Understanding Fat-Trees: Revolutionizing Supercomputing Networks

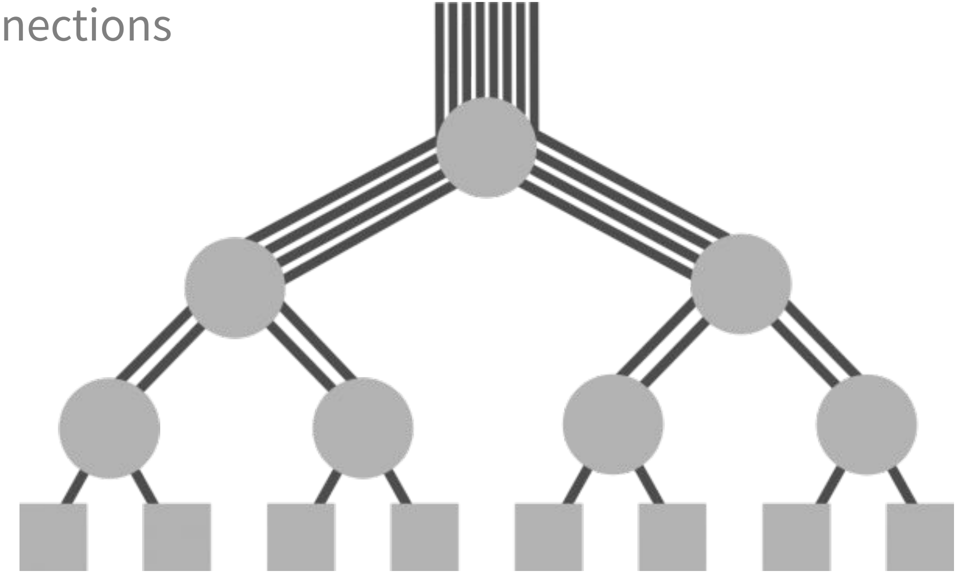
Eddie Brees, James Osborne

The Problem...

- The year is 1985
- Processors are near if not entirely all serial
  - Nary a multicore chip in sight
- Parallel drastically speeds up most applications
- How can we get a number of processors to talk to one another, at a reasonable cost and speed?

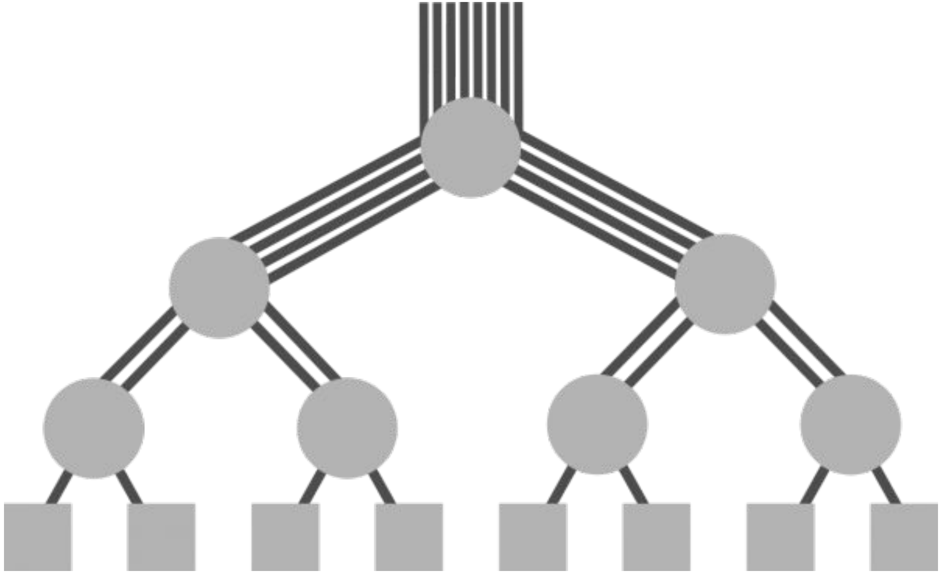
Fat-Trees

- Circles are routers
- Rectangles are processors
- As a binary tree but with more connections
- Takes 2^d parts to construct before connections
- One lane for each processor



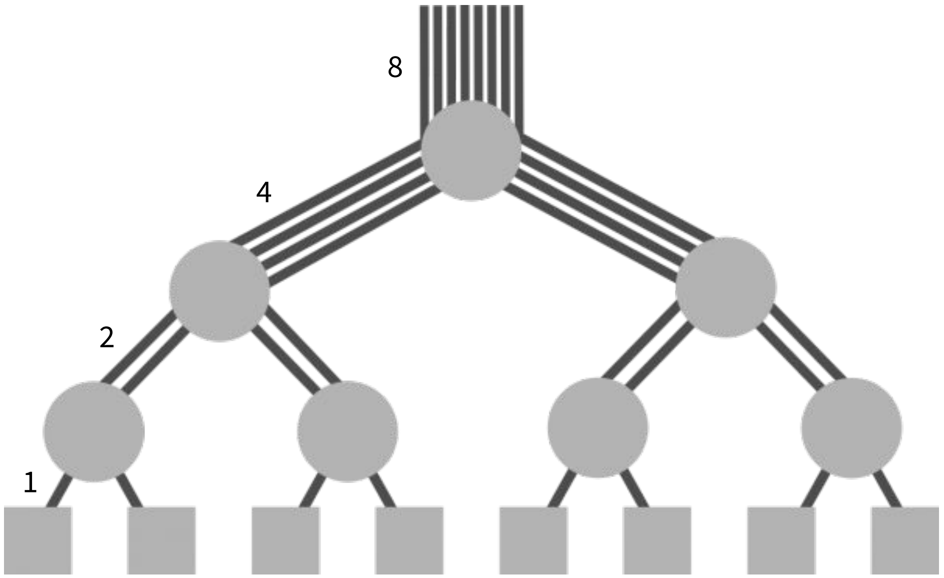
Parameters and Bandwidth

- n - number of processors
- d - depth
- k - layer
- c - connections



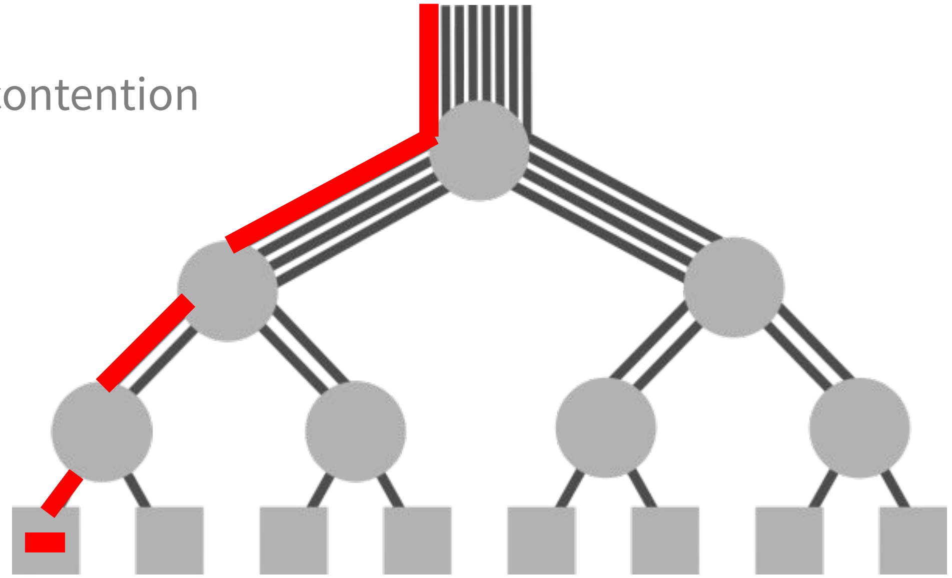
The Relationship of Bandwidth and Depth

- Where d is depth from bottom to top
- Start numbering at 0
- $d = 2^k$



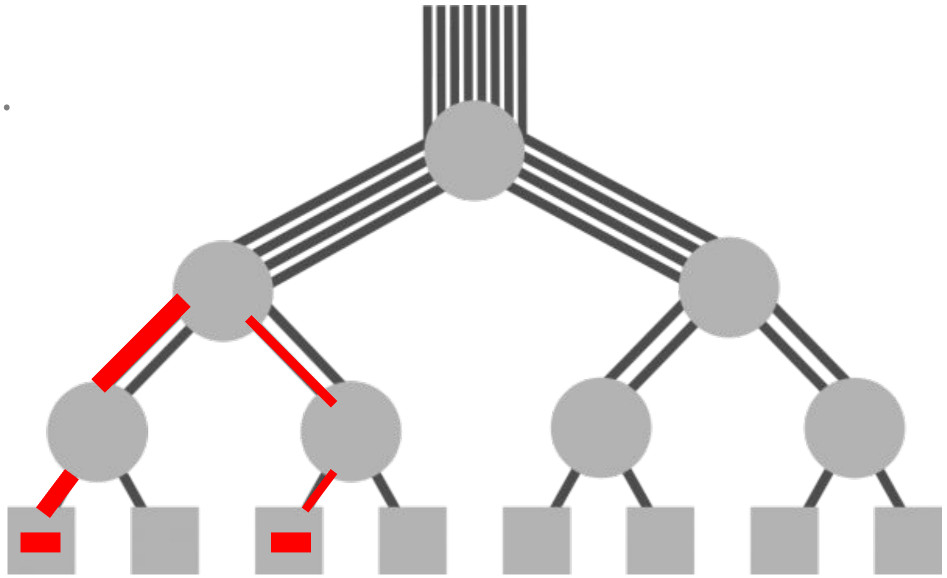
The Path Taken

- Every pair of processors has a unique path.
- Path ascends to the nearest common ancestor, then descends.
- Efficient routing ensures minimal contention and uniform bandwidth.



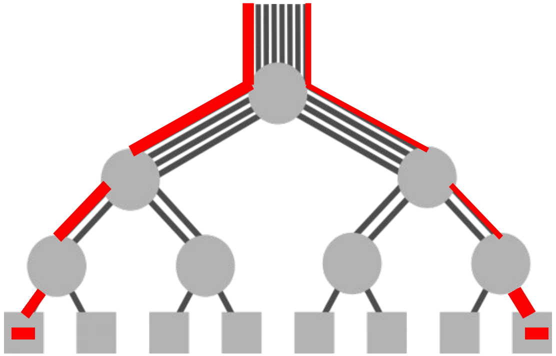
The Uniform Bandwidth

- Every pair of processors can communicate with one another at a consistent rate.
- Ensures fairness in communication.
- Predictable and efficient parallel processing.

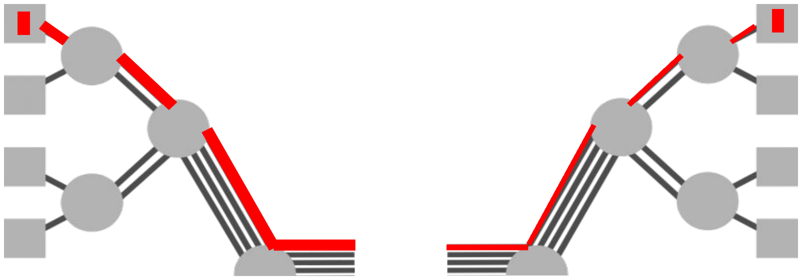


A Bisection Bandwidth

- Ensures that if the network is divided into two equal parts, both halves can communicate with each other at maximum speed.
- Even as processors work on different tasks, communication remains unhindered and efficient.

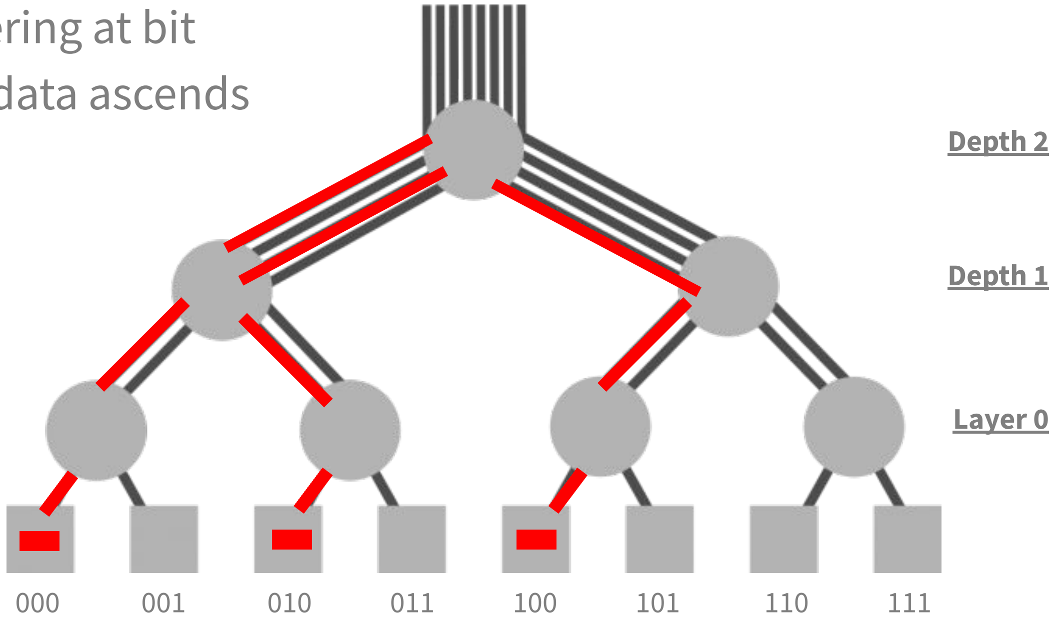


Is equal to…



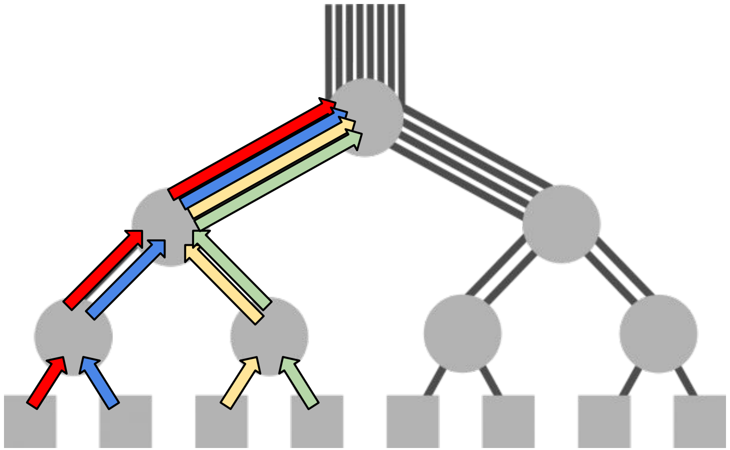
The Logic of Routing

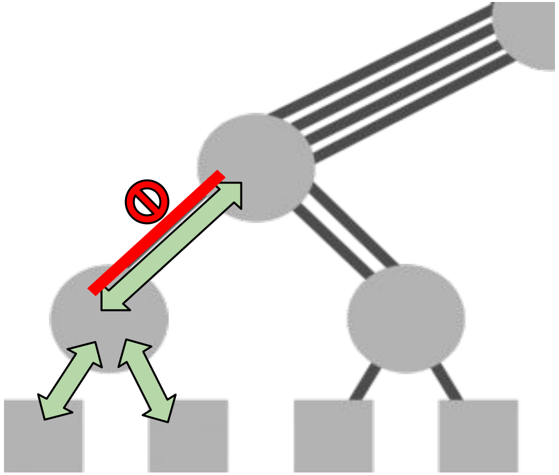
- Routing is based on the binary representation of processor addresses.
- For processors with addresses differing at bit position **i** (starting from left), their data ascends to layer **i** and then descends.



Fault Tolerance

- Even in worst case scenario (i.e. every processor trying to communicate at once) the fat tree can handle it pretty well
- Even with a switch failure, fat-trees can reroute communications to ensure continued functionality, albeit with potential reduced bandwidth.





End