Threads and Concurrency

1. Explain what the advantages and disadvantages of parallel programming paradigm (shared memory) are in front of the distributed programming paradigm (shared nothing).

Both paradigms have their own advantages and disadvantages. Shared memory has easier communication, with data sharing being easy and intuitive, requiring simpler programming. With this communication being done though the shared memory, it means faster speeds than inter-process communication. In addition, data consistency is easier due to a shared data view. The direct access to shared data structures translates to easier synchronization.

On the other hand, its scalability is limited by the capacity and performance of the memory subsystems, where resource contention will become a bottleneck. It also carries certain hardware dependencies, with an architecture fit to specific hardware configurations that limit portability and adaptability.

Shared nothing architectures present opposing stances on most of these issues. They have high scalability, with each node operating independently. Fault tolerance where the failure of a node does not impact the others. Hardware flexibility for each node, easy to adapt to different hardware environments because of the flexibility in choices for each node. Data partitioning, making for efficient the handling of large data sets. But they still carry the disadvantages of this kind of system, with a high degree of complexity in programming, slow inter-node communication due to overhead, and data consistency challenges with nodes accessing and editing the same data.

2. Describe three types of real-world applications of parallel programming and three types of real-world applications of distributed programming.

Parallel Programming:

- Tracking, processing and storing big data
  The use of a shared memory allows for better handling of these data. All the different nodes manage this data, while the different tasks can be done concurrently between the different nodes.

- Multimedia sharing

  The files are stored in a cloud server where each node (Device that has access to the multimedia) can store some of these files but needs to retrieve the others from the shared memory system.

- Online search engines

  There are main memory devices that store all this data, while the different nodes retrieve for each search query the data on the shared memory system.

Distributed Programming:

- Computer networks (internet)

  Computers sharing resources distributed in nodes. Using communication protocols to send data between the different nodes. Each node acts as part of a mesh of computers.

- Massive multiplayer online games

  The system is distributed on different servers. Each server occupies a different geographical zone. In some cases, this can change depending on demand. Each server acts independently.

- World Wide Web

  Each computer stores information. This information is sent between these different computers that act as nodes for information transferring. It creates a mesh of computers connected by the internet.

6. Justify the difference between using the run method and the start method of the Thread class.

There is a big difference between both methods of the Thread class. First, the `run()` method does not create a new thread, thus, by using it we will not be using multithreading. It will just execute the behavior we have specified on the run method of the class. While if we use the `start()` method, we have a new thread created each time, with each thread executing the `run()` method. The difference between both methods is that start creates a new thread, the behavior will be the same as both functions will execute the `run()` method. In our case, we will use `start()` as we want multithreading functionalities that execute in parallel.