

REPORT – QUESTION 3

This question asked us to compare the performance of normal merge sort (NMS) against another version of merge sort which creates a separate process for each partition and uses shared memory between the forked processes (SMMS – Shared Memory Merge Sort).

The input was a text file which contained randomly generated numbers.

The number of integers in the different input files were 3000,6000,9000,12000,15000 and 18000. The tests were stopped at 18000 integers as beyond this number, the SMMS was unable to fork new processes.

In each instance the NMS outperformed the SMMS by a huge margin as one can see from the charts. (Please refer attached .png charts)

The main reason for this is the forking that we perform for each partition in SMMS. This creates a lot of child processes for huge input which results in sluggish performance.

A large number of child processes, though they are using shared memory, implies

- a) More Context Switches
- b) More CPU Migrations
- c) More Page Faults

It can be seen from the respective charts that SMMS has a huge number of context-switches, cpu-migrations and page-faults as compared to NMS.

More number of processes leads to a lot of time consumed in switching between processes.

It is also observed that though SMMS utilizes more CPU than NMS, it still is very slow in comparison due to the time consumed for switching between processes.

On creating a new process for each partition, we can sort that partition in parallel with other partitions. But this gain is overwhelmingly negated by the huge number of processes created which though executed in parallel, have to be context switched in and out between each other due to the limitation of the number of CPUs. There are also a large number of CPU migrations in each case of the input in SMMS as compared to NMS.

So to improve the performance and attain optimality one needs to ensure that the number of processes forked equals the number of CPUs. In this way each Partition which is a different process runs independently of other such forked processes on a separate CPU. (Note that context switches and CPU migrations will still occur as the OS has to schedule other processes on these CPUs as well)

But the performance of SMMS will be improved vastly and it will beat NMS which is just a single process sorting serially.