**Multicore:** Parallelization, Amdhal's Law, Gustafson's Law, Effectiveness, Design Space of Shared Memory Architecture

Dr. A. Sahu  
CSE, IIT Guwahati

---

**Parallel Architectures**

**Flynn’s Classification**

Architecture Categories

- SISD
- SIMD
- MISD
- MIMD

---

**Parallel Architectures**

**Sima’s Classification**

Parallel architectures PAs

- Data-parallel architectures
- Function-parallel architectures

---

**Parallel Decomposition**

- Data Parallelism
- Function Parallelism
- Pipeline Parallelism
- Mixed Parallelism (D+F+P)

---

**Function Parallel Architectures**

Function-parallel architectures

- Instruction level PAs
- Thread level PAs
- Process level PAs

Built using general purpose processors

MIMDs

Shared Memory MIMD

Distributed Memory MIMD
Issues from user’s perspective

• Specification / Program design
  – explicit parallelism or
  – implicit parallelism + parallelizing compiler
• Partitioning / mapping to processors
• Scheduling / mapping to time instants
  – static or dynamic
• Communication and Synchronization

Parallelizing example

```plaintext
m=0
for (i=0; i<n; i++) {
  m = m+3
  a[i] = (a[m]+a[m+1]+a[m+2])/3
}
```

Can all iterations be done in parallel?

Dependence 1: m = m + 3
Dependence 2:

Parallelizing example - contd.

Eliminate dependence based on induction variable
```plaintext
for (i=0; i<n; i++) {
  m = i*3
  a[i] = (a[m]+a[m+1]+a[m+2])/3
}
```

Parallelizing example - contd.

Eliminate forward dependency using double buffer
```plaintext
for (i=0; i<n; i++) {
  m = i*3
  aa[i] = (a[m]+a[m+1]+a[m+2])/3
}
barrier()
for (i=0; i<n; i++) {
  a[i] = aa[i]
}
```

Grain size and performance

![Graph showing speed up and efficiency](image)

Speed up and efficiency

Let:
- \( T_p \) = Time on uniprocessor
- \( T_p \) = Time on \( p \) processors

Speed up = \( S_p = \frac{T_i}{T_p} \)
Efficiency = \( E_p = \frac{T_i}{p \cdot T_p} \)

Usually, \( S_p < p \) or \( E_p < 1 \) due to overheads

Sometimes superlinear speedup (\( S_p > p \) or \( E_p > 1 \)) is reported

This may be due to:
  • failure to use the best uniprocessor algorithm
  • advantage due to larger memory
Amdahl’s Law

Serial fraction \( s = \frac{T_s}{T}
\)

\( T_p = T_s \left( 1 + \frac{T_r}{T} \right) \)

\( S_p = T_s \frac{T_i}{T} \)

Generalization: Equal work Hypothesis

- Suppose \( p \) processor system
- Equal amount of work for 1 Processor is active

\( T_{1,p} = T_s \)

- Control flow limit the amount of work available so that it equally distribute among \( p \) processor that is \( i \). \( T_{1,i,p} = 2T_{2,p} = 3T_{3,p} \)
- Equal amount of work for 2 processor is active

\( T_{2,p} = \frac{1}{2} T_s \)

- So on...

\( T_{i,p} = \frac{1}{i} T_s \)
Effectiveness of Parallel Processing

- Relationship between these parameters
  \[1 \leq S(p) \leq p\]
  \[U(p) = R(p)E(p)\]
  \[E(p) = S(p)/p\]
  \[Q(p) = E(p)S(p)/R(p)\]
  \[1/p \leq E(p) \leq U(p) \leq 1\]
  \[1 \leq R(p) \leq 1/E(p) \leq p\]
  \[Q(p) \leq S(p) \leq p\]

Assume Addition is vertically aligned
- Vertical Arrow don’t Require any communication
- Oblique Arrow require unit communication

T(1) = 15, Assume p = 8
W(8) = 22, T(8) = 7, E(8) = 15/(8x7) = 27%
S(8) = 15/7 = 2.14, R(8) = 22/15 = 1.47, Q(8) = 0.39

Effectiveness : Example

- Design Space of Shared Memory Architectures
  - Extent of address space sharing
  - Location of memory modules
  - Uniformity of memory access
Address Space

Each processor sees an exclusive address space
Each processor sees partly exclusive and partly shared address space
Each processor sees same shared address space

Location of Memory

Centralized
Distributed
Mixed

Clustered Architecture

Global Interconnection Network

Uniformity of Access

• UMA (Uniform Memory Access)
  – Uniformity across memory address space
  – Uniformity across processors
• NUMA (Non-Uniform Memory Access)
• CC-NUMA (Cache Coherent NUMA)
• COMA (Cache Only Memory Architecture)

UMA:
Symmetrical Shared Memory Multiprocessor (SMP)
NUMA:
Distributed Shared Memory Multiprocessor

Location and Sharing

full SHARING partial none

centralized UMA
mixed NUMA
distributed

A Sahu