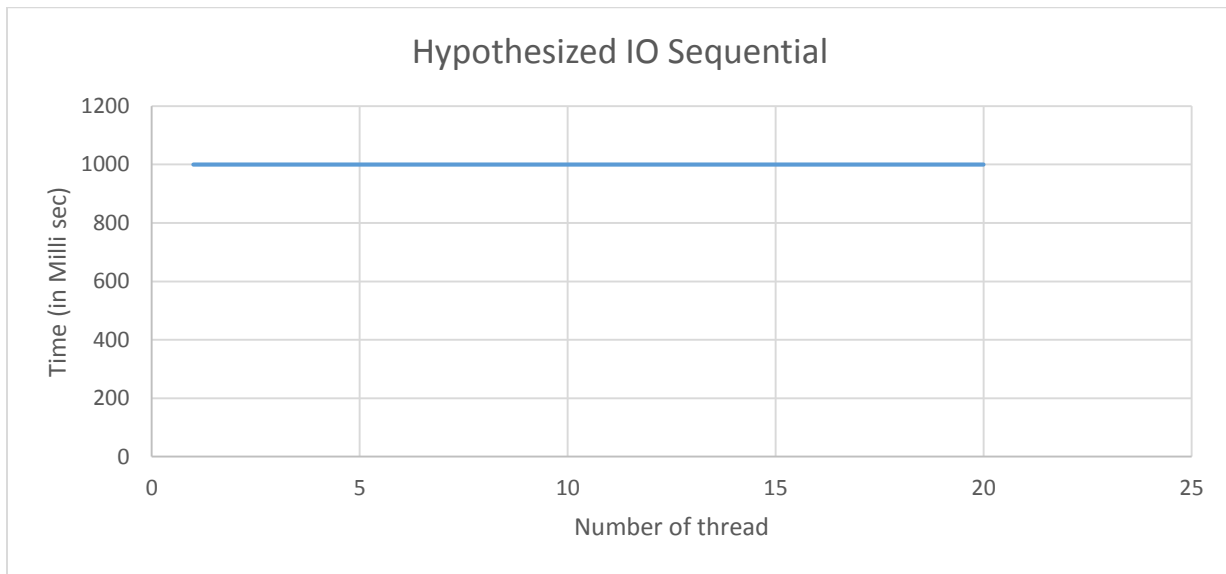


3.a) Total run time of a sequential implementation that runs in a single thread for I/O Task

The hypothesized I/O curve should be straight line as it is independent of number of thread.

Therefore curve should be $Y = C$ (where C is a constant) [Time is constant].

Curve as follows:-

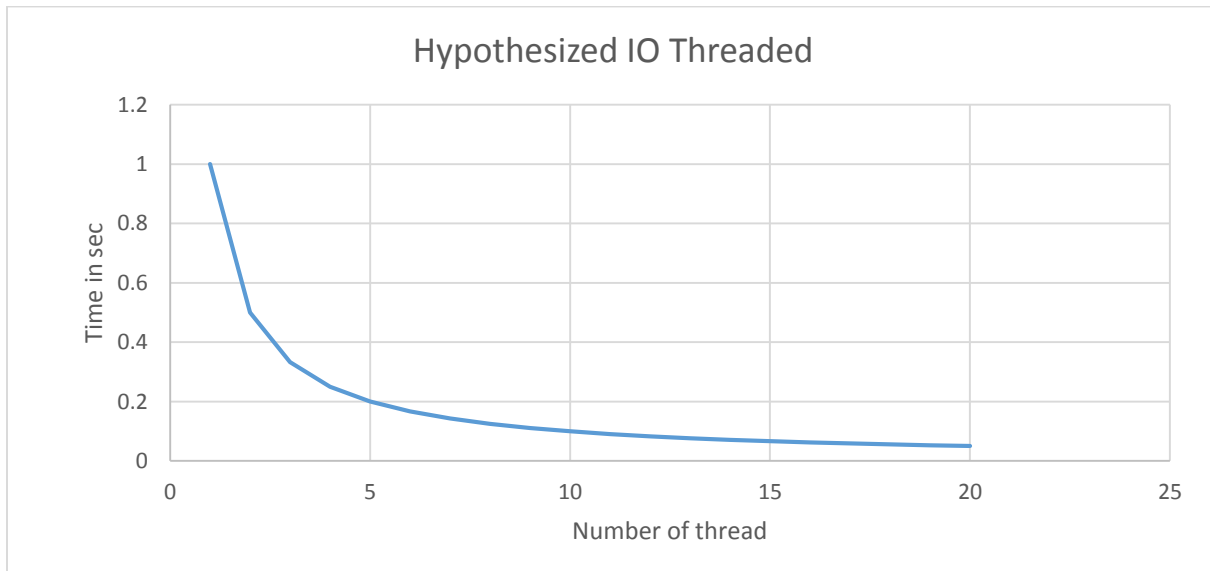


Total run time of the threaded implementation against number of threads for I/O Task

Time elapsed should drastically decreases with the increase in number of thread.

Therefore curve should be $Y = 1/X$ (where X is the number of thread spawn)

Curve as follows:-

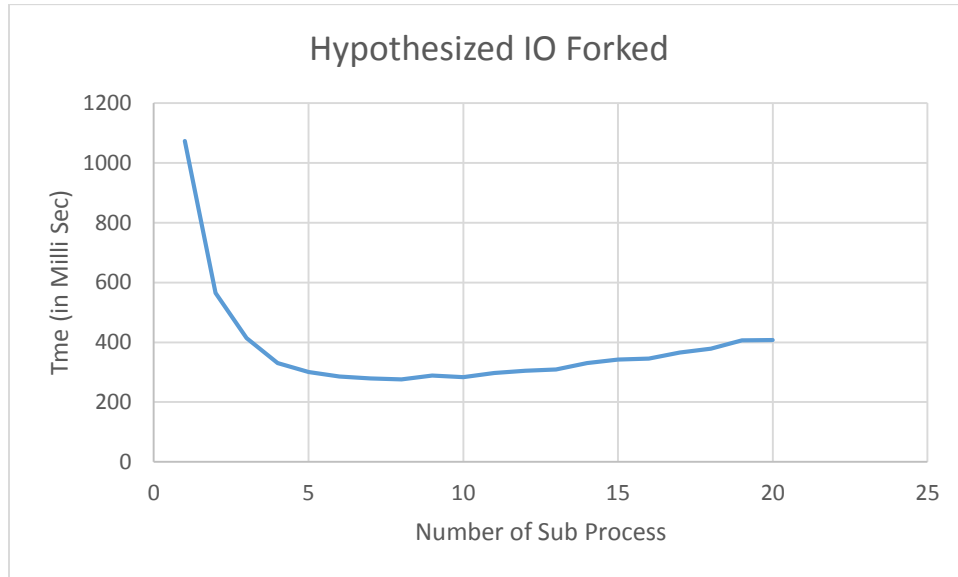


Total run time of the multi-process implementation against the number of processes for I/O Task

Time elapsed should drastically decreases with the increase in number of process.

Therefore curve should be $Y = 1/X$ (where X is the number of process). But with increase in number of process there will be significant impact because of creating new process. Hence after creating certain processes time elapsed will increase with increases in sub-process.

Curve as follows:-

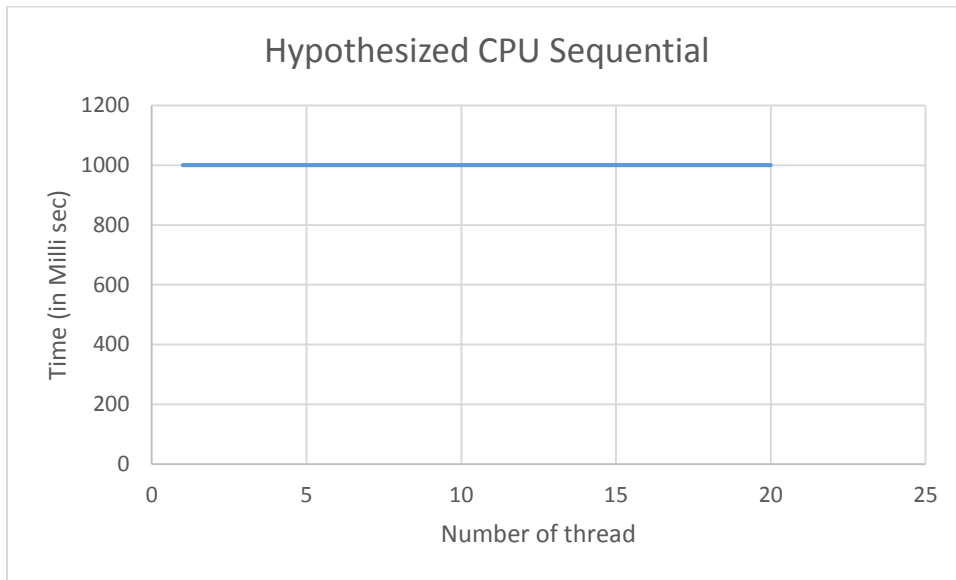


3 b.) Total run time of a sequential implementation that runs in a single thread for CPU Task

The hypothesized CPU curve should be straight line as it is independent of number of thread.

Therefore curve should be $Y = C$ (where C is a constant) [Time is constant].

Curve as follows:-

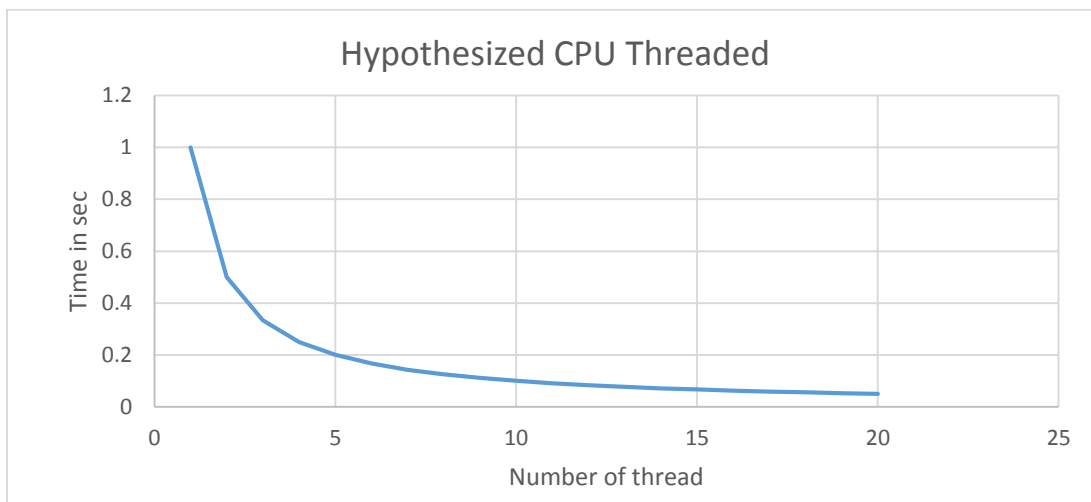


Total run time of the threaded implementation against number of threads for CPU Task

Time elapsed should drastically decreases with the increase in number of thread as more thread will parallel execute the task on CPU.

Therefore curve should be $Y = 1/X$ (where X is the number of thread spawn)

Curve as follows:-

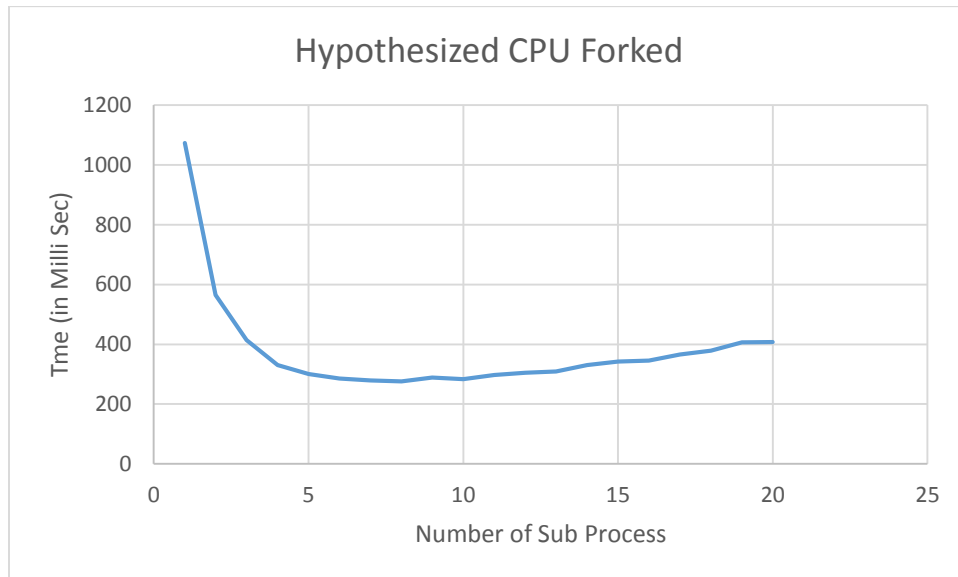


Total run time of the multi-process implementation against the number of processes for CPU Task

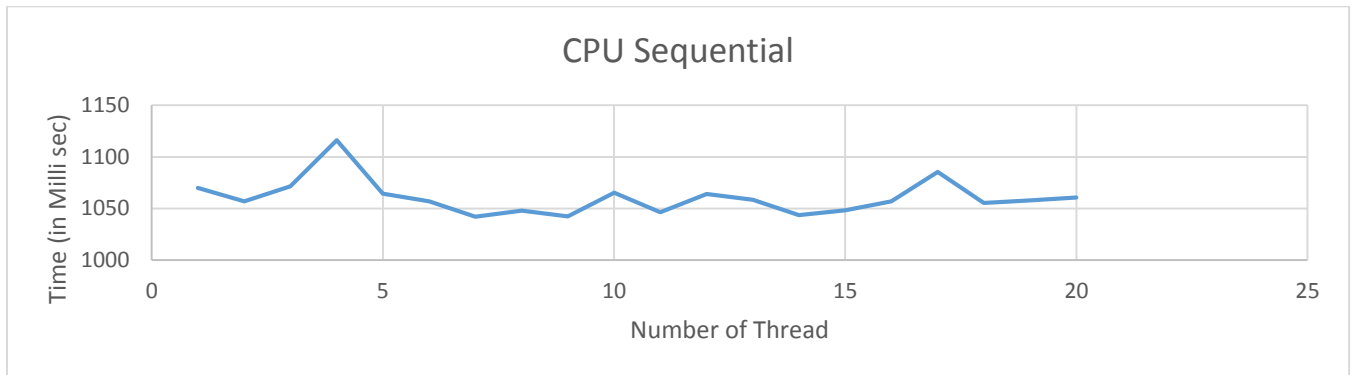
Time elapsed should drastically decreases with the increase in number of process.

Therefore curve should be $Y = 1/X$ (where X is the number of process). But with increase in number of process there will be significant impact because of creating process. Hence the time elapsed will increase later with increase in process.

Curve as follows:-

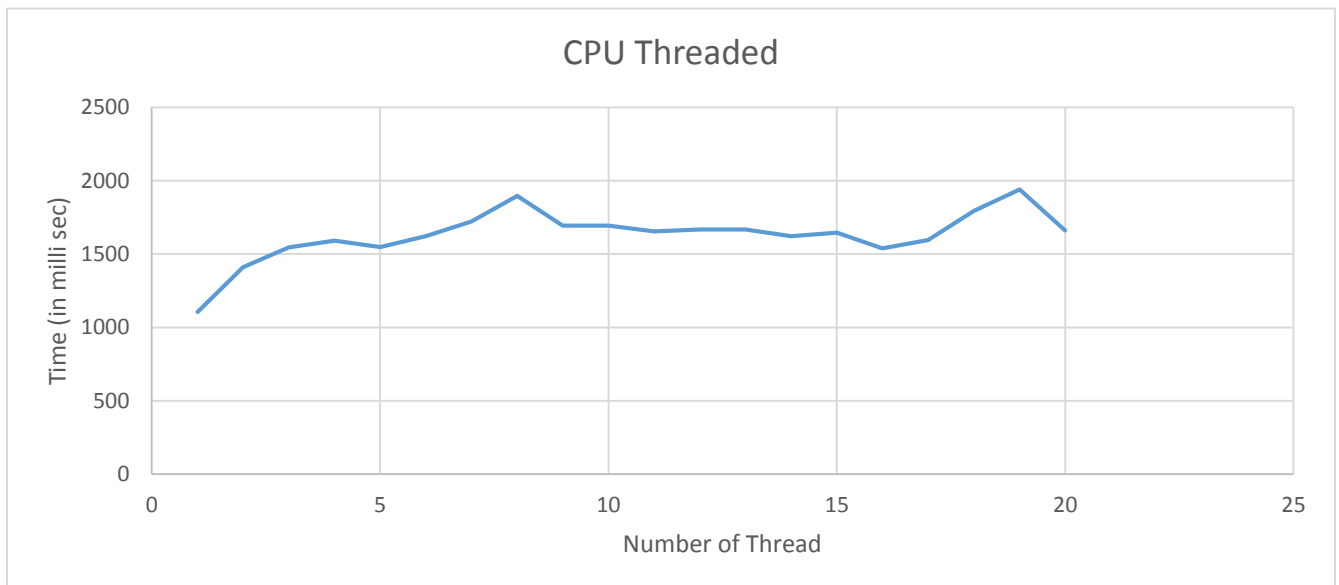


3 d) Total run time of a sequential implementation that runs in a single thread for CPU Task



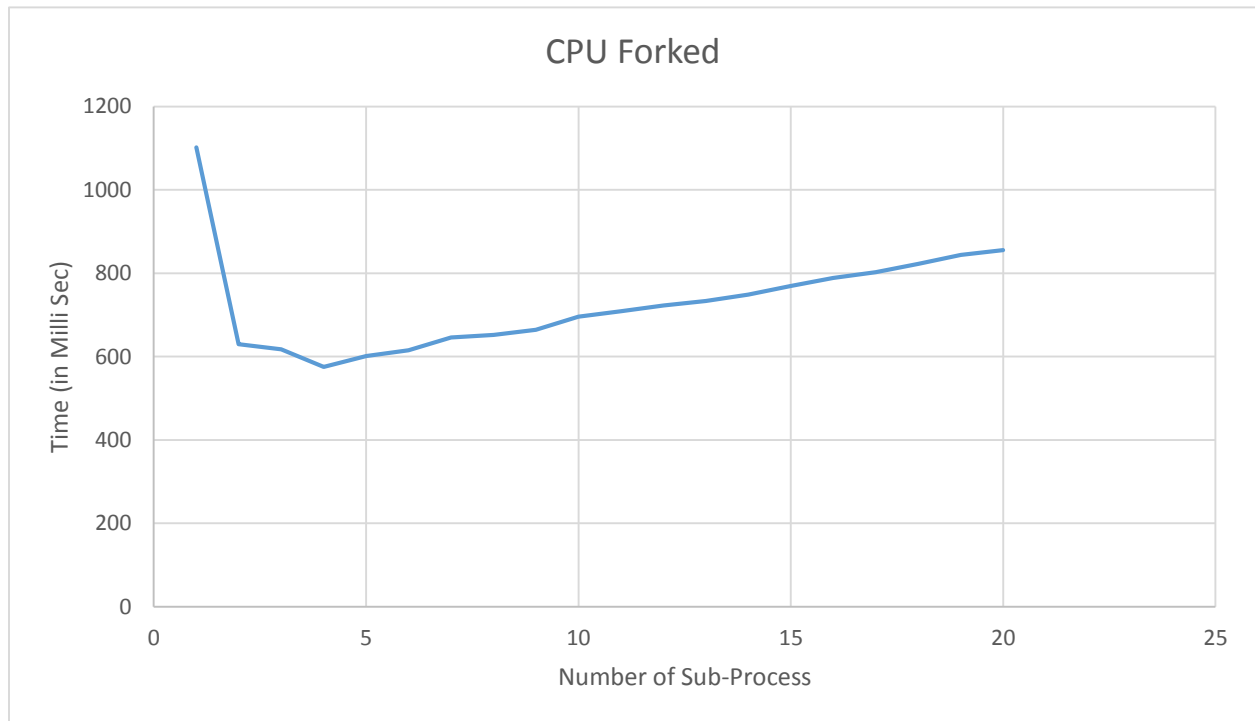
The curve is almost a straight line which is expected as it is independent of number of thread.

Total run time of the threaded implementation against number of threads for CPU Task



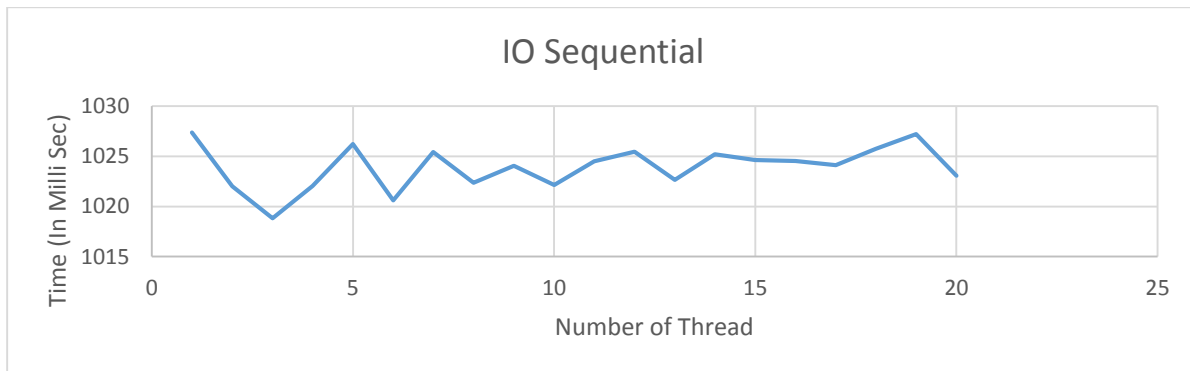
The curve indicates that the time elapsed is slightly increasing with number of thread. This seems to be because of internal implementation of thread in python which is not taking use of advantage of parallelism or may be scheduling the threads on same core which result in frequent context switch resulting in increase overall execution time.

Total run time of the multi-process implementation against the number of processes for CPU Task



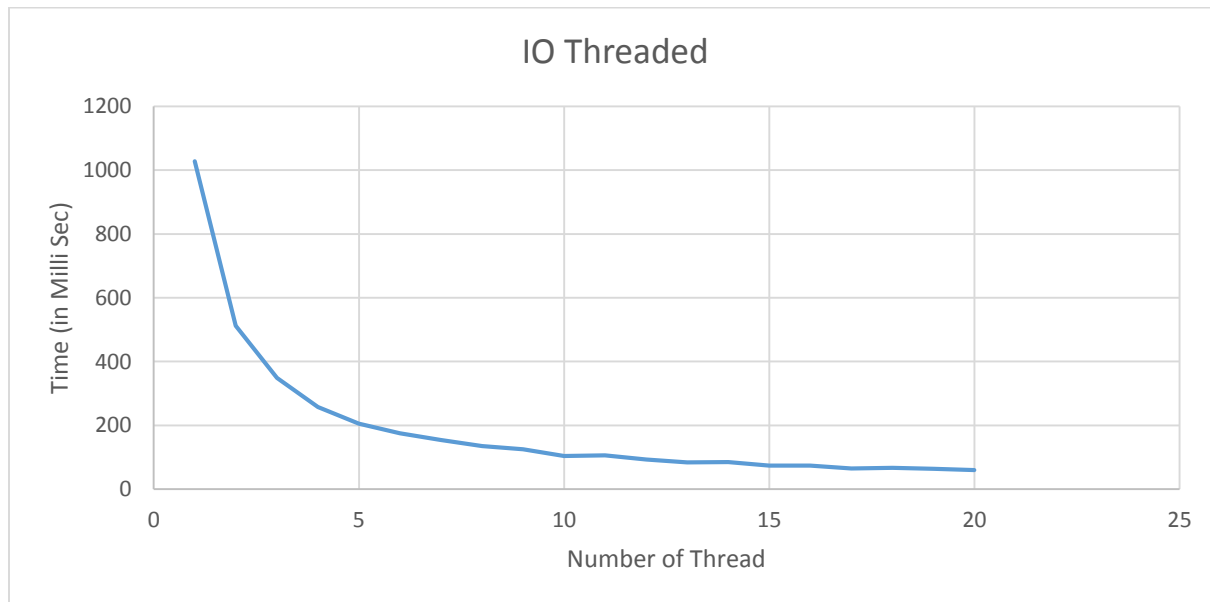
From the Curve we can observe that the time elapsed is drastically decreases with the increase in number of sub process which is happening because of parallelism. But after a certain process the overhead of creating new process is significantly impacting the overall task. Hence time elapsed start increasing with further sub-process.

Total run time of a sequential implementation that runs in a single thread for I/O Task



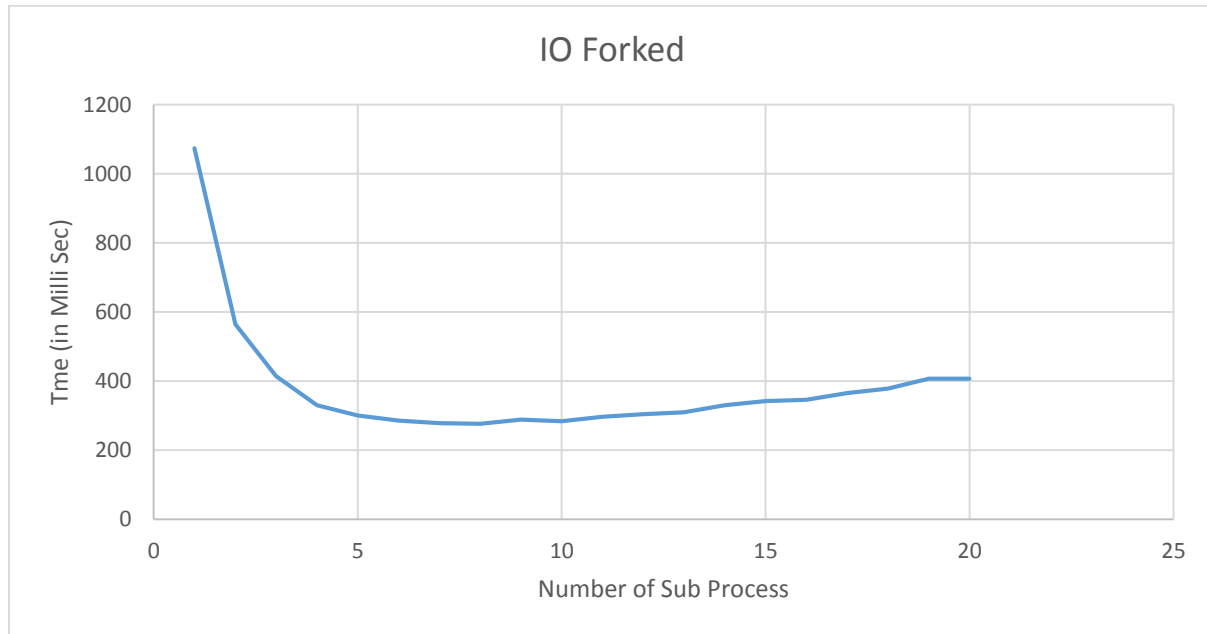
The curve is almost a straight line which is expected as it is independent of number of thread.

Total run time of the threaded implementation against number of threads for I/O Task



From the Curve we can observe that the time elapsed is drastically decreases with the increase in number of thread which is happening because of parallelism. The threads parallelly carry out the task resulting in significant reduction of overall task.

Total run time of the multi-process implementation against the number of processes for I/O Task



From the Curve we can observe that the time elapsed is drastically decreases with the increase in number of sub process which is happening because of parallelism. But after a certain process the overhead of creating new process is significantly impacting the overall task.