Suggested Solutions – Exercises of Lecture 7 – MVP'06

Alexandre David

Exercise 4.9

In a hypercube of dimension 3, given any node A, all the nodes at the shortest distance 2 are all the nodes that differ from A with 2 bits (this comes directly from the E-cube routing algorithm). Since we have 3 bits, the possible ways of negating 2 bits of $A = b_1b_2b_3$ are $\bar{b_1}\bar{b_2}b_3$, $\bar{b_1}b_2\bar{b_3}$, and $b_1\bar{b_2}\bar{b_3}$. We have 3 nodes. The interesting point is to have a more general expression of the shortest distance between any pair of nodes (i, j). This is the number of bits that differ between i and j, i.e., the Hamming distance between i and j (noted $H_{i,j}$). The first question could have been re-phrased as how many nodes in a hypercube of dimension 3 are at a Hamming distance 2 from a fixed node. Now that we have a way to measure the distance between two nodes, we can count how many nodes are l-links away from a fixed node in a hypercube of dimension $d = \log(p)$, which is the general version of the first question. The answer is the number of ways to negate l bits in a d-bits number (with $l \leq d$), which is the number of ways to choose l bits in a set of d bits (known combinatorial common problem). That's $\binom{d}{l} = \frac{d!}{l!(d-l)!} = \frac{\log(p)!}{l!(\log(p)-l)!}$.

Exercise 4.11

Let's consider a 2-D mesh without wrap-around connections. Notice that the algorithm uses the ring procedure for the rows and the columns but that's fine since we have the bidirectional link free to close the ring. The number of words transmitted by every p processor is $m(p-1)$. The average path length taken by the packets is $\sqrt{p}$. This gives us the total traffic $p\sqrt{p}m(p-1)$. The number of links (counting the bidirectional ones) is $4p$. The time lower bound is then $T_{low} = t_w\frac{\sqrt{p}m(p-1)}{4}$. We have then $\frac{t_w mp(\sqrt{p}-1)}{T_{low}} = 4\frac{\sqrt{p}}{\sqrt{p}+1} \leq 4$ using the fact that $(p-1) = (\sqrt{p}-1)(\sqrt{p}+1)$.

Exercise 4.14

Sparse 3-D meshes were not discussed in this chapter and they would have the same cost-effectiveness than 3-D meshes since by removing edges you gain in cost but you loose in congestion. The comparison was meant between 3-D meshes and hypercubes. Number of links in 3-D meshes $= 3p - 3p^{2/3}$ ($3p$ for wrap-around and you remove the edges connecting the 3 opposite faces). Number of links in hypercubes $= \frac{p\log(p)}{2}$. If $t_s = 0$ then the simplified communication times are given in Table . To get the times for the 3-D mesh you can extend the algorithms given for 2-D mesh. For the cases where the communication time is the same, the 3-D mesh is more cost effective than the hypercube since it is cheaper. The all-to-all personalized and the circular shift are in favor of the hypercube since we have to compare $p\log(p)$ to $p^{\frac{4}{3}}$ in both cases. Intuitively, this is to be expected.

Exercise 4.15

If we consider sparse 3-D mesh with infinite bandwidth ($t_w = 0$) then congestion does not matter and the sparse 3-D mesh wins. Now we can do the same as in the previous exercise and recompute the formulas and compare them but since congestion does not matter, we can map the best possible algorithm to either the hypercube or the 3-D mesh without effective congestion, which means the 3-D mesh always wins since it is cheaper.

	Hypercube	3-D mesh
One-to-all broadcast/ All-to-one reduction	$t_w m\log(p)$	$t_w m\log(p)$
All-to-all broadcast/ All-to-all reduction	$t_w m(p-1)$	$t_w m(p-1)$
All-reduce	$t_w m\log(p)$	$t_w m\log(p)$
Scatter/gather	$t_w m(p-1)$	$t_w m(p-1)$
All-to-all personalized	$t_w m(p-1)$	$\frac{3}{2}t_w mp(p^{\frac{1}{3}}-1)$
Circular shift	$t_w m$	$\frac{3}{2}t_w m(p^{\frac{1}{3}}+1)$

Table 1: Cost effectiveness for different algorithms on a 3-D mesh and a hypercube.

Exercise 4.22

The proof is by induction as it is suggested in the exercise. The first hint (if $q > p/2$ then a q-shift is isomorphic to a $(p-q)$-shift means that the shifts are symmetric.

As the base of induction, it is easy to see that the statement is true for a 2-processor hypercube. Let all the p data paths be congestion-free in a p-processor hypercube for all $q < p$. In a $2p$-processor hypercube (induction step), if q-shifts for all $q < p$ are congestion-free then q-shifts for all $q < 2p$ are also congestion-free (hint 1). The proof is complete if we show that q-shifts for all $q < p$ are congestion-free in a $2p$-processor hypercube (these are the missing cases but also the ones added by the induction step).

Consider a circular q-shift for any $q < p$ on a $2p$-processor hypercube. All the $p-q$ data paths leading from processor i to a processor j such that $i < j < p$ are the same as in a p-processor hypercube, and hence, by the induction hypothesis, do not conflict with each other (we are stay on a sub-cube with p processors). The remaining q data paths leading from a processor i to a processor j on the p-processor hypercube, such that $j < i < p$, lead to processor $j+p$ on a $2p$-processor hypercube. processor $j+p$ is connected to processor j by a single link in the highest dimension (by construction of the hypercube, that's the $(\log(p)+1)^{\text{th}}$ dimension) of the $2p$-processor hypercube. Thus, following the E-cube routing, the data path from processor i to processor $j+p$ in a circular q-shift on the $2p$-processor hypercube is the data path from processor i to processor j in a circular q-shift on a p-processor hypercube appended by a single link. The original path from processor i to j is congestion free (induction hypothesis) and the last link is not shared by any other message because it is unique for each message. Thus q-shifts are congestion-free in a $2p$-processor hypercube. □

Exercise 4.25

Cost of network = total number of links. The number of links in a 2-D mesh with wrap around is $2p$ and for a hypercube is $\frac{p\log(p)}{2}$. We have $s = \frac{\log(p)}{4}$. The communication times for the hypercube are in Table 4.1 of the book (page 187), they are the base reference with $t_w = 1$. For a 2-D wrap-around mesh and each link $(\log(p)/4)$-channel wide, the communication times are as follows:

$$T_{one2all_broadcast} = t_s\log(p) + 4m$$

$$T_{all2all_broadcast} = 2t_s(\sqrt{p}-1) + \frac{4m(p-1)}{\log(p)}$$

$$T_{one2all_personalized} = 2t_s(\sqrt{p}-1) + \frac{4m(p-1)}{\log(p)}$$

$$T_{all2all_personalized} = 2t_s(\sqrt{p}-1) + \frac{4mp(\sqrt{p}-1)}{\log(p)}$$

A mesh is asymptotically more cost-effective than a hypercube for all operations except all-to-all personalized communication. The result is the same as in exercise 4.14.

Cost of network = bisection width. The bisection width of a 2-D mesh with wrap-around is $2\sqrt{p}$ and for a hypercube is $\frac{p}{2}$. We have $s' = \frac{\sqrt{p}}{4}$. The communication times on a 2-D wrap-around mesh with $(\sqrt{p}/4)$-channel wide are:

$$T_{one2all_broadcast} = t_s \log(p) + \frac{4m \log(p)}{\sqrt{p}}$$

$$T_{all2all_broadcast} = 2t_s(\sqrt{p} - 1) + 4m\sqrt{p}$$
$$T_{one2all_personalized} = 2t_s(\sqrt{p} - 1) + 4m\sqrt{p}$$
$$T_{all2all_personalized} = 2t_s(\sqrt{p} - 1) + 4mp$$

A mesh is asymptotically more cost-effective than a hypercube except for all-to-all personalized communication where both interconnection networks have the same asymptotic cost-performance characteristics.

Exercise 4.26

Interestingly, these operations take the same amount of time on a hypercube and a completely connected network. Obviously the added connectivity does not improve anything for these operations, which may seem counter-intuitive.