Multi-Thread Parallelism
Outline

* Some Examples
  - CM
  - HEIP
  - Cosmic Cube
* MP vs. Multicomputer Network
* One Supercomputer vs. "The Multi"
* Amdahl's Law
* Speed-up, Efficiency, Utilization, Redundancy
* Interconnection Networks
* Cache Coherence

* Sequential Consistency
Tightly-coupled vs Loosely-coupled

**Tightly coupled (i.e., Multiprocessor)**
- Shared memory
- Each processor capable of doing work on its own
- Easier for the software
- Hardware has to worry about cache coherency, memory contention

**Loosely-coupled (i.e., Multicomputer Network)**
- Message passing
- Easier for the hardware
- Programmer's job is tougher
Note: A well-meaning student told me to get rid of this slide. cm* is old. People will think you are an old man, and not take you seriously.
The HEP

Function Units

Results

Data

Memory

Operands

GPRs

Full/Empty Bit

Branch Target

+1

Inst. Decode

PC

Inst/PC

Operand Address

Inst/PC

Inst/PC

Inst/PC

Program Memory

PC
Cosmic Cube

(Example: $k = 4$)
One Supercomputer
vs.
"The Multi"

(...Except Even Supercomputers have adopted the multi approach)

\[ 1 \times 2^n \quad 2^k \times 2^{n-k} \quad 2^n \times 1 \]

Why do we care?

-- Economic Answer
-- Strategic Answer
-- Scientific Answer

Scalability

- SIMD easy
- MIMD hard
- Very large Scale
  - Cm 1 - Thinking Machines - \(2^6\) cores
  - non-Von - \(2^{20}\) cores
  - Boolean Vector Machine - \(2^{30}\) cores
Amdahl's Law

* Speed-up as a function of the parallelizability ($\alpha$) of the application

---

**Speed-up vs. Parallelizability for a given number of processors ($p$)**

- $p = 20$
- $p = 10$
- $p = 5$
- $p = 2$

---

* Speed-up of an application as we add more and more processors ($p$)

---

**Speedup vs. Number of Processors ($p$) for a given $\alpha$**

- $\alpha = .98$
- $\alpha = .95$
- $\alpha = .90$
- $\alpha = .50$
- $\alpha = .05$
\[ T_p = \frac{\alpha T_1}{p} + \frac{(1-\alpha) T_1}{1} \]

\[ S_p = \frac{T_1}{T_p} = \frac{\frac{T_1}{\alpha \frac{T_1}{p} + (1-\alpha)}}{p} \]

\[ S_p = \frac{1}{\frac{\alpha}{p} + (1-\alpha)} \]
Example: \( a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0 \)

\[ a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0 \]

\( \rho = 3 \)

\[ S_p (A) = \frac{T_1 (A)}{T_p (A)} = \frac{11}{5} = 2.2 \]

Right? [WRONG] Why?
Note: \[ q_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0 = x(a_4 x^3 + a_3 x^2 + a_2 x + a_1) + a_0 \]

\[ x(a_4 x^2 + a_3 x) + a_2 \]

\[ x(a_4 x + a_3) \]

\[ T_1 = 8 \]

\[ S_p(A) = \frac{T_1(A)}{T_p(A)} = \frac{8}{5} = 1.6 \]
Efficiency = \frac{1 \cdot T_i}{\rho \cdot T_p} = \frac{8}{3 \times 5} = \frac{8}{15} \\

Utilization = \frac{O_p}{\rho \cdot T_p} = \frac{11}{3 \times 5} = \frac{11}{15} \\

Redundancy = \frac{O_p}{O_1} = \frac{11}{8} \\

E \cdot R = U
<table>
<thead>
<tr>
<th>Type</th>
<th>Cost</th>
<th>Latency</th>
<th>Contention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bus</td>
<td>$O(n)$</td>
<td>1</td>
<td>Worst</td>
</tr>
<tr>
<td>Full Crossbar</td>
<td>$O(n^2)$</td>
<td>1</td>
<td>Best</td>
</tr>
<tr>
<td>Omega Network</td>
<td>$nk \log_k n$</td>
<td>$O(\log_k n)$</td>
<td></td>
</tr>
<tr>
<td>Tree</td>
<td>$O(n)$</td>
<td>$O(\log_2 n)$</td>
<td></td>
</tr>
<tr>
<td>Hypercube</td>
<td>$O(n \log n)$</td>
<td>$O(\log_2 n)$</td>
<td></td>
</tr>
<tr>
<td>Ring</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
<td></td>
</tr>
<tr>
<td>Mesh</td>
<td>$O(n)$</td>
<td>$O(\sqrt{n})$</td>
<td></td>
</tr>
</tbody>
</table>
**More Detail, Omega Network**

**Omega Network (Duncan Laurie, UIUC)**

- \( n \)
- \( k \)

\[ \text{Col.} (1) \]
\[ \text{Col.} (\log_2 n) \]

**Example:** \( n = 8, \ k = 2 \)

**Banyan Tree (G. Jack Lipovski, UT)**

- \( l \) = \# of levels of switches
- \( p \) = \# of connections on processor side
- \( m \) = \# of connections on memory side
Omega Networks (Bonus Example)
BANYAN TREE (CONTINUED)

A SWITCH:

Example: \( l = 2 \), \( p = 2 \), \( m = 3 \)

Example: \( l = 3 \), \( p = 2 \), \( m = 3 \)
Example: \( L = 4, p = 2, m = 3 \)

Note: \( p^L \) processes, \( m^L \) memories

Level 1 switches: \( p^{L-1} \)

1. \( 2 \) \( u \) : \( m \cdot p \)
2. \( 3 \) \( u \) : \( m^2 \cdot p \)
3. \( \vdots \)
4. \( L \) \( u \) : \( m^{L-1} \)

Note: For both Omega and Banyan:

A unique path from each to each.
Cache Coherency

**Directory**

![Diagram of cache coherence with directories and switches]

Memory directory consists of \( n + 1 \) bits/cache line.

* If cache has it in **read** mode, bit = 1
* If cache has it in **write** mode, exclusive bit = 1

In private cache, if private bit = 1, write mode.
* Private caches are not write through.

Examples for \( n = 3 \):

- \[1010\], \( C_1, C_3 \) have it in **read** mode.
- \[10101\], \( C_2 \) has it in **write** mode.
Snoopy Cache

```
<table>
<thead>
<tr>
<th>P_1</th>
<th>P_2</th>
<th>...</th>
<th>P_n</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_1</td>
<td>C_2</td>
<td></td>
<td>C_n</td>
</tr>
</tbody>
</table>
```

Mem

```
INV  VALID  RSVD
```

Not: Cache line = One Element

MESI/MOESI
Cache Coherency
(Snoopy Cache)

Update
Invalidation

P
PP, PW
C
PR, PW

BUS

Memory

INV
BR, BW

DIRTY
BR, BW

SHARED
BW, BR

RSVD
PW, PR

WRITE
ONCE

CACHE LINE = DATA ELEMENT
SYNAPSE
S. FRANCI

MOESI

write through
 DON'T WRITE THROUGH

write though

write through

GOODMAN
ISCA 1983

KAMEL PATHI
IILLINOIS
ISCA 1984
1 Nov 4 Sep 1984
BACA
PATH DESIGN
WRITE

PR, PW

PR, PW
Sequential Consistency

Why: To protect against two threads simultaneously accessing a critical section

Thread 1
- $L1 = \emptyset$
- 
- $A: L1 \leftarrow 1$
- $B: \text{if}(L2 = \emptyset)\left\{\text{critical section}\right\}$
- $C: L1 \leftarrow \emptyset$

Thread 2
- $L2 = \emptyset$
- 
- $X: L2 \leftarrow 1$
- $Y: \text{if}(L1 = \emptyset)\left\{\text{critical section}\right\}$
- $Z: L2 \leftarrow \emptyset$

What causes the problem:
- $B \rightarrow X$ and $Y \rightarrow A$

Sequential Consistency

LD/ST Access Memory in Program Order.

i.e. $A \rightarrow B \rightarrow C$ and $X \rightarrow Y \rightarrow Z$
In order for critical section to be accessible by both and sequential consistency to be maintained

\[ A \xrightarrow{x} X \xrightarrow{Y} B \xrightarrow{x} Y \xrightarrow{A} \]

\[ \text{IMPOSSIBLE} \]

\[ \therefore \text{Sequential consistency } \rightarrow \text{mutually exclusive access to critical section} \]

Accesses (legitimate) for sequential consistency

1. A
2. B
3. C
4. X
5. Y
6. Z

\[ 1 \xrightarrow{1} 2 \]

\[ 2 \xrightarrow{2} \]

\[ \text{Now} \]

\[ 1 \xrightarrow{1} \]