Parallel Processing & Distributed Systems

Lectured by: Phạm Trần Vũ
Prepared by: Thoại Nam
Course Detail

- Two lectures per week (90 minutes each)
  - Tuesday: 10:00 – 11:35
  - Thursday: 8:15 – 9:50

- References
  - *MPI*: [http://www.mpi-forum.org/docs/docs.html](http://www.mpi-forum.org/docs/docs.html)
Chapter 1: Introduction

- Introduction
  - What is parallel processing?
  - Why do we use parallel processing?

- Applications

- Parallelism
Sequential Processing

- 1 CPU
- Simple
- Big problems???
Application Demands
Grand Challenge Problems

- A grand challenge problem is one that cannot be solved in a reasonable amount of time with today’s computers.

- Ex:
  - Modeling large DNA structures
  - Global weather forecasting
  - Modeling motion of astronomical bodies
Solutions

- Power processor
  - 50 Hz -> 100 Hz -> 1 GHz -> 4 Ghz -> ... -> Upper bound?

- Smart worker
  - Better algorithms

- Parallel processing
N-body

- The $N^2$ algorithm:
  - N bodies
  - N-1 forces to calculate for each body
  - $N^2$ calculations in total
  - After the new positions of the bodies are determined, the calculations must be repeated

- A galaxy:
  - $10^7$ stars and so $10^{14}$ calculations have to be repeated
  - Each calculation could be done in 1µs ($10^{-6}$s)
  - It would take 10 years for one iteration
  - But it only takes 1 day for one iteration with 3650 processors
Parallel Processing Terminology

- Parallel processing
- Parallel computer
  - Multi-processor computer capable of parallel processing
- Throughput:
  - The throughput of a device is the number of results it produces per unit time.
- Speedup
  \[ S = \frac{\text{Time(sequential algorithm)}}{\text{Time(parallel algorithm)}} \]
- Parallelism:
  - Pipeline
  - Data parallelism
  - Control parallelism
- A number of steps called **segments** or **stages**
- The output of one segment is the input of other segment
Data Parallelism

- Applying the same operation simultaneously to elements of a data set
Pipeline & Data Parallelism

1. Sequential execution

2. Pipeline

3. Data Parallelism
Pipeline & Data Parallelism

- Pipeline is a special case of control parallelism
- T(s): Sequential execution time
  - T(p): Pipeline execution time (with 3 stages)
  - T(dp): Data-parallelism execution time (with 3 processors)
- S(p): Speedup of pipeline
- S(dp): Speedup of data parallelism

<table>
<thead>
<tr>
<th>widget</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>T(s)</td>
<td>3</td>
<td>6</td>
<td>9</td>
<td>12</td>
<td>15</td>
<td>18</td>
<td>21</td>
<td>24</td>
<td>27</td>
<td>30</td>
</tr>
<tr>
<td>T(p)</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>T(dp)</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>S(p)</td>
<td>1</td>
<td>1+1/2</td>
<td>1+4/5</td>
<td>2</td>
<td>2+1/7</td>
<td>2+1/4</td>
<td>2+1/3</td>
<td>2+2/5</td>
<td>2+5/11</td>
<td>2+1/2</td>
</tr>
<tr>
<td>S(dp)</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2+1/2</td>
<td>3</td>
<td>2+1/3</td>
<td>2+2/3</td>
<td>3</td>
<td>2+1/2</td>
</tr>
</tbody>
</table>
Pipeline & Data Parallelism
Control Parallelism

- Applying different operations to different data elements simultaneously
Scalability

- An algorithm is scalable if the level of parallelism increases at least linearly with the problem size.
- An architecture is scalable if it continues to yield the same performance per processor, albeit used in large problem size, as the number of processors increases.

- Data-parallelism algorithms are more scalable than control-parallelism algorithms.