Namal University Mianwali

Department of Computer Science

PDC Assignment 02

Student:
Mahmood Yousaf
(2018-uet-nml-cs-11)

Instructor:
Dr Imran Ashraf

October 30, 2021

Contents

1 Task 01

factorial_using_omp.cpp × mulitplication_using_omp_dynamic_memory_allocation.cpp

factorial_using_omp.cpp

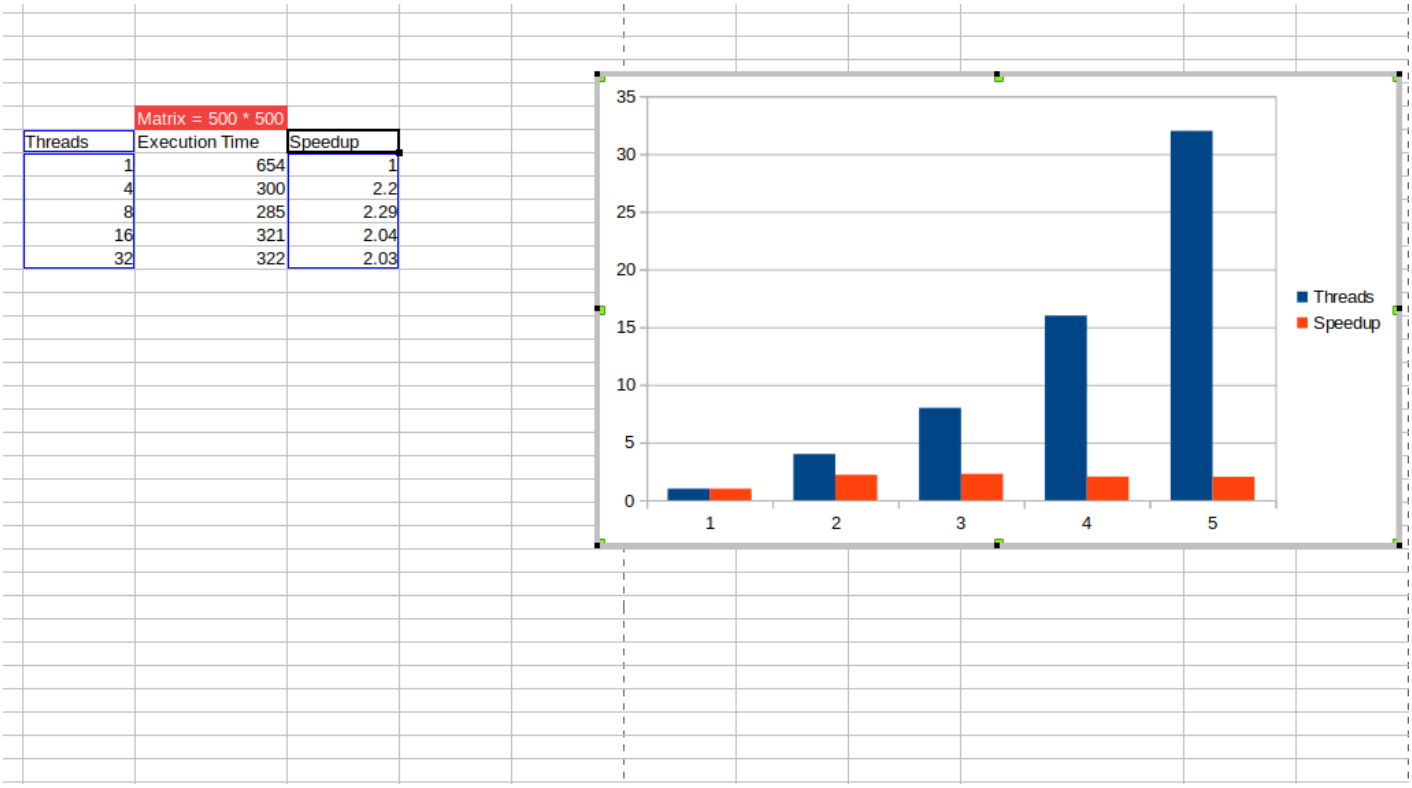
```cpp
#include <iostream>
#include <omp.h>
using namespace std;

int main(){
    long input,fact=1;
    cout << "Enter number for calculation of factorial\n";
    cin >> input;
    #pragma omp parallel for reduction(*:fact)
    for(int i=input;i>=1;i--){
        cout << i << "\n";
        fact = i*fact;
    }
    cout << fact << "\n";
    return 0;
}
```

Figure 1: Parallel Version

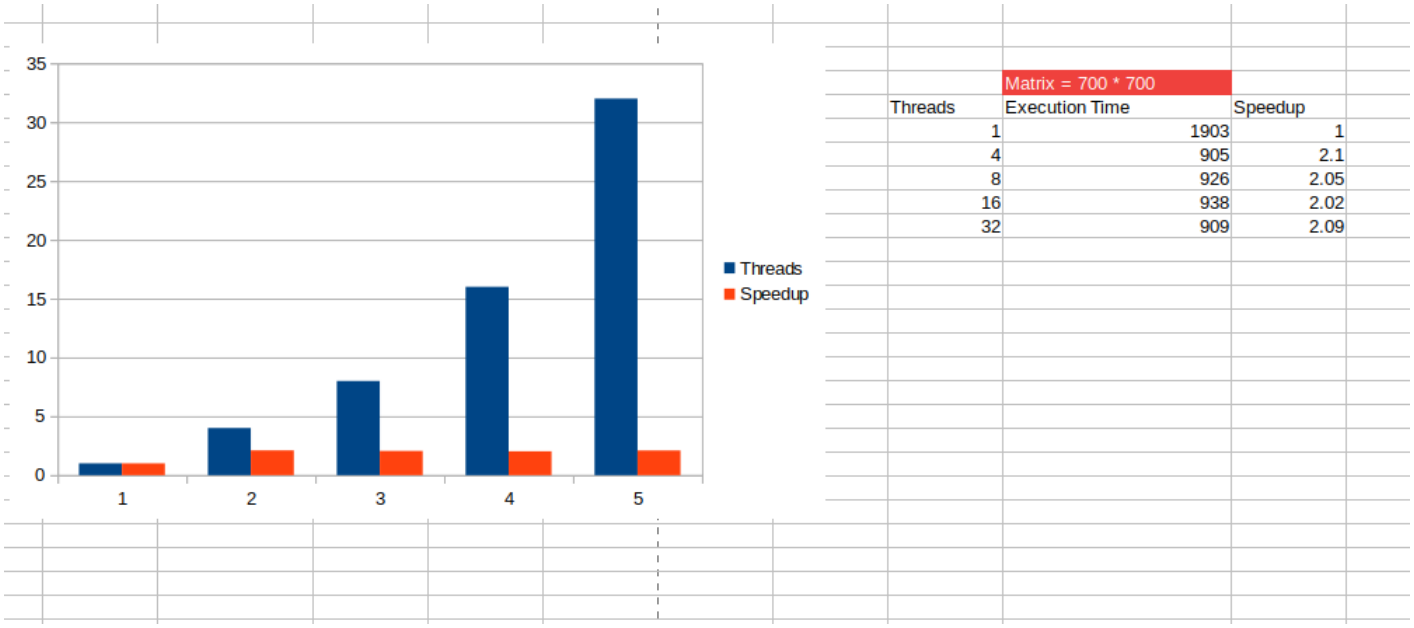
2 Task 02

Multiplication program is parallelized using openmp.Code is tested on different matrix sizes like 500* 500 and 700*700. Code is tested on dual core system and performance gain is analyzed by calculating speedup and execution time on different number of threads. Code is available on my github. [1]



	Matrix = 500 * 500	
Threads	Execution Time	Speedup
1	654	1
4	300	2.2
8	285	2.29
16	321	2.04
32	322	2.03

Figure 2: Speedup on 500*500 Matrix



	Matrix = 700 * 700	
Threads	Execution Time	Speedup
1	1903	1
4	905	2.1
8	926	2.05
16	938	2.02
32	909	2.09

Figure 3: Speedup on 700*700 Matrix

	Matrix = 500 * 500	
Threads	Execution Time	Speedup
1	654	1
4	300	2.2
8	285	2.29
16	321	2.04
32	322	2.03

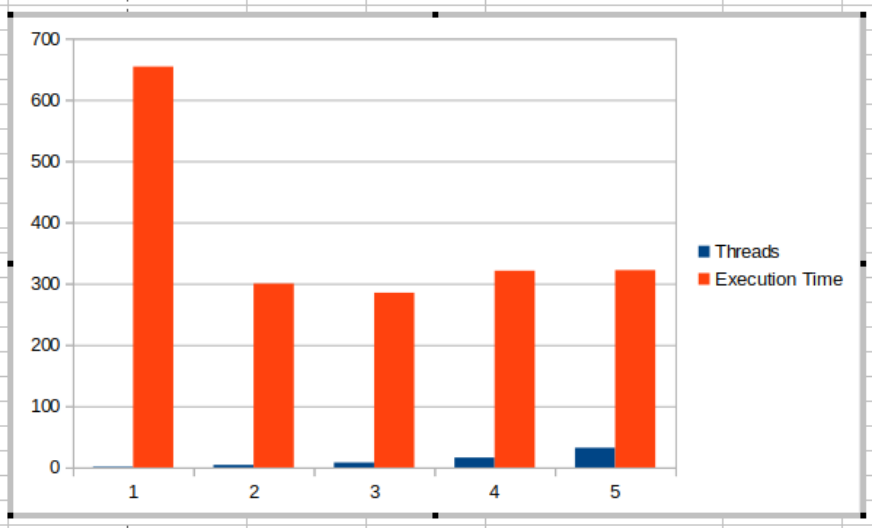
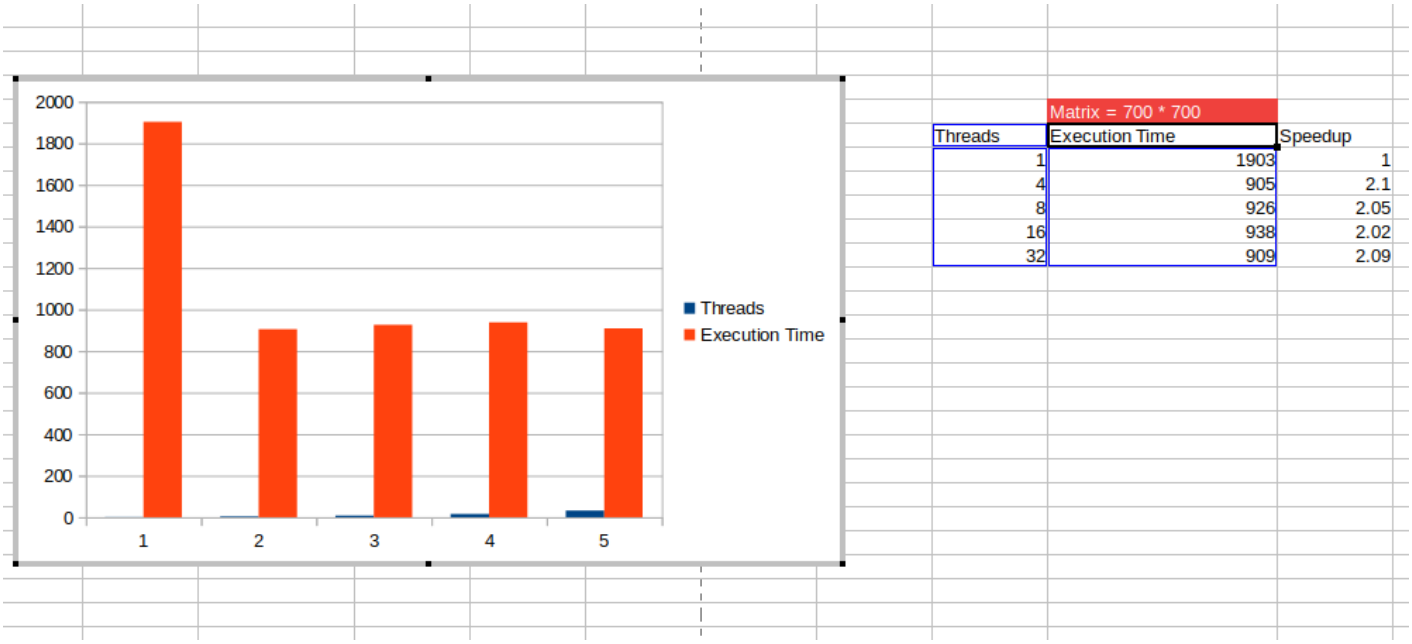


Figure 4: Execution time on 500*500 Matrix



	Matrix = 700 * 700	
Threads	Execution Time	Speedup
1	1903	1
4	905	2.1
8	926	2.05
16	938	2.02
32	909	2.09

Figure 5: Execution time on 700*700 Matrix

References

[1] Mahmood Yousaf. Parallel programming using openmp — ,, 2021. [https://github.com/yousaf2018/OpenMP-programming-].