

## HPC

**High Performance Computing (HPC)** refers to the use of **supercomputers and parallel processing techniques** to solve complex computational problems quickly and efficiently.

### **Parallel Computing:**

parallel computing involves dividing a large computational problem into smaller, independent tasks that can be executed simultaneously on multiple processors or computers

- Speeding up processing time
- Handling large-scale problems
- Efficient use of multicore CPUs and GPUs

### **1. BFS & DFS:**

#### **Breadth-First Search**

Breadth-First Search (or BFS) is an algorithm for searching a tree or an undirected graph data structure. Here, we start with a node and then visit all the adjacent nodes in the same level and then move to the adjacent successor node in the next level. This is also known as level-by-level search.

Parallelized using queues and assigning **multiple threads** to explore neighbors of multiple nodes simultaneously.

**Queue:** FIFO (First In, First Out)

Enqueue: Add an element at the rear.

Dequeue: Remove an element from the front.

#### **Depth-First Search:**

DFS is an algorithm for searching a tree or an undirected graph data structure. Here, the concept is to start from root node and traverse as far as possible in the same branch.

Harder to parallelize due to recursion/stack dependencies but can be done by spawning tasks for independent subtrees.

**Stack:** LIFO (Last In, First Out)

Push: Add an element at the top.

Pop: Remove an element from the top.

## 2. Parallel Bubble Sort and Merge Sort

- **Parallel Bubble Sort:**  
Uses multiple threads to compare and swap elements in adjacent pairs simultaneously (often inefficient).
- **Parallel Merge Sort:**  
Divide the array recursively, sort parts in parallel, then merge. **Highly parallelizable and efficient.**

## 3. Parallel reduction:

refers to algorithms which combine an array of elements producing a single value as a result. by leveraging multiple processors or threads simultaneously. Problems eligible for this algorithm include Sum of an array. Minimum/Maximum of an array.

**4. CUDA** (Compute Unified Device Architecture) is a parallel computing platform and API/programming model developed by NVIDIA, to use GPUs for general-purpose processing.

**CUDA C:** An extension of C/C++ that enables writing programs to run on **NVIDIA GPUs** for parallel computation.

## DL

### ◆ What is a Neural Network?

A **Neural Network (NN)** is a computational model inspired by the human brain. A set of **interconnected layers of neurons** that learn patterns from data using weights, biases, and activation functions.

**Types:** Feedforward, CNN, RNN, etc.

- **Structure:**
  - **Input Layer:** Receives data
  - **Hidden Layers:** Perform computations
  - **Output Layer:** Produces final result

### ◆ What is a Deep Neural Network (DNN)?

A **Deep Neural Network** is a neural network with **multiple hidden layers** between the input and output layers.

- **Advantage:** Can learn more complex features and patterns.

### ◆ What is Linear Regression?

Linear regression is a **supervised learning algorithm** used to predict a **continuous value** based on one or more input features.

Types: Simple and Multiple Linear Regression:

#### ✓ Applications:

- Predicting **house prices**
- **Sales forecasting**
- **Medical risk estimation**
- **Stock market trend analysis**

### ◆ What is Gradient Descent?

Gradient Descent is an **optimization algorithm** used to **minimize the loss** in machine learning models by updating weights.

- It moves the weights in the direction **opposite to the gradient** of the loss function.

**Types:** Batch Gradient Descent and Stochastic Gradient Descent (SGD)

**OCR (Optical Character Recognition) Dataset** refers to a collection of **labeled images of text or characters**, used to train machine learning models to recognize and convert **printed or handwritten text into digital form**.

### ◆ CNN (Convolutional Neural Network)

**(CNN)** is a type of deep neural network designed primarily for **image and visual data processing**.

1. **Convolutional Layer:** Applies filters/kernels to input image to detect **features** like edges, corners, textures. Result: feature map.
  2. **ReLU Activation:** Adds non-linearity, helps model complex patterns.
  3. **Pooling Layer:** Reduces dimensionality and computational cost. Keeps important features.
  4. **Fully Connected Layer:** Used for final classification or regression.
-

#### ◆ RNN (Recurrent Neural Network)

A neural network designed for **sequence data** (e.g., time series, text), where **output of previous steps** is used as input for the next.

---

#### ◆ LSTM (Long Short-Term Memory)

An advanced version of RNN that solves the **vanishing gradient problem** using **gates** to control memory flow. **Application:** Text prediction, speech recognition.

#### ◆ 1. SISD – Single Instruction, Single Data

A **single processor** executes a **single instruction** on **one data stream** at a time.

#### ◆ 2. SIMD – Single Instruction, Multiple Data

**One instruction** operates on **multiple data points simultaneously**.

#### ◆ 3. MISD – Multiple Instruction, Single Data

**Multiple processors** execute **different instructions** on the **same data**.

#### ◆ 4. MIMD – Multiple Instruction, Multiple Data

**Multiple processors** execute **different instructions** on **different data** independently.

#### IMDB (Internet Movie Database):

The IMDB dataset is a popular **benchmark dataset for sentiment analysis**. It consists of:

- 50,000 movie reviews, rating from the Internet Movie Database (IMDB)

#### ◆ What is OpenMP?

**OpenMP** (Open Multi-Processing) is an **API** that supports **multi-platform shared memory multiprocessing programming** in C, C++, and Fortran. **Speed up execution**, Reduce execution time

⇒ `g++ -fopenmp your_code.cpp -o your_program`

> `g++ -fopenmp name.cpp -o search`

> `./search`

### ◆ #pragma omp parallel

#pragma omp parallel is a **directive** in OpenMP that tells the compiler to **create multiple threads** to execute the following block of code **in parallel**.

### preprocessor directives:

- #include <iostream>:

This line includes the iostream library, which provides standard input/output functionalities. It allows the program to interact with the user through the console, using objects like cin for input and cout for output.

- #include <omp.h>:

This line includes the OpenMP library, which provides support for parallel programming. It enables the program to utilize multiple threads for concurrent execution, potentially speeding up computationally intensive tasks.

- using namespace std;

This line brings the standard namespace into the current scope. The standard namespace contains commonly used C++ components like cin, cout, and various data types. By using this line, you can use these components directly without explicitly specifying the std:: prefix.

- #include <vector>

vector is a dynamic array that can resize itself automatically when elements are added or removed. It is part of the C++ Standard Template Library (STL) and provides a sequence container for storing elements of the same data type.

- #include <climits>:

This line includes the climits library, which defines various limits for integer data types. It provides constants like INT\_MAX and INT\_MIN representing the maximum and minimum values that an integer variable can hold.