Parallel Computing

UIN: 729007151

Priyanka Karuppuch Samy

Q1. The code is available in the file attached along with the zipped file.

To compile the code, I used commands

g++ -c Rinverse_final.cpp -fopenmp

g++ -o exec Rinverse_final.o -fopenmp

To run it, I used

CMP_NUM_THREADS = #n_threads ./exec #mat_dim

The picture below shows the execution details.

```
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=2 ./exec 10
1.32155e+10 8.46931e+08 1.68169e+09 1.71464e+09 1.95775e+09 4.24238e+08 7.19885e+08 1.64976e+09 5.96517e+08 1.18964e+09
0 1.1499e+10 7.83369e+08 1.10252e+09 2.0449e+09 1.96751e+09 1.36518e+09 1.54038e+09 3.04089e+08 1.30346e+09
0 0 1.08763e+10 1.72696e+09 3.36466e+08 8.61022e+08 2.78723e+08 2.33665e+08 2.14517e+09 4.68703e+08
0 0 0 1.67168e+10 1.36913e+09 1.1259e+09 1.05996e+09 2.08902e+09 6.28175e+08 1.65648e+09
0 0 0 0 1.4133e+10 7.56899e+08 1.73458e+09 1.97359e+09 1.49798e+08 2.03866e+09
0 0 0 0 0 1.2035e+10 1.37807e+08 4.29992e+07 9.82907e+08 1.35497e+08
0 0 0 0 0 0 8.86549e+09 1.63262e+09 1.10066e+09 1.43393e+09
0 0 0 0 0 0 0 1.60022e+10 1.37434e+09 7.60314e+08
0 0 0 0 0 0 0 0 1.02558e+10 1.47461e+09
0 0 0 0 0 0 0 0 0 1.43739e+10
Time elapsed: 0.156523ms
7.56685e-11 -5.5732e-12 -1.12984e-11 -6.22655e-12 -8.80326e-12 1.88257e-13 -2.46701e-12 -4.94988e-12 -4.5267e-13 -2.87009e-12
0 8.69644e-11 -6.26362e-12 -5.08849e-12 -1.19407e-11 -1.2542e-11 -1.0055e-11 -5.08326e-12 2.18e-12 -4.23536e-12
0 0 9.19428e-11 -9.49833e-12 -1.26874e-12 -5.60949e-12 -1.41954e-12 2.1379e-13 -1.79698e-11 3.03184e-13
0 0 0 5.98202e-11 -5.79507e-12 -5.23183e-12 -5.93697e-12 -6.47477e-12 -1.57315e-12 -4.92643e-12
0 0 0 0 7.07562e-11 -4.44995e-12 -1.37746e-11 -7.30923e-12 1.85079e-12 -8.42259e-12
0 0 0 0 0 8.30909e-11 -1.29158e-12 -9.14983e-14 -7.81248e-12 1.519e-13
0 0 0 0 0 0 1.12797e-10 -1.15081e-11 -1.05633e-11 -9.56011e-12
0 0 0 0 0 0 0 6.24913e-11 -8.37424e-12 -2.4464e-12
0 0 0 0 0 0 0 0 9.75058e-11 -1.00031e-11
0 0 0 0 0 0 0 0 0 6.95706e-11
[priyanka1331@ada8 HW2]$ vi Rinverse_final.cpp
[priyanka1331@ada8 HW2]$ g++ -c Rinverse_final.cpp -fopenmp
[priyanka1331@ada8 HW2]$ g++ -o exec Rinverse_final.o -fopenmp
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=1 ./exec 10
Time elapsed: 0.094923ms
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=2 ./exec 10
Time elapsed: 0.124289ms
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=3 ./exec 10
Time elapsed: 0.139522ms
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=4 ./exec 10
Time elapsed: 0.156234ms
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=1 ./exec 100
```

Q2) I used parallelization for recursive calls in the function that computes inverse. To improve algorithm performance, I used gauss Jordan algorithm to calculate inverse for the base conditions of recursion (i.e matrix size < 16). I created tasks for recursion operations as mentioned in the problem statement. I tried without using taskwait initially to see if the code gives correct output. But, I found out synchronization is necessary and included taskwait directive for synchronization.

Apart from this, another design choice I used was to pass by reference for function implementation.

Q3) I have calculated speed up and efficiency for matrix size 900 and 2000.

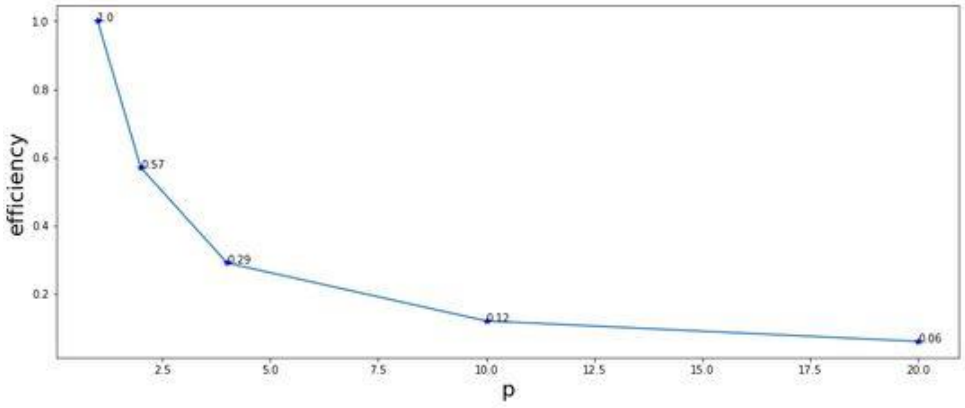
Speed up and efficiency for size 900 is:

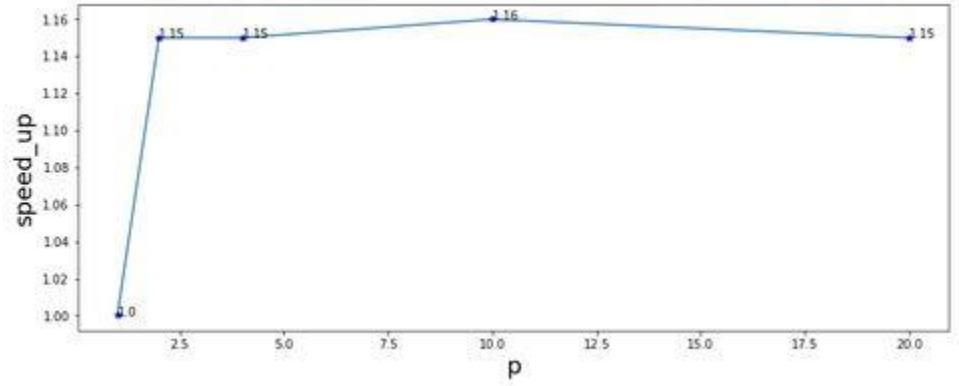
```
p =          [1, 2, 4, 10, 20]
speed_up =  [1.0, 1.15, 1.15, 1.16, 1.15]
efficiency = [1.0, 0.57, 0.29, 0.12, 0.06]
```

Speed up and efficiency for size 2000 is:

```
p =           [1, 2, 4, 10, 20]
speed_up =    [1.0, 1.13, 1.13, 1.12, 1.15]
efficiency =  [1.0, 0.56, 0.28, 0.11, 0.06]
```

```
[priyanka1331@ada8 HW2]$ echo ------------------------
------------------------
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=1 ./exec 900
Time elapsed: 4861.38ms
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=2 ./exec 900
Time elapsed: 4283.74ms
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=4 ./exec 900
Time elapsed: 4211.82ms
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=10 ./exec 900
Time elapsed: 4197.73ms
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=20 ./exec 900
Time elapsed: 4219.55ms
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=1 ./exec 1000
Time elapsed: 6682.45ms
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=2 ./exec 1000
Time elapsed: 5876.9ms
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=4 ./exec 1000
Time elapsed: 5893.21ms
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=10 ./exec 1000
Time elapsed: 5774.26ms
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=20 ./exec 1000
Time elapsed: 5786.04ms
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=1 ./exec 2000
Time elapsed: 57063.8ms
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=2 ./exec 2000
Time elapsed: 50667.7ms
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=4 ./exec 2000
Time elapsed: 50529.3ms
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=10 ./exec 2000
Time elapsed: 51010.3ms
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=20 ./exec 2000
Time elapsed: 49761.7ms
[priyanka1331@ada8 HW2]$ OMP_NUM_THREADS=40 ./exec 2000
Time elapsed: 49774.4ms
[priyanka1331@ada8 HW2]$
```


efficiency
p
1.0
0.57
0.29
0.12
0.06


speed_up
p
1.0
1.15
1.15
1.16
1.15