Chapter 2
Parallel Computer Models & Classification

Thoai Nam
Chapter 2: Parallel Computer Models & Classification

- Abstract Machine Models:
  - PRAM, BSP, Phase Parallel
- Pipeline, Processor Array, Multiprocessor, Data Flow Computer
- Flynn Classification:
  - SISD, SIMD, MISD, MIMD
- Pipeline Computer
Abstract Machine Models

Thoai Nam
Abstract Machine Models

- An abstract machine model is mainly used in the design and analysis of parallel algorithms without worry about the details of physics machines.

- Three abstract machine models:
  - PRAM
  - BSP
  - Phase Parallel
RAM (random access machine)

- Read-only input tape
- Location counter
- Program
- Write-only output tape
- Memory
Parallel random-access machine

- Control
- Interconnection network
- Global memory
- Private memory
- Parallel random-access machine

- $P_1$
- $P_2$
- $P_n$
A control unit
An unbounded set of processors, each with its own private memory and an unique index
Input stored in global memory or a single active processing element
Step: (1) read a value from a single private/global memory location
   (2) perform a RAM operation
   (3) write into a single private/global memory location
During a computation step: a processor may activate another processor
All active, enable processors must execute the same instruction (albeit on different memory location)
Computation terminates when the last processor halts
Definition:
The cost of a PRAM computation is the product of the parallel time complexity and the number of processors used.

Ex: a PRAM algorithm that has time complexity $O(\log p)$ using $p$ processors has cost $O(p \log p)$
Time Complexity Problem

- Time complexity of a PRAM algorithm is often expressed in the big-O notation.
- Machine size $n$ is usually small in existing parallel computers.
- Ex:
  - Three PRAM algorithms A, B and C have time complexities if $7n$, $(n \log n)/4$, $n \log \log n$.
  - Big-O notation: $A(O(n)) < C(O(n \log \log n)) < B(O(n \log n))$
  - Machines with no more than 1024 processors:
    $\log n \leq \log 1024 = 10$ and $\log \log n \leq \log \log 1024 < 4$
    and thus: $B < C < A$
Conflicts Resolution Schemes (1)

- PRAM execution can result in simultaneous access to the same location in shared memory.
  - Exclusive Read (ER)
    » No two processors can simultaneously read the same memory location.
  - Exclusive Write (EW)
    » No two processors can simultaneously write to the same memory location.
  - Concurrent Read (CR)
    » Processors can simultaneously read the same memory location.
  - Concurrent Write (CW)
    » Processors can simultaneously write to the same memory location, using some conflict resolution scheme.
Conflicts Resolution Schemes(2)

- **Common/Identical CRCW**
  - All processors writing to the same memory location must be writing the same value.
  - The software must ensure that different values are not attempted to be written.

- **Arbitrary CRCW**
  - Different values may be written to the same memory location, and an arbitrary one succeeds.

- **Priority CRCW**
  - An index is associated with the processors and when more than one processor write occurs, the lowest-numbered processor succeeds.
  - The hardware must resolve any conflicts.
Begin with a single active processor active

Two phases:
- A sufficient number of processors are activated
- These activated processors perform the computation in parallel

$\lceil \log p \rceil$ activation steps: $p$ processors to become active

The number of active processors can be double by executing a single instruction
Parallel Reduction (1)

\[
\begin{array}{cccccccc}
4 & 3 & 8 & 2 & 9 & 1 & 0 & 5 \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\
7 & 10 & 10 & 5 & 9 \\
\downarrow & \downarrow & \downarrow & \downarrow & \\
17 & 15 & 9 \\
\downarrow & \downarrow & \\
32 & 9 \\
\downarrow & \\
41 & \\
\end{array}
\]
Parallel Reduction (2)

(EREW PRAM Algorithm in Figure 2-7, page 32, book [1])

Ex: \( \text{SUM}(\text{EREW}) \)

Initial condition: List of \( n \geq 1 \) elements stored in \( A[0..(n-1)] \)

Final condition: Sum of elements stored in \( A[0] \)

Global variables: \( n, A[0..(n-1)], j \)

begin

\begin{align*}
\text{spawn} (P_0, P_1, \ldots, P_{\left\lfloor n/2 \right\rfloor -1}) \\
\text{for all } P_i \text{ where } 0 \leq i \leq \left\lfloor n/2 \right\rfloor -1 \text{ do} \\
\quad \text{for } j \leftarrow 0 \text{ to } \left\lceil \log n \right\rceil - 1 \text{ do} \\
\quad \quad \text{if } i \text{ modulo } 2i = 0 \text{ and } 2i+2i < n \text{ the} \\
\quad \quad \text{endif} \\
\quad \text{endfor} \\
\text{endfor} \\
\text{end}
\end{align*}

end
BSP – Bulk Synchronous Parallel

- BSP Model
  - Proposed by Leslie Valiant of Harvard University
  - Developed by W.F. McColl of Oxford University
BSP Model

- A set of n nodes (processor/memory pairs)
- Communication Network
  - Point-to-point, message passing (or shared variable)
- Barrier synchronizing facility
  - All or subset
- Distributed memory architecture
BSP Programs

- A BSP program:
  - \( n \) processes, each residing on a node
  - Executing a strict sequence of supersteps
  - In each superstep, a process executes:
    » Computation operations: \( w \) cycles
    » Communication: \( gh \) cycles
    » Barrier synchronization: \( l \) cycles
Three Parameters

- The basic time unit is a cycle (or time step)
  - \( w \) parameter
    - Maximum computation time within each superstep
    - Computation operation takes at most \( w \) cycles.
  - \( g \) parameter
    - Number of cycles for communication of unit message when all processors are involved in communication - network bandwidth
    - \((\text{total number of local operations performed by all processors in one second}) / (\text{total number of words delivered by the communication network in one second})\)
    - \( h \) relation coefficient
    - Communication operation takes \( gh \) cycles.
  - \( l \) parameter
    - Barrier synchronization takes \( l \) cycles.
A Figure of BSP Programs

Superstep 1
Computation
Communication
Barrier
Superstep 2
Computation
Communication
Barrier
Time Complexity of BSP Algorithms

- Execution time of a superstep:
  - Sequence of the computation, the communication, and the synchronization operations: \( w + gh + l \)
  - Overlapping the computation, the communication, and the synchronization operations: \( \max\{w, gh, l\} \)
Phase Parallel

- Proposed by Kai Hwang & Zhiwei Xu
- Similar to the BSP:
  - A parallel program: sequence of phases
  - Next phase cannot begin until all operations in the current phase have finished
  - Three types of phases:
    » **Parallelism phase**: the overhead work involved in process management, such as process creation and grouping for parallel processing
    » **Computation phase**: local computation (data are available)
    » **Interaction phase**: communication, synchronization or aggregation (e.g., reduction and scan)
- Different computation phases may execute different workloads at different speed.
Parallel Computer Models (1)

- A parallel machine model (also known as *programming model*, *type architecture*, *conceptual model*, or *idealized model*) is an abstract parallel computer from programmer's viewpoint, analogous to the von Neumann model for sequential computing.

- The abstraction need not imply any structural information, such as the number of processors and interprocessor communication structure, but it should capture implicitly the relative costs of parallel computation.

- Every parallel computer has a native model that closely reflects its own architecture.
Parallel Computer Models (2)

- Five semantic attributes
  - Homogeneity
  - Synchrony
  - Interaction mechanism
  - Address space
  - Memory model

- Several performance attributes
  - Machine size
  - Clock rate
  - Workload
  - Speedup, efficiency, utilization
  - Startup time
  ...