Assignment -2

Suyog Kharage A20402686 skharage@hawk.iit.edu

Files Used:

GuassElimination_parallel_Pthread.c

GuassElimination_parallel_openMP.c

Methods:

1. **GuassElimination_parallel_Pthread.c**

   gaussianElimination – This method is used to perform parallel gaussian elimination on matrix.

   backSubstituion – This method used to multiply the coefficient vector with the generated random Matrix.

   Synchronization: - It is performed in 'gaussianElimination' method. In each row, the threads are joined to compute common result of each element. After join of all threads, next row execution starts by forking threads on row.

   Optimization-: For optimizing the performance, we must restrict the fork and join of threads. i.e. there will be less number of thread forks and joins, and hence execution time is improved.

   Different Version tried: - I tried to parallelize innermost loop for optimization but it fails to achieve smaller execution time, because it creates more number of forks and joins as it is inner most loop.

   Example*: -*execution time of pthread code for 2000 variables and 50 threads was taking 2.4 more second than execution time of code with 2000 variables and 5 threads. As the number of threads are less it will have less fork and join.

2. **GuassElimination_parallel_openMP.c**

   gaussianElimination – This method is used to perform parallel gaussian elimination on matrix.

   backSubstituion - This method used to multiply the coefficient vector with the generated random Matrix.

   Synchronization: - It is performed in 'gauss' method.

   All threads are scheduled dynamically for efficiency achievement. This is because threads do not wait for any other thread to complete its execution. In each row, the threads are joined to compute common result of each element. After join of all threads, next row execution starts by forking threads on that row of matrix.

   Optimization-: I used the same method as we used for pthread.

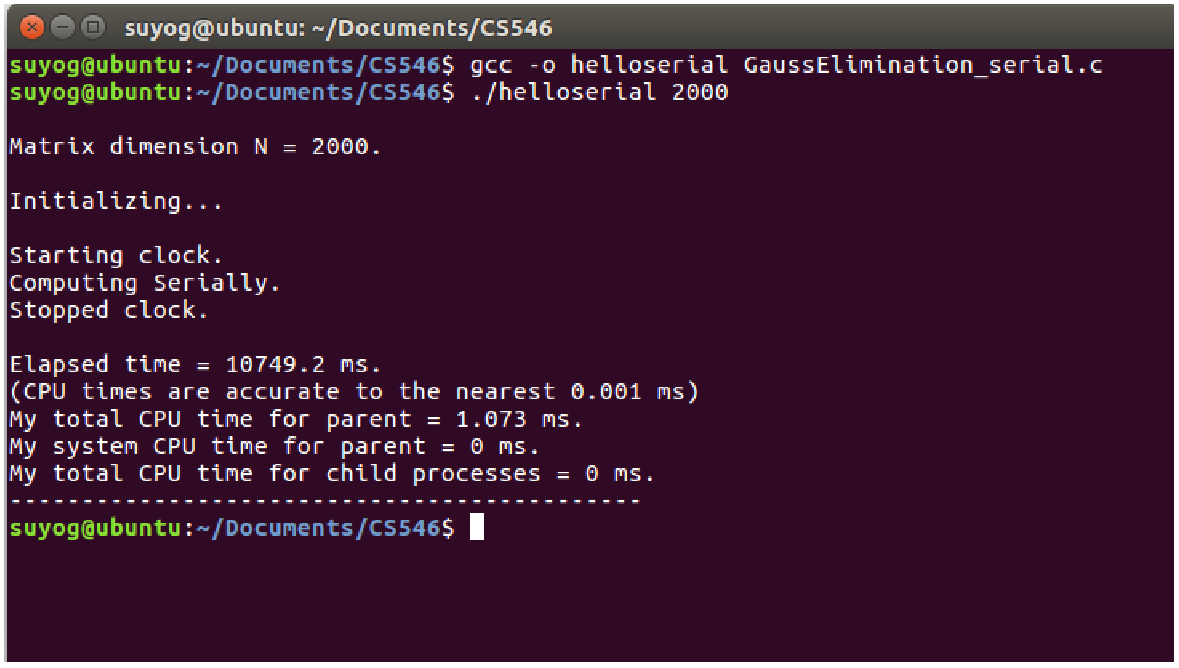
Different Version tried: - I tried to parallelize innermost loop but optimization was not achieved. The reason for this is more number of forks and joins as it is inner most loop. So, it was not efficient method.

3. **Dependencies-:**

1. In Gaussian Algorithm, there are total 3 loops that can be used to parallelize the task. Here, two inner loops are depending on the outer loop.
2. Thread index and matrix size used in 2nd loop of serial algorithm, are not used in 1st loop for any calculation. Hence there is no any dependency between these two loops. Therefore, we can parallelize 1st loop using Pthread.
3. In the 2nd loop of algorithm, statement is performing operations on different values of thread indexes and these are dependent on 1st loop. Hence 2nd loop cannot be parallelized.
4. In the innermost loop of algorithm, statement is reading the value of multiplier written by statement of 2nd loop. Hence, innermost loop cannot be parallelized.

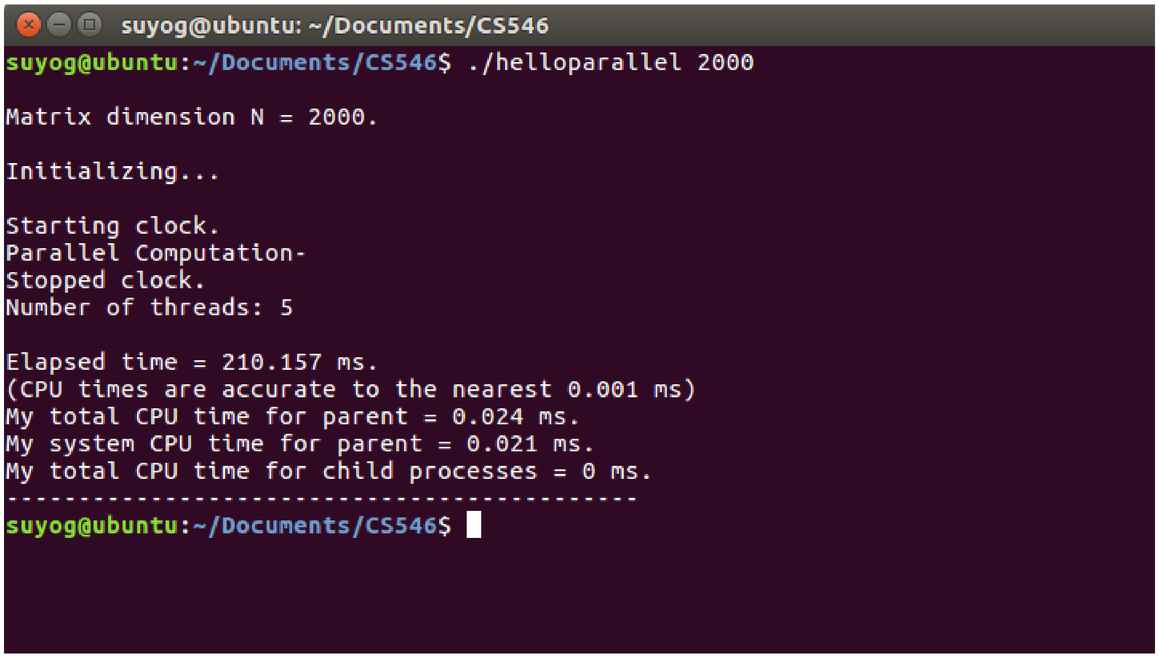
4. **Screenshots:**

Serial Gaussian Elimination



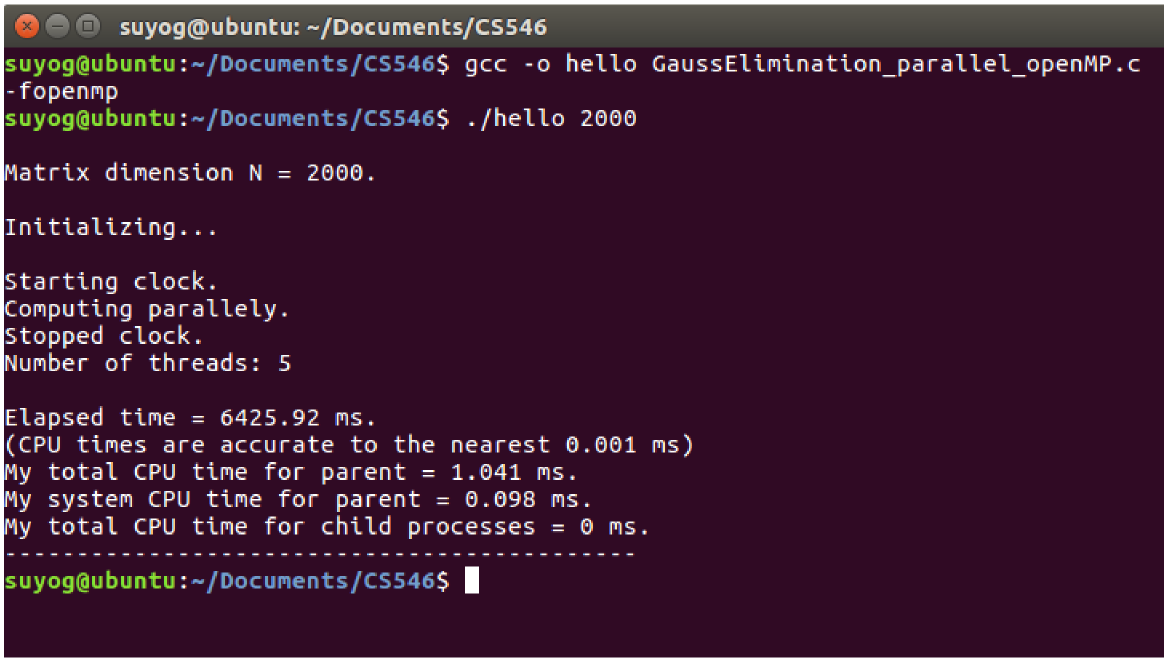
```
suyog@ubuntu: ~/Documents/CS546
suyog@ubuntu:~/Documents/CS546$ gcc -o helloserial GaussElimination_serial.c
suyog@ubuntu:~/Documents/CS546$ ./helloserial 2000

Matrix dimension N = 2000.

Initializing...

Starting clock.
Computing Serially.
Stopped clock.

Elapsed time = 10749.2 ms.
(CPU times are accurate to the nearest 0.001 ms)
My total CPU time for parent = 1.073 ms.
My system CPU time for parent = 0 ms.
My total CPU time for child processes = 0 ms.
--------------------------------------------
suyog@ubuntu:~/Documents/CS546$ 
```

GuassElimination_parallel_Pthread.c



```
suyog@ubuntu: ~/Documents/CS546
suyog@ubuntu:~/Documents/CS546$ ./helloparallel 2000

Matrix dimension N = 2000.

Initializing...

Starting clock.
Parallel Computation-
Stopped clock.
Number of threads: 5

Elapsed time = 210.157 ms.
(CPU times are accurate to the nearest 0.001 ms)
My total CPU time for parent = 0.024 ms.
My system CPU time for parent = 0.021 ms.
My total CPU time for child processes = 0 ms.
-------------------------------------------
suyog@ubuntu:~/Documents/CS546$
```

GuassElimination_parallel_openMP.c



```
suyog@ubuntu: ~/Documents/CS546
suyog@ubuntu:~/Documents/CS546$ gcc -o hello GaussElimination_parallel_openMP.c
-fopenmp
suyog@ubuntu:~/Documents/CS546$ ./hello 2000

Matrix dimension N = 2000.

Initializing...

Starting clock.
Computing parallely.
Stopped clock.
Number of threads: 5

Elapsed time = 6425.92 ms.
(CPU times are accurate to the nearest 0.001 ms)
My total CPU time for parent = 1.041 ms.
My system CPU time for parent = 0.098 ms.
My total CPU time for child processes = 0 ms.
-------------------------------------------
suyog@ubuntu:~/Documents/CS546$
```