Key Issues

Run PhotoShop on 1 PC or N PCs

Programmability
- How to program a bunch of PCs viewed as a single logical machine.

Performance Scalability - Speedup
- Run PhotoShop on 1 PC (forget the specs of this PC)
- Run PhotoShop on N PCs

- Will it run faster on N PCs? Speedup = ?
Types of Multiprocessors

Key: Data and Instruction

**Single Instruction Single Data (SISD)**
- Intel processors, AMD processors

**Single Instruction Multiple Data (SIMD)**
- Array processor
- Pentium MMX feature

**Multiple Instruction Single Data (MISD)**
- Systolic array
- Special purpose machines

**Multiple Instruction Multiple Data (MIMD)**
- Typical multiprocessors (Sun, SGI, Cray,...)

**Single Program Multiple Data (SPMD)**
- Programming model

---

Shared-Memory Multiprocessor

Processor

Interconnection network

Main Memory

Storage

I/O

---

Lecture 10: Intro to Multiprocessors/Clustering

12/7/2003 A. Sohn
Distributed-Memory Multiprocessor

Key Issues

**Programmability**
- Intel Pentium, AMD machines

**Performance (Scalability) - Speedup**
- Run PhotoShop on 1 machine, or n machines
- Execution time on 1 machine = \( x \)
- Execution time on n machines = \( y \)
- Speedup = \( x/y \) = the best speedup = n (often higher due to various reasons including cache behavior)
Parallelization of Applications

Parallel portion 80%

Serial portion 20%

Lecture 10: Intro to Multiprocessors/Clustering

Parallelization for 4 Processors

Consider a loop with 100 iterations:

for (i=0; i<100; i++) a[i] = b[i] + c[i];

The loop has no loop-carried dependence, straight forward to parallelize the code for 4 processors. Each processor executes 1/4 of the loop, i.e. 25 iterations as follows:

P0: for (i=0; i<25; i++) a[i] = b[i] + c[i];
P1: for (i=25; i<50; i++) a[i] = b[i] + c[i];
P2: for (i=50; i<75; i++) a[i] = b[i] + c[i];
P3: for (i=75; i<100; i++) a[i] = b[i] + c[i];
Shared-Memory Multiprocessor

Processor

Processor

Processor

Interconnection network

for (i=0;i<100;i++)
a[i] = b[i] + c[i];

Storage

I/O

Distributed-Memory Multiprocessor

Processor

Processor

Processor

Interconnection network

IO/S

for (i=0;i<25;i++)
a[i] = b[i] + c[i];

for (i=25;i<50;i++)
a[i] = b[i] + c[i];

for (i=50;i<75;i++)
a[i] = b[i] + c[i];

for (i=75;i<100;i++)
a[i] = b[i] + c[i];

IO/S
Amdahl’s Law

Maximum speedup is limited by the serial fraction of a program

- \( N \) = the number of processors,
- \( s \) = the time spent by a processor on serial part of a program,
- \( p \) = the time spent by a processor on parallel part of a program
- \( t = \text{total time} = s + p = 1 \)

Maximum possible speedup is given by:

\[
D = \frac{s + p}{s + \frac{p}{N}} = \frac{1}{1 - \frac{p}{p}} = \frac{1}{1 - \frac{1}{D}}
\]

\[
p = \frac{1 - \frac{1}{D}}{1 - \frac{1}{N}}
\]

Amdahl’s Law - Example

\( N = 10 \) PCs, the speedup to achieve \( D = 8 \), the portion of program to be parallelized will be

\[
p = \frac{1 - \frac{1}{8}}{1 - \frac{1}{10}} = \frac{0.875}{0.9} = 0.972
\]

What do these numbers mean? want to achieve 8 fold speedup on 10 PCs --> 97.2% of the entire code has to be parallelized!
Amdahl’s Law - Example

```c
main()
{
    .......... 
    .......... 
    .......... 
    .......... 
    .......... 
    .......... 
    .......... 
    .......... 
    .......... 
    .......... 
    .......... 
    .......... 
    .......... 
    .......... 
    .......... 
    .......... 
    .......... 
    .......... 
    .......... 
    .......... 
}
```

Clustering of PCs

- A Poor Man’s Supercomputer
- Linux box clustering
- MS Windows clustering
- For scalability and high availability

- Mail server clustering: MS Exchange
- DB server clustering - MS SQL server, MySQL server, ...
- VPN box clustering
- Firewall clustering
- Filtering clustering
- Web clustering
Data Center Infrastructure
Chapter 6 of our textbook