Multiprocessors connected by networks (Section 6.7)

- Each processor has private physical address space

Loosely Coupled Clusters

- Network of independent computers
  - Each has private memory and OS
  - Connected using I/O system (ex: Ethernet or a switch)
- Suitable for applications with independent tasks
  - Web servers, databases, simulations, …
- High availability, scalable, affordable
- Problem: Low interconnect bandwidth (compared to SMP)

- Grid Computing
  - Computers interconnected by long-haul networks (ex: Internet)
  - Work units farmed out, results sent back
  - Can make use of idle time on PCs (ex: PITTGRID)
Programming a distributed address space machine

- Assume that 10000 values are stored in the local memories of 16 processors such that 625 values are stored in x[0] … x[624] in the local memory of each processor.
- All variables are local variables (each processor has its own copy) – no shared variables.
- The function “send(m,p)” sends a message containing the value of m to processor p.
- The function “receive(m)” receives a message and puts the received value in m.

```
sum = 0;
for (i=0 ; i < 625 ; i++)
    sum = sum + x[i];
```

```
half = 8; /* P = 16 */
for (i=0 ; i < 4 ; i++)
    { synchronize; /* a barrier */
      if (Pid < half ) sum = sum + sum[Pid + half];
      half = half / 2; }
```

```
sum = 0;
for (i=0 ; i < 625 ; i++)
    sum = sum + x[i];
```

```
half = 8; /* P = 16 */
for (i=0 ; i < 4 ; i++)
    { if (2*half > Pid >= half ) send(sum, Pid – half);
      if (Pid < half ) { receive(remote_sum);
                      sum += remote_sum; }
      half = half / 2; }
```

- No shared variables.
- Where is the global sum?
- The distribution of the initial data to the local memories is done either by the programmer or by the compiler.

Interconnection network (Section 6.8)

- To connect processors to memories or processors to processors

```
2D Mesh
N-cube (N = 3)
```

```
Bus
Ring
```

- Issues
  - Latency
  - Bandwidth
  - Cost (wires, switches, ports, …)
  - Scalability

- Topology has been a focus of architects
Evaluating Interconnection Network topologies

- **Diameter**: The distance between the farthest two nodes in the network.
- **Average distance**: The average distance between any two nodes in the network.
- **Node degree**: The number of neighbors connected to any particular node.
- **Bisection Width**: The minimum number of wires you must cut to divide the network into two equal parts.
- **Cost**: The number of links or switches (whichever is asymptotically higher) is a meaningful measure of the cost. However, a number of other factors, such as the ability to layout the network, the length of wires, etc., also factor in to the cost.

2-D torus

- **Diameter??**
- **Bisection bandwidth??**
- **Routing algorithms**
  - x-y routing
  - Adaptive routing
- **2D mesh (without the wrap-around connections)**

- **Variants**
  - 1-D (ring), 3-D.
Hypercube interconnections

- An interconnection with low diameter and large bisection width.
- A q-dimensional hypercube is built from two (q-1)-dimensional hypercubes.

Dimension 0

1-dimension binary hypercube  2-dimension binary hypercube  3-dimension binary hypercube

A 4-dimensional Hypercube (16 nodes)

- Can recursively build a q-dimension network – has $2^q$ nodes
Centralized switching: Buses and crossbars

- Cost
- Latency
- Bandwidth
- Scalability

Each switch is a 2x2 switch that can be set to one of 2 settings

Centralized switching: Multistage networks

- NxN Omega network: log N stages, with N/2, 2x2 switches.
- A blocking network: some input-output permutations cannot be realized due to path conflicts.

Circuit switching: circuits are established between inputs and outputs – arbitrate entire circuits.

Packet switching: packets are buffered at intermediate switches – arbitrate individual switches.
Fat tree networks

A fat tree networks using 2x2 bidirectional switches