Finding a Hamiltonian Cycle in a Hierarchical Dual-Net with Base Network of p-Ary q-Cube

Yamin Li, Department of Computer Science, Hosei University, yamin@hosei.ac.jp
Shietung Peng, Department of Computer Science, Hosei University, speng@hosei.ac.jp
Wanming Chu, Department of Computer Hardware, University of Aizu, w-chu@u-aizu.ac.jp
Hierarchical Dual-Net (HDN)
  Why
  What
Finding a Hamiltonian Cycle in an HDN
  How
Interconnection Networks

- The modern high-performance supercomputers consist of hundreds of thousands of CPUs
- In the near future, the number of CPUs in supercomputers will reach several millions
- How to connect these extremely large number of CPUs is an important issue for achieving high performance of the supercomputers
- A “good” interconnection network should use a small number of links and meanwhile keep the diameter as shorter as possible
- The symmetric structure and efficient routing should also be considered
Interconnection Networks

- Ring
  - Too simple, long diameter
- Complete graph
  - Too complex, expensive
- 3D Torus
  - Long diameter in a large-scale system
- Hypercubes
  - The number of links increases logarithmically
- Tree / Fat Tree
  - Not symmetric
The Hierarchical Dual-Net (HDN)

- Use a small popular IN as *base network*
- Hierarchical structure
- Small and flexible node-degree
- Short diameter
- Efficient routing and broadcasting
- Flexible scale (small – very large systems)
Building an HDN(B, i, S) from HDN(B, i – 1, S)

HDN(B, i, S):

B is a base network; k is the level of the HDN;
S = \{s_1, s_2, \ldots, s_k\} where s_i is the number of nodes in a super-node at the level i.
An HDN(B,1,S) with $s_1 = 2$ (Base = 2-Cube)
An HDN(B,2,S) with $s_1 = 2$ and $s_2 = 4$
Select a symmetric product graph as a base network

For example,
- hypercube,
- torus, or
- p-ary q-cube

a 5-ary 2-cube
The Number of Nodes in HDN(B,k,S)

Suppose that the base network is a 2 * 3 * 5 torus

<table>
<thead>
<tr>
<th>Degree = 7</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>k = 1</td>
<td>s₁ = 1</td>
<td>s₁ = 2</td>
<td>s₁ = 3</td>
<td>s₁ = 5</td>
<td>s₁ = 6</td>
<td>s₁ = 10</td>
<td>s₁ = 15</td>
<td>s₁ = 30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1,800</td>
<td>900</td>
<td>600</td>
<td>360</td>
<td>300</td>
<td>180</td>
<td>120</td>
<td>60</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Degree = 8</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>k = 2</td>
<td>s₂ = 1</td>
<td>s₂ = 2</td>
<td>s₂ = 3</td>
<td>s₂ = 5</td>
<td>s₂ = 6</td>
<td>s₂ = 10</td>
<td>s₂ = 15</td>
<td>s₂ = 30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s₁ = 1</td>
<td>6,480,000</td>
<td>3,240,000</td>
<td>2,160,000</td>
<td>1,296,000</td>
<td>1,080,000</td>
<td>648,000</td>
<td>432,000</td>
<td>216,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s₁ = 2</td>
<td>1,620,000</td>
<td>810,000</td>
<td>540,000</td>
<td>324,000</td>
<td>270,000</td>
<td>162,000</td>
<td>108,000</td>
<td>54,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s₁ = 3</td>
<td>720,000</td>
<td>360,000</td>
<td>240,000</td>
<td>144,000</td>
<td>120,000</td>
<td>72,000</td>
<td>48,000</td>
<td>24,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s₁ = 5</td>
<td>259,200</td>
<td>129,600</td>
<td>86,400</td>
<td>51,840</td>
<td>43,200</td>
<td>25,920</td>
<td>17,280</td>
<td>8,640</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s₁ = 6</td>
<td>180,000</td>
<td>90,000</td>
<td>60,000</td>
<td>36,000</td>
<td>30,000</td>
<td>18,000</td>
<td>12,000</td>
<td>6,000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s₁ = 10</td>
<td>64,800</td>
<td>32,400</td>
<td>21,600</td>
<td>12,960</td>
<td>10,800</td>
<td>6,480</td>
<td>4,320</td>
<td>2,160</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s₁ = 15</td>
<td>28,800</td>
<td>14,400</td>
<td>9,600</td>
<td>5,760</td>
<td>4,800</td>
<td>2,880</td>
<td>1,920</td>
<td>960</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s₁ = 30</td>
<td>7,200</td>
<td>3,600</td>
<td>2,400</td>
<td>1,440</td>
<td>1,200</td>
<td>720</td>
<td>480</td>
<td>240</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Topological Properties

<table>
<thead>
<tr>
<th>Network</th>
<th>The number of nodes (N)</th>
<th>Degree (d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D Torus</td>
<td>$x \times y \times z$</td>
<td>6</td>
</tr>
<tr>
<td>n-cube</td>
<td>$2^n$</td>
<td>n</td>
</tr>
<tr>
<td>CCC(n)</td>
<td>$n \times 2^n$</td>
<td>3</td>
</tr>
<tr>
<td>Dual-Cube(n)</td>
<td>$2^{2n-1}$</td>
<td>n</td>
</tr>
<tr>
<td>RDN(m, k)</td>
<td>$(2m)^{2^k}/2$</td>
<td>$d_0 + k$</td>
</tr>
<tr>
<td>HDN(B, k, S)</td>
<td>$(2</td>
<td>B</td>
</tr>
<tr>
<td>Network</td>
<td>Diameter (D)</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>3D Torus</td>
<td>((x + y + z)/2)</td>
<td></td>
</tr>
<tr>
<td>n-cube</td>
<td>(n)</td>
<td></td>
</tr>
<tr>
<td>CCC(n)</td>
<td>(2n + \lfloor n/2 \rfloor - 2)</td>
<td></td>
</tr>
<tr>
<td>Dual-Cube(n)</td>
<td>(2n)</td>
<td></td>
</tr>
<tr>
<td>RDN(m, k)</td>
<td>(2^k \times D_0 + 2^{k+1} - 2)</td>
<td></td>
</tr>
<tr>
<td>HDN(B, k, S)</td>
<td>(2^k (D(B) - \sum_{j=0}^{k-1} 2^j (D(SN^{k-j})) + 2^{k+1} - 2))</td>
<td></td>
</tr>
</tbody>
</table>
**Weighted Cost Ratio**

Define Weighted Cost Ratio: 
\[ CR_w(G) = \frac{w_1 \times d(G) + w_2 \times D(G)}{\log_2 |(G)|} \]

B: 2 * 3 * 5 torus; w_1 = w_2 = 50%

<table>
<thead>
<tr>
<th>Network</th>
<th>N</th>
<th>d</th>
<th>D</th>
<th>CR_w</th>
</tr>
</thead>
<tbody>
<tr>
<td>10-cube</td>
<td>1,024</td>
<td>10</td>
<td>10</td>
<td>1.00</td>
</tr>
<tr>
<td>3D-Tori(10)</td>
<td>1,000</td>
<td>6</td>
<td>15</td>
<td>1.05</td>
</tr>
<tr>
<td>HDN(B, 1, (1))</td>
<td>1,800</td>
<td>7</td>
<td>10</td>
<td>0.79</td>
</tr>
<tr>
<td>HDN(B, 1, (2))</td>
<td>900</td>
<td>7</td>
<td>9</td>
<td>0.82</td>
</tr>
<tr>
<td>HDN(B, 1, (3))</td>
<td>600</td>
<td>7</td>
<td>9</td>
<td>0.87</td>
</tr>
</tbody>
</table>
Define Weighted Cost Ratio:

$$\text{CR}_w(G) = \frac{w_1 \times d(G) + w_2 \times D(G)}{\log_2|\langle G \rangle|}$$

B: 2 * 3 * 5 torus; $w_1 = w_2 = 50\%$

<table>
<thead>
<tr>
<th>Network</th>
<th>N</th>
<th>d</th>
<th>D</th>
<th>CR$_w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>19-cube</td>
<td>524,288</td>
<td>19</td>
<td>19</td>
<td>1.00</td>
</tr>
<tr>
<td>3D-Tori(80)</td>
<td>512,000</td>
<td>6</td>
<td>120</td>
<td>3.32</td>
</tr>
<tr>
<td>HDN(B, 2, (2, 2))</td>
<td>810,000</td>
<td>8</td>
<td>19</td>
<td>0.69</td>
</tr>
<tr>
<td>HDN(B, 2, (2, 5))</td>
<td>324,000</td>
<td>8</td>
<td>18</td>
<td>0.71</td>
</tr>
<tr>
<td>HDN(B, 2, (5, 2))</td>
<td>129,600</td>
<td>8</td>
<td>17</td>
<td>0.74</td>
</tr>
</tbody>
</table>
Linear array and ring are two fundamental networks and many algorithms were designed based on linear array and ring

Thus embedding linear array or ring in networks is important for emulating those algorithms

A Hamiltonian cycle of an undirected graph $G$ is a simple cycle that contains every node in $G$ exactly once

A Hamiltonian path in a graph is a simple path that visits every node exactly once

A graph that contains a Hamiltonian cycle is said to be Hamiltonian
A p-ary q-cube connects $p^q$ nodes
- $p$: the number of nodes per dimension
- $q$: the network dimension

Each node can be identified by a $q$-digit radix-$p$ address $(a_0, a_1, \ldots, a_i, \ldots, a_{q-1})$

There is a link connecting node A with address $(a_0, a_1, \ldots, a_{q-1})$ and node B with address $(b_0, b_1, \ldots, b_{q-1})$ if and only if there exists $i$ ($0 \leq i \leq q - 1$) such that $a_i = (b_i + 1) \% p$ and $a_j = b_j$ for $0 \leq j \leq q - 1$ and $i \neq j$

For examples, a 4-ary 2-cube is a $4 \times 4$ torus and a 2-ary 4-cube is a 4-cube
A Hamiltonian cycle in a 5-ary 2-cube

(a) 5-ary 2-cube

(b) A cycle
Lemma: There is a Hamiltonian cycle in a p-ary q-cube
There are $2n_1$ clusters and each cluster contains $n_1$ super-nodes.

Building a virtual Hamiltonian cycle (VHC)
- An HVC connects all $2n_1$ clusters
- Two super-nodes in a cluster are in the VHC
- Including all super-nodes
  - Replacing the two super-nodes with the Hamiltonian path
- Renaming super-nodes
- Including all nodes
  - Expanding from super-node level to node level
A Hamiltonian Cycle for Super-Nodes

(a) A virtual Hamiltonian cycle connecting all clusters

(b) Including and renaming all super-nodes
A Hamiltonian Cycle for All Nodes in HDN(B,1,(4))
Building A Hamiltonian Cycle in Level 2

- Build an VHC connecting all clusters at level 2
- Insert all super-nodes to the cycle
- Expand super-nodes to nodes
  - There is a problem: some nodes cannot be included in super-node i
  - To solve this problem, we route in super-node i first

Lemma: For an HDN with the base network of p-ary q-cube, given any node \( u \in \) renamed super-node \( i \) in a cluster, for \( i = 0, 1, \ldots, n - 1 \), there is a path \( u \rightarrow v \) containing all of the nodes in the cluster with \( v \in \) renamed super-node \( (i + 1) \% n \)
Some Nodes in Super-Node i Can Not Be Included

Cross-edge of level 2

Super-node i

\[(i + 1) \% n\]

Cross-edge of level 1

\[(i - 1) \% n\]

\[(i + 2) \% n\]
Routing in Super-Node $i$ First

Cross-edge of level 2

Super-node $i$ -- $\cdots$ -- $w$, $u$ -- $\cdots$ -- $v$ -- $(i + 1)\% n$

Cross-edge of level 1

$(i - 1)\% n$ -- $\cdots$ -- $(i + 2)\% n$
Part of Hamiltonian cycle at level 2

Level 1 super-node id

Cluster 0

Class 0

Level 2 super-node id
A Hamiltonian cycle in HDN(B,2,(4,4))
A Hamiltonian cycle in HDN(B,2,(4,16))
Algorithm 1: HDN_HC(HDN(B, k, S))

begin
    $p_{c0}$ = Hamiltonian cycle of base network;
    for $j \leftarrow 1$ to $k$ do /* $k$ levels */
        group nodes to super-nodes based on $s_j$;
        $n_j$ = the number of super-nodes in HDN(B, $k - 1$, S);
        based on the cycle build at level $j - 1$,
        rename the super-node_id in HDN(B, $j - 1$, S) such
        that super-nodes $i$ and $(i + 1)\%n_j$ are neighbors;
        $u = 0$; /* starting node_id in a super-node */
        for $i \leftarrow 0$ to $n_j - 1$ do /* $n_j$ clusters of each class */
            /* build a Hamiltonian path in cluster $i$ of class 0 */
            /* based on $j - 1$ level Hamiltonian cycle $p_{c_{j-1}}$ */
            $hp^0_i$ = (0, i, i, u) → (0, i, $(i + 1)\%n_j$, v);
            /* build a Hamiltonian path in cluster $(i+1)\%n_j$ of class 1 */
            /* based on $j - 1$ level Hamiltonian cycle $p_{c_{j-1}}$ */
            $hp^1_i$ = (1, $(i + 1)\%n_j$, i, v) → (0, i, $(i + 1)\%n_j$, w);
            $u = w$; /* end node_id → starting node_id of next cluster */
        endfor
    /* we get the $j$ level Hamiltonian cycle $p_{c_j}$ */
    $p_{c_j}$ = $\emptyset$;
    for $i \leftarrow 0$ to $n_j - 1$ do /* $n_j$ clusters of each class */
        $p_{c_j}$ = $p_{c_j}$ $\cup$ $hp^0_i$ $\cup$ $hp^1_i$;
    endfor
end
Conclusions

- The Hierarchical dual-net, HDN, has many attractive properties including:
  - Small and flexible node-degree
  - Short diameter
  - Efficient routing and broadcasting

- We presented an algorithm for finding a Hamiltonian cycle in HDN with the base network of p-ary q-cube

- The HDN can be used as an efficient interconnection network of a supercomputer of the future generation