

## Question 2

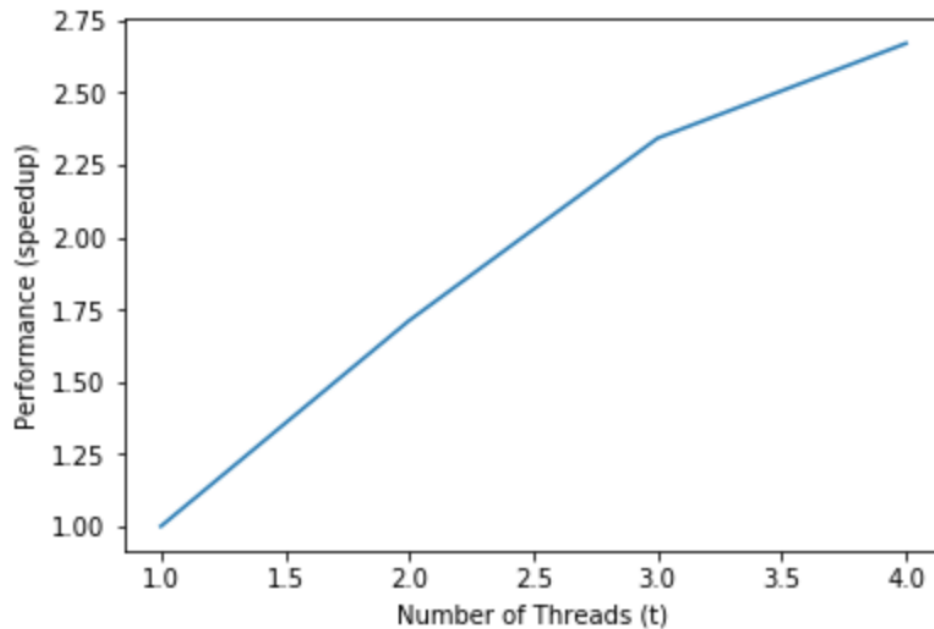


Fig. Relative *speedup* of multithreaded versions over the single-threaded version for 2, 3, and 4 threads.

From the above figure, it is observed that increasing the number of threads results in significantly better performance initially. However, after reaching a certain number of threads, the speedup starts to flatten.

A multi-core machine allows us to split a program task among multiple threads. This permits simultaneous use of multiple cores for computation and more efficient utilization of CPU resources. Hence, multithreading is expected to allow better performance.

A program can speed up execution only so far, even if we use more and more threads to do the work. The reason for this could be one or more of the following :

1. Performance depends on how much of our program can be transformed for parallel execution. The smaller the parallelizable portion, the less speedup we can achieve using multiple cores.
2. Our multi-core machines have a limited number of cores. Therefore, some number(s) of threads will allow speedup, but not all values. Threads can be sharing the same processor/core.
3. Having too many threads means too many context switches which slows down the program.

Screenshot of the observed values used to plot the graph above.

```
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 1 20
Execution time : 244ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 1 20
Execution time : 241ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 1 20
Execution time : 217ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 1 20
Execution time : 263ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 1 20
Execution time : 206ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 1 20
Execution time : 279ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 2 20
Execution time : 137ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 2 20
Execution time : 132ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 2 20
Execution time : 143ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 2 20
Execution time : 154ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 2 20
Execution time : 131ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 2 20
Execution time : 150ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 3 20
Execution time : 109ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 3 20
Execution time : 97ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 3 20
Execution time : 112ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 3 20
Execution time : 93ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 3 20
Execution time : 105ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 3 20
Execution time : 103ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 4 20
Execution time : 85ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 4 20
Execution time : 99ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 4 20
Execution time : 91ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 4 20
Execution time : 92ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 4 20
Execution time : 94ms
[mobasserazaman@Mobasseras-MacBook-Pro desktop % java fault 1000 1000 4 20
Execution time : 82ms
```

