CS3523 : OPERATING SYSTEMS 2
Programming Assignment 2

MANOJ KUMAR
ES21BTECH11010

Contents

1 Goal :

- Goal is to design a multithreaded program which will determine whether a solution to a Sudoku puzzle is valid or not using Pthreads and OpenMP and comparing the two implementations.

2 Input :

- First line of Input is pair of numbers:

  1. Number of Threads to be used in the execution of the program (*K*).
  2. Size of sudoko used in the experiment (*N*).

- The following *N* lines contain the sudoko with each row having *N* columns.

3 Output :

- We have two output files for *Pthread* and *OpenMP* respectively :

  1. Log data of each thread.
  2. The validity of the sudoko.
  3. Time of execution of the program.

4 Analyzing Program Output :

- We analyze the program in two modes:

  1. Number of threads which the program uses is constant (*K* is constant).
  2. The dimension of the sudoko is constant (*N* is constant).

4.1 *K* is constant :

- The value of *K* used by the program is fixed and is equal to 16.

- Now we see the variation of time of excecuiton of program (T) by varying the value of dimension of sudoko (*N*).

4.1.1 Graph :

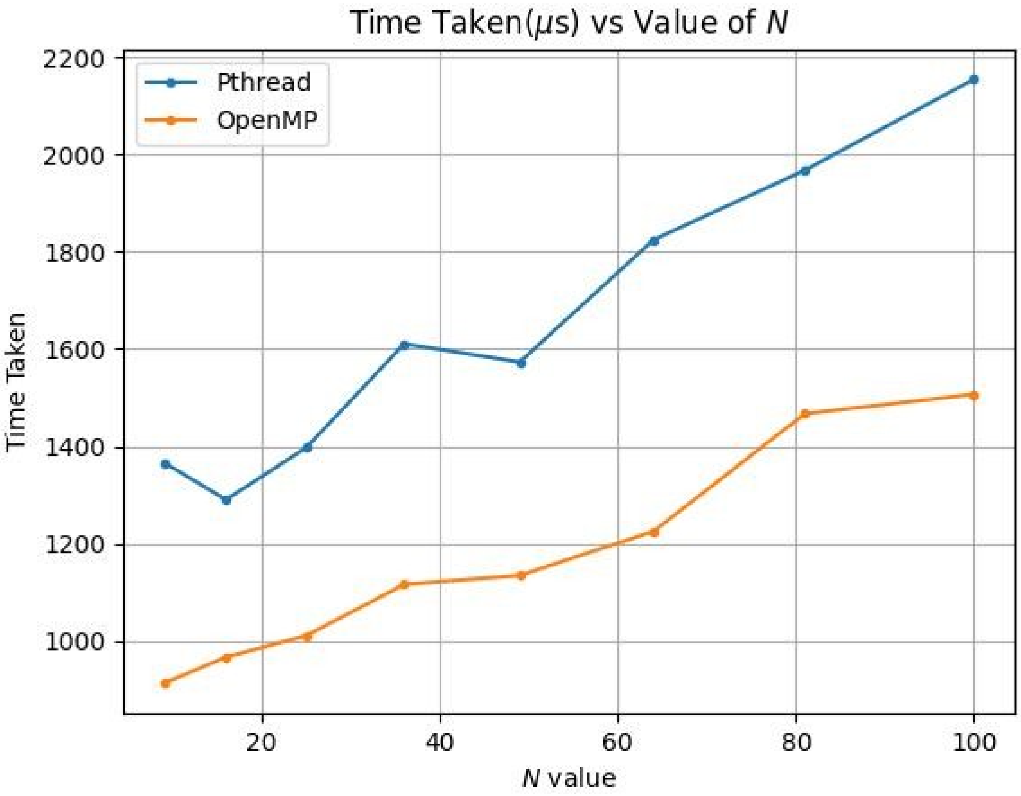

Figure 1: T vs N (K is constant)

4.1.2 Observations :

- **Expected** : As N increases T increases.
- **Observed** :
  - The performance of OpenMP is better than Pthread.
  - As expected the value of T increases regularly in case of OpenMP on increasing N.
  - In case of Pthread the graph looks to be increasing overally but there are small depressions at some values of K, this signifies that the peformace in case of those two threads is nearly same (as we are working with time in μs).

4.2 N is constant :

- The value of N is fixed and is equal to 25.
- Now we see the variation of time of excecuiton of program (T) by varying the total number of threads (K).

4.2.1 Graph :

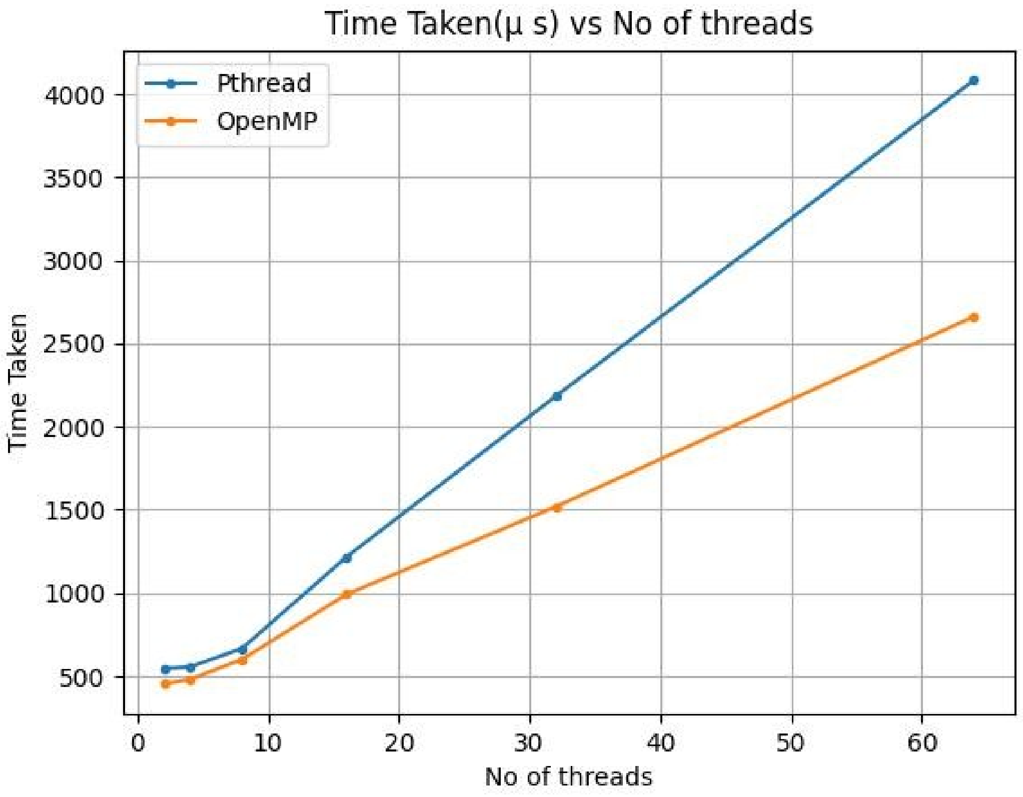

Figure 2: T vs K (N is constant)

4.2.2 Observations :

- **Expected** : As K increases T decreases.
- **Observed** :
  - From the above graph we can conclude that OpenMP is faster than Pthreads.
  - The value of time increases as K increases because as the value of N beign low.
  - Generally threads work efficiently when we have a large amount of data, because in this context using threads makes sense, as splitting of work matters here.
  - But when we have only less data doing the using threads is inefficient as at the first go there is no need of threading to be used.
  - So here now as $N = 25$, which is a low value and which doesn't need threading implementation.
  - This is the reason why in this case as K increases the execution time T increases.

5 Final Conclusion:

- Using OpenMP we can do threading more fastly and efficiently than Pthreads.