Applied Big Data and Cloud Computing (GIK2Q3)

Assignment-2

Charu Bisht (h21chabi@du.se)

Binara  Siriwardhana ([h22bisir@du.se](mailto:h22bisir@du.se))

Mouayad Mouayad ([h20moumo@du.se](mailto:h21moush@du.se))

Reza Derakhshan([v22rezde@du.se](mailto:v22rezde@du.se))

Samaneh Ghanipour([v22samgh@du.se](mailto:v22samgh@du.se))

Soroush Yousefzadeh ([v22soryo@du.se](mailto:v22soryo@du.se))

**Report**

**Serial Computing:**

Serial computation is the traditional type of computation in which one process is carried out on one processor at a time. This type of computation is generally slower and requires more user time than parallel computation.

We can understand the functionality of serial processing using the following analogy. Assume a supermarket with multiple queues and only one cashier. The cashier finishes billing the products of one customer and then moves on to another customer. He performs billing one after the other.

Cord:



Output: It took only 1513 microsecond to finish reading the text file we used, following screenshot is the output

Text

Description automatically generated with medium confidence

**Parallel Computing:**

Parallel computation is the type of computation in which multiple processes are carried out simultaneously on multiple processors. This type of computation is generally faster and requires less user time than serial computation. Parallel computation also enables more complex tasks to be completed in a shorter amount of time.

We can understand the functionality of parallel processing using the following example. In a supermarket, there are multiple queues, and there is a cashier for each queue. Each cashier bills the products of the customers in his own queue.



Output:

Text

Description automatically generated

**Conclusion:**

In our case, we used two text files for this experiment. We used both serial and parallel computing processing for reading the text files. Usually, parallel processing is faster than serial processing, but in our scenario, we concluded that it is vice versa. There can be many possible causes for that, such as,

* One of the reasons causing the parallel process to take longer than the serial process is overhead of creating and managing multiple processes may be high in parallel processing.
* The amount of data being processed, and the number of processes being used could also affect the performance of the parallel process.
* The Pool class and Map function are some specific implementations of the parallel process, in this scenario, they could also impact its performance.
* The specific machine or system that the code is running on may not have enough resources, such as memory or CPU power, to handle the parallel process efficiently.