6.896
3/31/04
L13.1

"Ideal" parallel computer (Slides 2-3)
Problem: # wires = $\Theta(N^2)$ bad
degree = $\Theta(N)$ bad
diameter = $\Theta(1)$ good

Implement as low-degree network (Slides 4-8)
$N \times N$ mesh of trees:
# switches = $\Theta(N^2)$ bad
degree = $\Theta(1)$ good
diameter = $\Theta(\lg N)$ good

<u>Direct</u> network: every node is a processor
<u>Indirect</u> network: processors + switches (inputs/outputs)

<u>Routing on $N \times N$ MOT</u>
N messages at row roots
Route to column roots.
- Assume perm, since otherwise <u>hotspot</u> could make any network look bad.
Time = $\Theta(\lg N)$ — but lots of hardware.

<u>Hypercube</u> (Slides 9-10)

Routing: flip any bit that's wrong by routing on that dimension.

$\underline{1}\,\underline{0}\,1\,\underline{1}\,1\,\underline{0}\,1\,0 \rightarrow 0\,1\,1\,0\,1\,1\,1\,0$

Bitwise XOR of current msg location and dest.
$1\,1\,0\,1\,0\,1\,0\,0 \rightarrow 0\,0\,0\,0\,0\,0\,0\,0$
But, msgs may collide.
Also, degree = $\lg N$.

6.896
3/31/04
L13.2

Cube-connected cycles (Slide 11)

$N = n \lg n$ nodes
degree $= \Theta(1)$
diam $= \Theta(\lg N)$

Butterfly (FFT) network (Slides 12-13)

n inputs, n outputs   «direct vs. indirect»
$N = n \lg n$ nodes
$\Theta(1)$ degree
Diameter $= \Theta(\lg N)$ «little tricky if not I or O»

Isomorphic to CCC, but authors didn't realize.

Decomposing a butterfly (Slides 14-24)

Remove major cycles $\Rightarrow$ 2 $n/2$-input butterflies
" minor cycles $\Rightarrow$ " " " "

Routing on butterfly (Slide 25)

- Just like hypercube, but uses a specific order of dimensions

$\left\{ \begin{array}{l} \text{dest} = 0 \Rightarrow \text{go up} \\ \phantom{\text{dest} = {}} 1 \Rightarrow \text{go down} \end{array} \right\}$ or $\left\{ \begin{array}{l} \text{XOR} = 0 \Rightarrow \text{straight} \\ \phantom{\text{XOR} = {}} 1 \Rightarrow \text{cross} \end{array} \right\}$

Tree embeddings in butterfly (Slides 26-27)

- CBT rooted at each input
- CBT " " " output

6.896
3/31/04
L13.3

Packet routing on butterfly

source $\qquad\qquad$ dest

$x_{d-1} x_{d-2} \cdots x_0 \rightarrow y_{d-1} y_{d-2} \cdots y_0$

Route major to minor:

$x_{d-1} x_{d-2} \cdots x_0$

$y_{d-1} x_{d-2} \cdots x_0$ $\qquad$ $d = \lg n$ steps, but might have congestion!

$y_{d-1} y_{d-2} \cdots x_0$

$\vdots$

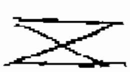
$y_{d-1} y_{d-2} \cdots y_0$

n packets on n-input butterfly
What is worst-case perm?
- $\sqrt{n}$ packets at sources $x_1 x_2 x_3 x_4 0000$ go to dests $0000 x_1 x_2 x_3 x_4$

All go through line 00000000 halfway through network $\Rightarrow$ congestion $= \sqrt{n}$.

Beneš network (Slides 28–29)

Thm. Any n-perm can be routed (off-line) on an n-input Beneš with node-disjoint paths.

Pf. Induction on n.
Base ($N=2$): $\begin{smallmatrix}0 \\ 1\end{smallmatrix}$ or $\begin{smallmatrix}1 \\ 0\end{smallmatrix}$ 

Inductive case (Slides 31–39) $\boxtimes$

Corollary A dn-input Beneš network can simulate any n-node, degree-d network in $O(\lg(dn))$ time. $\boxtimes$

Bounded-degree $\Rightarrow O(\lg n)$ time

Beneš network is $O(\lg n)$-universal for offline simulation of bounded-degree networks.
«Analogy to universal Turing machines»