

Parallel Programming

Exercise sheet 2: Topologies

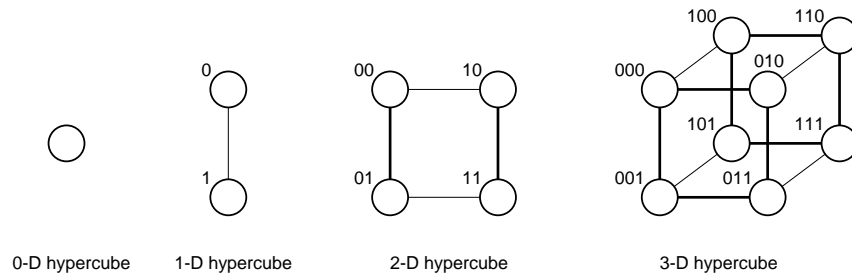
Issued at 10/05/07, to be discussed at 15/05/07

1 Meshes

Given is a 2-D mesh with size $m \times n$, the nodes are labelled from 1 to m in x_1 -direction and from 1 to n in x_2 -direction. Consider two arbitrary nodes $n_1 = (x_1^1, x_2^1)$ and $n_2 = (x_1^2, x_2^2)$. How many different shortest paths from node n_1 to node n_2 do exist? Give a general formula for calculating the amount of different shortest paths from two arbitrary nodes n_1 to n_2 , depending only of $x_1^1, x_2^1, x_1^2, x_2^2, m$, and n .

2 Hypercubes

The hypercube topology has two nodes along each dimension and $\log_2(n)$ dimensions. The construction of a hypercube goes as follows, in general a d -dimensional hypercube is constructed by connecting corresponding nodes of two $(d - 1)$ dimensional hypercubes. It is useful to derive a numbering scheme for nodes in a hypercube. A simple numbering scheme can be derived from the construction of a hypercube. If we have a numbering of two subcubes of $n/2$ nodes, we can derive a numbering scheme for the cube of n nodes by prefixing the labels of one of the subcubes with a “0” and the labels of the other subcube with a “1”.



- a) Given is a k -dimensional hypercube. Consider two arbitrary nodes $n_1 = x_1^1 x_2^1 \dots x_k^1$ and $n_2 = x_1^2 x_2^2 \dots x_k^2$. How many different shortest paths from node n_1 to node n_2 do exist? Give a general formula for calculating the amount of

different shortest paths from two arbitrary nodes n_1 to n_2 , depending only of $x_1^1 x_2^1 \dots x_k^1, x_1^2 x_2^2 \dots x_k^2$, and k .

- b) Consider a 3-D hypercube and 4-D hypercube. There are different shortest paths from node 000 to node 111 in 3-D and from node 0000 to node 1111 in 4-D. What is the minimum amount of processing nodes that have to be “short-circuited” in 3-D and 4-D, thus, only **one** shortest path remains. Illustrate your solution in a small sketch!
- c) Considering the construction algorithms of k -dimensional meshes and hypercubes, show the relationship between a $k-2$ mesh and a k hypercube.

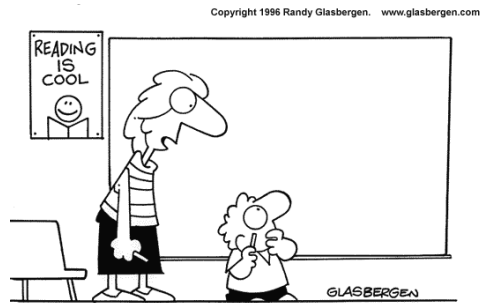
3 Network evaluation: mesh vs. hypercube

The bisection width of a network is defined as the minimum number of communication links that must be removed to partition the network into two equal halves. The bisection bandwidth of a network is defined as the minimum volume of communication allowed between any two halves of the network. It is the product of the bisection width and the channel bandwidth. Bisection bandwidth of a network is also sometimes referred to as cross-section bandwidth.

- a) Given is a 3-D mesh with wraparound (torus) with size $8 \times 8 \times 16$ (1024 processors). Furthermore, every communication link in this network runs in full-duplex mode at a speed of 400 MBytes/second. Every message sent through this network – assumed there’s no congestion – takes 0.5 ns to be passed from one processing node to another.
- i. Calculate the bisection bandwidth of this network. There are multiple ways for a bisection!
 - ii. Calculate the diameter of this network.
 - iii. Calculate the maximum latency – the time it takes for a message to get from one designated point to another – of this network.
- b) Given is also a 10-D hypercube (1024 processors). Furthermore, every communication link in this network runs in full-duplex mode at a speed of 120 MBytes/second. Every message sent through this network – assumed there’s no congestion – takes 0.7 ns to be passed from one processing node to another.
- i. Calculate the bisection bandwidth of this network.
 - ii. Calculate the diameter of this network.
 - iii. Calculate the maximum latency of this network.

- c) Calculate the cost of both networks described in a) and b) to make a decision dependent on a cost-benefit ratio which topology to choose for a 1024 processor machine, when we want a big bandwidth with a low latency and a low cost. Discuss your decision!

Have fun!



"There aren't any icons to click. It's a chalk board."