28}$, $2^{32}$, $2^{36}$, $2^{40}$
MiKANT — Mirrored K-Ary N-Tree
Dynamic IN: Fat Tree (3-ary 3-tree)

Root switches have fewer ports than others
Bidirectional Clos Network

High switch cost
High link cost
Long distance
Mirrored K-Ary N-Tree MiKANT(3,3)
Switch Address Format of MiKANT(n, k)

Value: [0, n – 2]

Value: [0, k – 1]

G: Group ID; L: Level (Stage) ID; D: Switch ID
Node Address Format of MiKANT(n, k)

Value: [0, k – 1]

G: Group ID; C: Node ID; C_{n-2} \ldots C_0: Switch ID
A switch

\[ \langle G, L, D_{n-2}, \ldots, D_{L+1}, D_L, D_{L-1}, \ldots, D_0 \rangle \]

will connect to switches

\[ \langle G, L + 1, D_{n-2}, \ldots, D_{L+1}, *, D_{L-1}, \ldots, D_0 \rangle \]

if \( 0 \leq L \leq n - 3 \); otherwise (\( L = n - 2 \)) to switches

\[ \langle \overline{G}, L, *, D_{n-3}, \ldots, D_1, D_0 \rangle \]
Switch-Compute Node Connections

There is a link between a switch

$$\langle G, 0, D_{n-2}, \ldots, D_1, D_0 \rangle$$

and a compute node

$$\langle G, C_{n-1}, C_{n-2}, \ldots, C_1, C_0 \rangle$$

if $$D_i = C_i$$ for all $$i \in \{n-2, \ldots, 1, 0\}$$. 
Mirrored K-Ary N-Tree MiKANT(3,4)
## Comparison of Topological Properties

<table>
<thead>
<tr>
<th></th>
<th>Classical k-ary n-tree</th>
<th>Clos k-ary n-tree</th>
<th>Mirrored k-ary n-tree</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong># nodes</strong></td>
<td>$k^n$</td>
<td>$2k^n$</td>
<td>$2k^n$</td>
</tr>
<tr>
<td><strong># switches</strong></td>
<td>$nk^{n-1}$</td>
<td>$(2n-1)k^{n-1}$</td>
<td>$(2n-2)k^{n-1}$</td>
</tr>
<tr>
<td><strong># links</strong></td>
<td>$nk^n$</td>
<td>$2nk^n$</td>
<td>$(2n-1)k^n$</td>
</tr>
<tr>
<td><strong>Degree</strong></td>
<td>$2k$</td>
<td>$2k$</td>
<td>$2k$</td>
</tr>
<tr>
<td><strong>Diameter</strong></td>
<td>$2n$</td>
<td>$2n$</td>
<td>$2n$</td>
</tr>
<tr>
<td><strong>Bisection</strong></td>
<td>$k^n/2$</td>
<td>$k^n/2$</td>
<td>$k^n/2$</td>
</tr>
<tr>
<td><strong>Ave. dist</strong></td>
<td>$2n - \frac{2}{k-1} + \frac{2}{(k-1)k^n}$</td>
<td>$2n - \frac{1}{k-1} + \frac{1}{(k-1)k^n}$</td>
<td>$2n - \frac{1}{k-1} + \frac{1}{(k-1)k^n} - \frac{1}{2}$</td>
</tr>
</tbody>
</table>
Cost Ratios of Links and Switches

Switches links of classical k-ary n-tree

Links, Mirrored k-ary n-tree

Switches, Mirrored k-ary n-tree
Performance Improvement of MiKANT

![Graph showing performance improvement of Mirrored k-ary n-tree vs Classical k-ary n-tree over different n values. The graph indicates a decrease in performance improvement as n increases from 2 to 8.](image-url)
Relative Cost Performance to Hypercube

- n-cube
- k-ary 2-tree
- k-ary 3-tree
- k-ary 4-tree
- k-ary 5-tree
- k-ary 6-tree
- k-ary 7-tree
- k-ary 8-tree

Number of compute nodes in the system
Average Packet Latencies (Clock Cycles)

- Average packet latency (cycles)
- Traffic load $\lambda$ (packets/cycle/node)
- Bidir. Clos Uniform
- MiKANT Uniform
- Bidir. Clos Inversion
- MiKANT Inversion

Graph showing the relationship between average packet latency and traffic load for different interconnection network configurations.
Successful routing ratio vs. Number of faulty switches

- **Go-back three**
- **Go-back two**
- **Go-back one**
- **All-port**
- **One-port**
- **Shortest path**

Yamin Li, CIS, Hosei University
Fault Tolerance — Path Length

Average path length

Number of faulty switches

Go-back three
Go-back two
Go-back one
All-port
One-port
Shortest path
Successful routing ratio vs. Switch faulty rate

- Go-back three
- Go-back two
- Go-back one
- All-port
- One-port
- Shortest path
- S/D switch faulty
Fault Tolerance — Path Length

Average path length vs. Switch faulty rate

- Go-back three
- Go-back two
- Go-back one
- All-port
- One-port
- Shortest path
The IN is at the center of supercomputers.

It is built with switches (routers) and cables that connect ports of switches by following some topologies.

Switches: Infiniband, Gigabit Ethernet, or custom

A good interconnection network should use a small number of links (low cost of switch) and have a short diameter (fast communication).

IN topologies for building supercomputers

- Dual-Cube, Metacube, and KMS-Cube
- RDN — Recursive Dual-Net
- MiKANT — Mirrored K-Ary N-Tree
Research Topics on Parallel Systems

- Shortest-path routing algorithm
- Multicast and Broadcast algorithms
- Collective communication
- Disjoint paths and Hamiltonian
- Fault-tolerant routing
- Algorithmic design
  - Parallel prefix computation
  - Parallel sorting
- Matrix multiplication and Linpack (Linear algebra)
- Parallel (programming languages and) compilers

Performance Evaluation
Define Cost Ratio $\text{CR}(G) = \frac{d(G) \times w_1 + D(G) \times w_2}{\log_2 |(G)|}$

- $d(G)$: the node degree of $G$ (the number of links of a node)
- $D(G)$: the diameter of $G$
- $|(G)|$: the total number of nodes in $G$

Question: Let $w_1 = w_2 = 50\%$. Calculate CRs for

(a) A 256-node Ring
(b) A 256-node Completely Connected Network
(c) A 256-node 8-Cube
(d) A 256-node $16 \times 16$ Two-Dimensional Torus
(e) A 512-node $8 \times 8 \times 8$ Three-Dimensional Torus
(f) A 512-node Dual-Cube DC(4)
(g) A 512-node Mirrored 4-ary 4-tree MiKANT(4, 4)

Report submission: [https://hoppii.hosei.ac.jp/portal](https://hoppii.hosei.ac.jp/portal)
Recommendations and CFP (CANDAR23)

- Format of conference/journal papers
  - Use **\LaTeX** to prepare the manuscript
  - Draw block diagrams with **\texttt{tgif}**
  - Generate graphs with **\texttt{Gnuplot}**

- Related international conference
  - CANDAR 2023, Nov. 28 - Dec. 1, Matsue
    - The Eleventh International Symposium on Computing and Networking

  - Abstract submission due: July 20, 2023
  - Paper (PDF) submission due: July 25, 2023