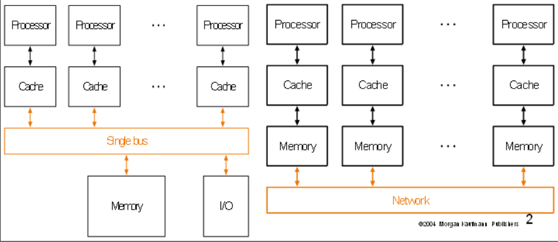
Chapter 9

Multiprocessors

- **Idea: create powerful computers by connecting many smaller ones**

good news: works for timesharing (better than supercomputer)

bad news: its really hard to write good concurrent programs
many commercial failures

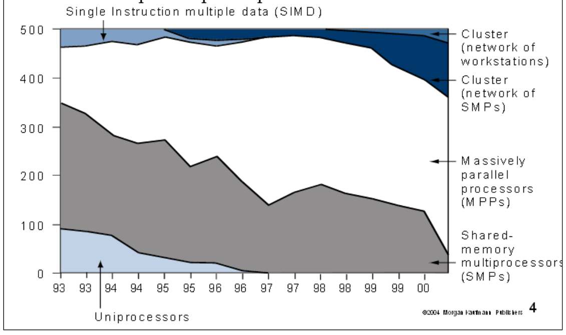


Questions

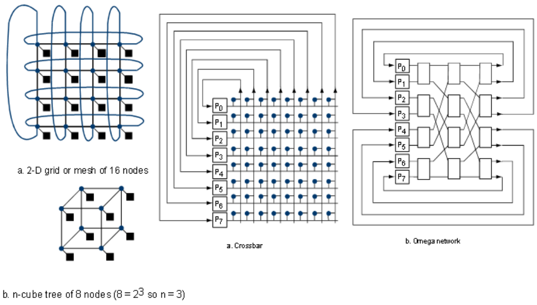
- **How do parallel processors share data?**
  - **— single address space (SMP vs. NUMA)**
  - **— message passing**
- **How do parallel processors coordinate?**
  - **— synchronization (locks, semaphores)**
  - **— built into send / receive primitives**
  - **— operating system protocols**
- **How are they implemented?**
  - **— connected by a single bus**
  - **— connected by a network**

Supercomputers

Plot of top 500 supercomputer sites over a decade:



Topologies



a. 2-D grid or mesh of 16 nodes

a. Crossbar

b. Omega network

b. n-cube tree of 8 nodes (8 = 2^3 so n = 3)

Clusters

- **Constructed from whole computers**
- **Independent, scalable networks**
- **Strengths:**
  - **Many applications amenable to loosely coupled machines**
  - **Exploit local area networks**
  - **Cost effective / Easy to expand**
- **Weaknesses:**
  - **Administration costs not necessarily lower**
  - **Connected using I/O bus**
- **Highly available due to separation of memories**
- **In theory, we should be able to do better**

Multithreading

- Fine-grained multithreading
  - Threads são intercaladas com pequena granularidade, uma thread por ciclo de clock
  - Ganha desempenho nos stalls
  - Divide a carga entre duas threads
- Coarse-grained multithreading
  - Threads são intercaladas com maior granularidade
  - Só troca de thread em stalls mais demorados
- Simultaneous multithreading (SMT)
  - Divide os recursos do processador entre as threads simultaneamente

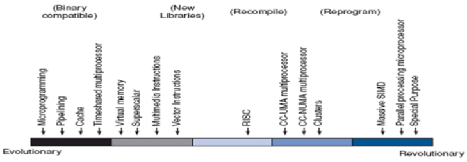
Caracterização de Arquiteturas

- Single instruction stream, singla data stream (SISD, the uniprocessor)
- Single instruction stream, multiple data streams (SIMD)
- Multiple instruction streams, single data stream (MISD)
- Multiple instruction streams, multiple data streams (MIMD)

Concluding Remarks

- Evolution vs. Revolution

"More often the expense of innovation comes from being too disruptive to computer users"



"Acceptance of hardware ideas requires acceptance by software people; therefore hardware people should learn about software. And if software people want good machines, they must learn more about hardware to be able to communicate with and thereby influence hardware engineers."