HPC Chapter 1: Introduction to Parallel Computing

Classification Models: Architectural Schemes (Flynn ‘s, Feng ‘s, Handler ‘s)

Flynn ‘s Classification Scheme is based on the notion of a stream of information. Two types of information flow into a processor: instructions and data.

The instruction stream is defined as the sequence of instructions performed by the processing unit.

The data stream is defined as the data traffic exchanged between the memory and the processing unit.

Types of Flynn ‘s Taxonomy:

1. SISD single-instruction single data stream
2. SIMD single-instruction multiple data stream
3. MISD multiple-instructions single data stream
4. MIMD multiple instructions multiple data stream

SISD (Single Instruction Single Data): Traditional single processor von Neumann computers are classified as SISD.

SIMD (Single Instruction Multiple Data): Multiple processors execute same instruction on different data in parallel.

MISD (Multiple Instruction Single Data): Multiple processors execute different instructions on the same data (not commonly used).

MIMD (Multiple Instruction Single Data): Multiple processors execute different instructions on the different data.

Further classified into:

- Shared Memory (SMP)
- Message Passing (Distributed Memory)

Feng ‘s Classification suggested the use of degree of parallelism to classify various computer architecture.

The maximum number of binary digits that can be processed within a unit time by a computer system is called the maximum parallelism degree P.

A bit slice is a string of bits one from each word at the same vertical position.

Types of Feng ‘s classification:

1. Word Serial Bit Serial (WSBS)
2. Word Parallel Bit Serial (WPBS)
3. Word Serial Bit Parallel (WSBP)
4. Word Parallel Bit Parallel (WPBP)

Word Serial Bit Serial (WSBS): One bit is processed at a time.

Word Parallel Bit Serial (WPBS): Processes m-bit slices at a time.

Word Serial & Bit Parallel (WSBP): One word at a time with bit parallelism.

Word Parallel & Bit Parallel (WPBP): Fully parallel processing.

Handler 's Classification has proposed a classification scheme for identifying the parallelism degree and pipelining degree built into the hardware structure of the computer system. He considers at three subsystem levels:

1. Processor Control Unit (PCU)
2. Arithmetic Logic Unit (ALU)
3. Bit Level Circuit (BLC)

Processor Control Unit (PCU): Represents CPU-level parallelism.

Arithmetic Logic Unit (ALU): Represents computational units.

Bit Level Circuit (BLC): Represents bit-wise operations.

<u>Levels of Parallelism:</u>

Parallelism in computing can be classified into different levels based on how the operations are carried out. Here are four main levels:

1. Instruction-Level Parallelism (ILP):
    a. Involves executing multiple instructions from a program simultaneously.
    b. Achieved through pipelining, superscalar execution and out-of-order execution.
    c. Example:
        i.
```
for (i=1; i<=100; i++)
    y[i] = y[i] + x[i];
```
        ii. Each iteration of this loop can be executed in parallel.
2. Thread-level or Task-level Parallelism (TLP):
    a. Involves executing different tasks or threads concurrently on multiple processors or cores.
    b. Example:
        i. One thread performs matrix multiplication while another thread sorts an array.
3. Data-Level Parallelism (DLP):
    a. Same operation is performed on multiple data points simultaneously.
    b. Example:
        i. A dual core system can sum two halves of an array in parallel.
4. Bit-Level Parallelism (BLP):
    a. Increasing the word-size of a processor to process more bits at once.

b. Example:
    i. An 8-bit processor adds two 16-bit numbers in two steps, whereas a 16-bit processor does it in one step.

What is Parallel Processing?

Processing of multiple tasks simultaneously on multiple processors is called parallel computing.

Memory Access Architecture

1. Shared Memory Architecture (UMA & NUMA):
    a. Uniform Memory Access (UMA): All processors share a common global memory with equal access time.
    b. Non-Uniform Memory Access (NUMA): All processors have local memory and access time varies depending upon the location.
2. Distributed Memory Architecture:
    a. Each processor has its own private memory.
    b. Processors communicate via message passing. (MPI)
3. Hybrid Distributed Share Memory:
    a. Combines shared and distributed memory approaches.
    b. Some part of memory is shared while other parts are local to the processors.

Parallel Architectures

1. Pipeline Architecture

Concept:

- A single instruction is broken into multiple stages, with each stage executing in parallel.
- Works similarly to an assembly line in a factory.

Key Characteristics:

- Improves instruction throughput.
- Each stage processes part of the instruction in parallel with others.
- Efficient for repetitive tasks like instruction execution.

Example:

- **Instruction Pipeline in CPUs:**
    1. **Fetch**: Retrieve instruction from memory.
    2. **Decode**: Identify operation and operands.
    3. **Execute**: Perform calculations.
    4. **Memory Access**: Read/write data.
    5. **Write Back**: Store the result in registers.

Types of Pipelining:

1. **Arithmetic Pipeline** – Used in floating-point operations.
2. **Instruction Pipeline** – Used in RISC processors.

2. Array Processor

Concept:

- A type of SIMD (Single Instruction Multiple Data) architecture.
- Multiple processing elements (PEs) perform the same instruction on different data simultaneously.

Key Characteristics:

- Suitable for vector processing applications.
- Improves performance in matrix operations and image processing.
- Uses a central control unit to manage operations.

Example:

- **Graphics Processing Units (GPUs)** use array processing for parallel execution of pixel operations.

3. Multiprocessor Architecture

Concept:

- Multiple CPUs (processors) work together to solve a problem faster.
- Can be **tightly coupled (shared memory)** or **loosely coupled (distributed memory)**.

Types of Multiprocessor Architectures:

1. **Symmetric Multiprocessing (SMP)**
    - All processors share the same memory and operate under a single OS.
    - Example: Modern multi-core processors.
2. **Asymmetric Multiprocessing (AMP)**
    - One processor controls others, which may have specific tasks.
    - Example: Older supercomputers.
3. **Massively Parallel Processing (MPP)**
    - Hundreds or thousands of processors working on separate tasks.
    - Example: Supercomputers used for climate modeling.