

Lab 2

2.1 Pthread Creation [3m]

1. [1m] In pthread create, what does the first argument mean?

1st -> pointer to the memory address where the ID of the newly created thread should be stored

2. [2m] What about the third and the fourth? (ignore the second argument for now).

3rd -> pointer to a function for the thread to execute (thread executes user code at this address)

4th -> pointer to the arguments passed to function (the 3rd argument)

2.2 Pthread Termination [2m]

1. [1m] What is this function for?

This function terminates the calling thread.

2. [1m] What does the argument mean?

Pointer to the return code or return value of the thread.

2.3 Pthread Join [4m]

1. [1m] What is this function for?

This function suspends execution of the calling thread until all the target threads terminate. It cleans up any resources associated with the target thread.

2. [2m] What are the reasons on why do we need to do this?

When we want to wait for the thread to finish or if we want to clean up any resources associated with the thread. Without pthread_join, the calling thread will exit after it has completed its tasks without caring about other threads it has created.

3. [1m] Does it block the main thread if the target thread has yet to terminate?

Yes.

2.4 Further Observation [2m]

1. [1m] Do you get the same output each time you run the program?

No.

2. [1m] Why do you think so?

The ordering depends on the thread scheduler library.

3 Speeding Up Processes with Pthreads: Matrix Multiplication

Task [9m]

1. Given M, N , and K , what is the dimension of the output matrix $M_3 = (M_1)(M_2)$?

Dimension of M_3 is $M \times K$.

4 Analysis [5m]

1. Suggest the optimum number of threads to perform this matrix multiplication task.

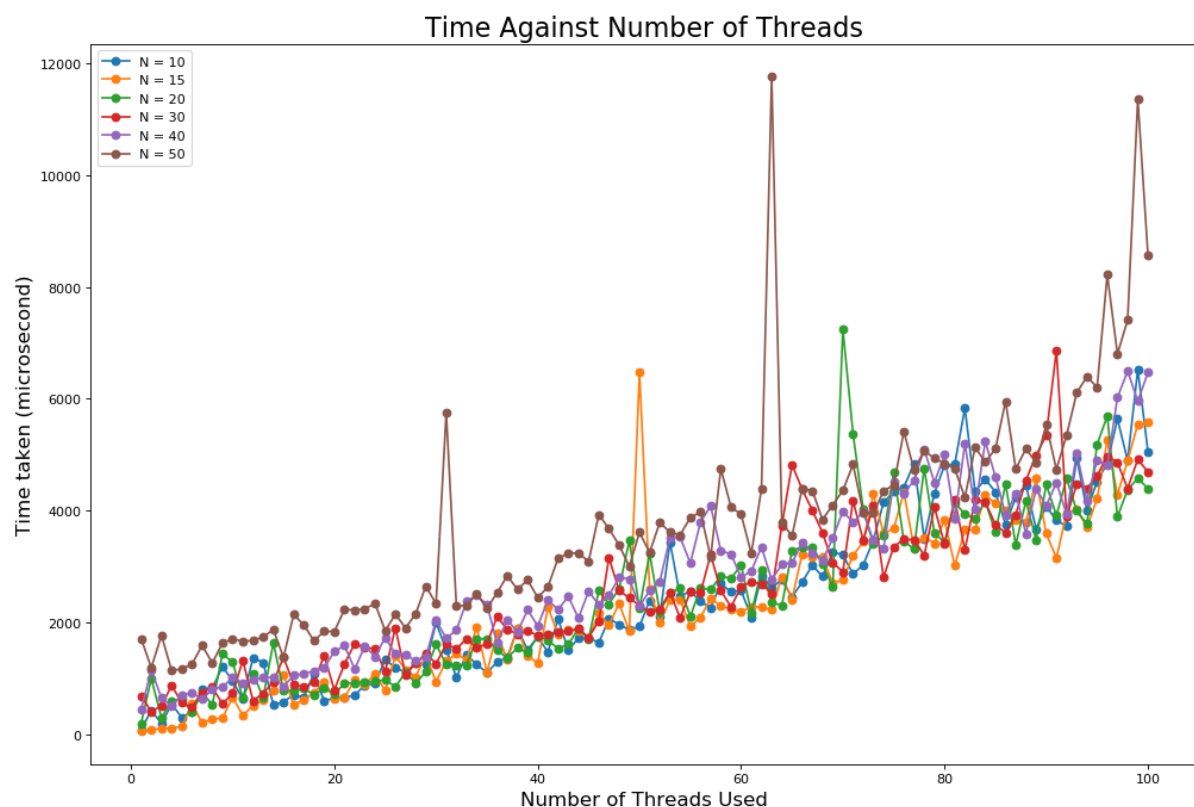
For $N \leq 50$, the optimum number of threads to perform the matrix multiplication is roughly 1.

For $N > 51$, the optimum number of threads to perform the matrix multiplication is roughly 5% of N .

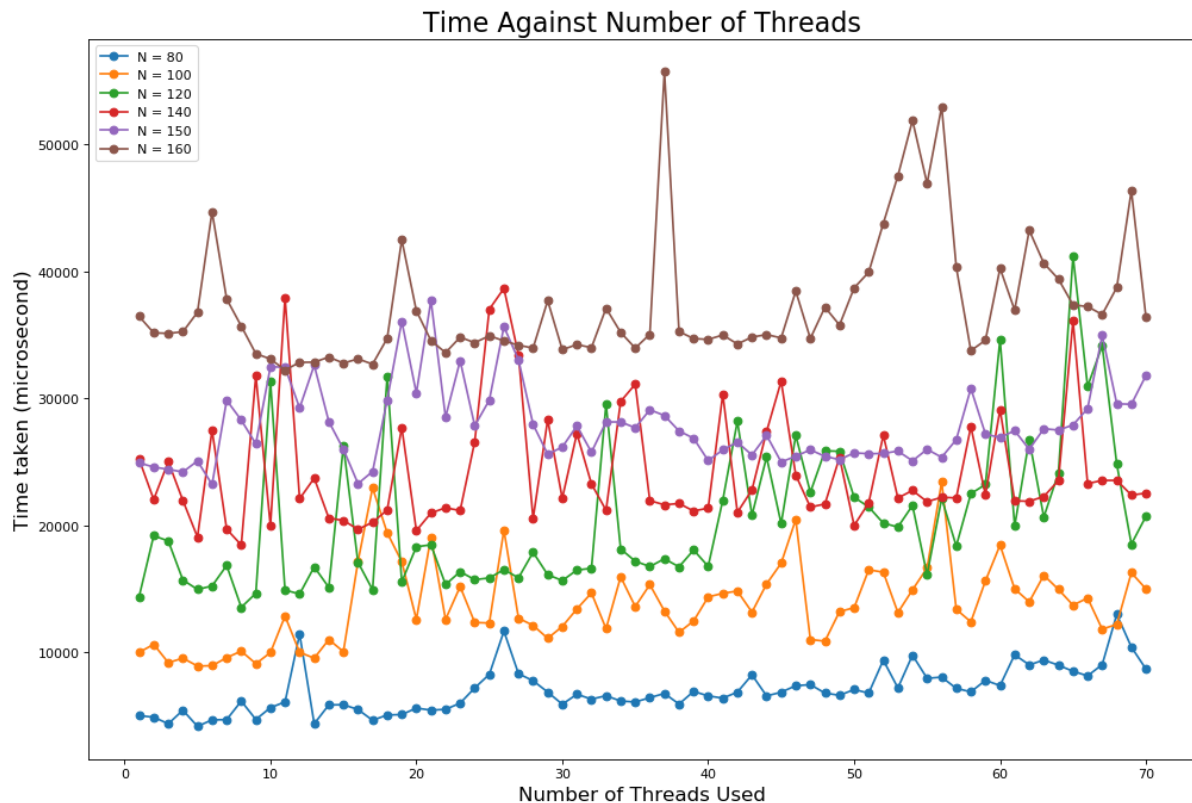
(Eg. for $N=100$, having 5 threads gives the optimum performance which is 5% of N)

2. Provide sufficient plots (time vs thread vs matrix size N) to support your answer.

For $N \leq 50$:



For $N > 51$:



5 Scheduling Policies [5m]

1. [4m] What does each scheduling type mean?

SCHED_FIFO: First-In, First-Out scheduling. A running SCHED_FIFO thread that has been pre-empted by another thread of higher priority will stay at the head of the list for its priority and will resume execution as soon as all threads of higher priority are blocked again.

SCHED_RR: Round-robin scheduling. If a SCHED_RR thread has been running for a time period equal to or longer than the time quantum, it will be put at the end of the list for its priority.

SCHED_OTHER: Default Linux time-sharing scheduling. Intended for all threads that do not require the special real-time mechanisms, ensures fair progress among all SCHED_OTHER threads.

SCHED_DEADLINE: Sporadic task model deadline scheduling. A sporadic task is one that has a sequence of jobs, where each job is activated at most once per period. Each job also has a relative deadline, before which it should finish execution, and a computation time, which is the CPU time necessary for executing the job.

2. [1m] Which of these are the confirming to “real-time” policies specified by POSIX standard?

SCHED_FIFO and SCHED_RR.

6 Pthread Priority [5m]

(a) [1m] Does higher number represent higher priority? You may want to Google this.

No. The lower the number, the higher the priority.

(b) [1m] Is thread scheduling pre-emptive? You may need to do some googling to answer this.

Yes. Pre-emptive scheduling is more common.

(c) [1m] Is there any difference in the order of output after we set the priority of the threads, as compared to when we did not set any priority?

No.

(d) [2m] Now set the priority level for each thread to be the same. Do you notice any difference in the order of output? Why do you think so?

No, it is still in random order. The kernel can interrupt the thread library. We can give priority to certain threads using the thread library but the kernel can choose not to listen to the thread library.