|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | P | Serial execution time (microseconds)  P=1 | Parallel execution time (microseconds) | SpeedUP |
| 4 | 4 | 3 | 308 | 0.00974 |
| 8 | 4 | 25 | 447 | 0.055928 |
| 16 | 4 | 168 | 455 | 0.369231 |
| 32 | 4 | 1214 | 764 | 1.589005 |
| 64 | 4 | 3139 | 2486 | 1.262671 |
| 128 | 4 | 26337 | 16077 | 1.638179 |
| 256 | 4 | 205307 | 123942 | 1.656476 |
| 512 | 4 | 1654083 | 994117 | 1.663872 |
| 1024 | 4 | 13350955 | 7822639 | 1.706707 |
| 2048 | 4 | 131715625 | 62440432 | 2.109461 |
|  |  |  |  |  |

The Given NxN matrix is divided to 4 sub matrices if N > 2. In the Parallel Programming each sub matrix is handled by one thread for any given matrix (N>2). Totally there will be 4 threads spawned to compute the sub matrices multiplication.

Here Strassen’s algorithm’s is used for matrix multiplication to provide reduced the execution time, though this algorithm gives the time complexity of O(n^2.80), it has few drawbacks

1. For the matrix size which is not equal to 2^x (Where x=1,2,3,…n) we are appending zero’s to the given matrix in order to make the matrix size as 2^x (Where x=1,2,3,…n), which results in additional computation for the added extra zero values in the given matrix.
2. Because of the limited precision of computer arithmetic on non-integer values, larger errors accumulate in Strassen’s algorithm than in Naive Method

Serial Execution Logs:

-bash-4.2$ ./matrixMultiplicationSerial.out 4

3 microseconds occurred

-bash-4.2$ ./matrixMultiplicationSerial.out 8

25 microseconds occured

bash-4.2$ ./matrixMultiplicationSerial.out 16

168 microseconds occured

-bash-4.2$ ./matrixMultiplicationSerial.out 64

3139 microseconds occured

-bash-4.2$ ./matrixMultiplicationSerial.out 128

26337 microseconds occured

-bash-4.2$ ./matrixMultiplicationSerial.out 256

205307 microseconds occured

-bash-4.2$

-bash-4.2$ ./matrixMultiplicationSerial.out 512

1654083 microseconds occured

-bash-4.2$

-bash-4.2$

-bash-4.2$ ./matrixMultiplicationSerial.out 1024

13350955 microseconds occurred

Parallel Execution Logs:

-bash-4.2$ srun -n1 -c2 ./matrixMultiplicationParallel.out 8

447 microseconds occured

-bash-4.2$ srun -n1 -c2 ./matrixMultiplicationParallel.out 16

455 microseconds occured

-bash-4.2$ srun -n1 -c2 ./matrixMultiplicationParallel.out 32

764 microseconds occured

-bash-4.2$ srun -n1 -c2 ./matrixMultiplicationParallel.out 64

2486 microseconds occured

-bash-4.2$ srun -n1 -c2 ./matrixMultiplicationParallel.out 128

16077 microseconds occured

-bash-4.2$ srun -n1 -c2 ./matrixMultiplicationParallel.out 256

123942 microseconds occured

-bash-4.2$ srun -n1 -c2 ./matrixMultiplicationParallel.out 512

994117 microseconds occured

-bash-4.2$ srun -n1 -c2 ./matrixMultiplicationParallel.out 1024

7822639 microseconds occurred

-bash-4.2$ srun -n1 -c2 ./matrixMultiplicationParallel.out 2048

62440432 microseconds occured