

# **OPENMP ASSIGNMENT**

## **PARALLEL COMPUTING ASSIGNMENT:- 4**

**SUBMISSION: 5/11/2021**

**NAME: KHAN MD SHAHEDUL ISLAM  
STUDENT ID: 2018380130**

### **Matrix-Multiplication:**

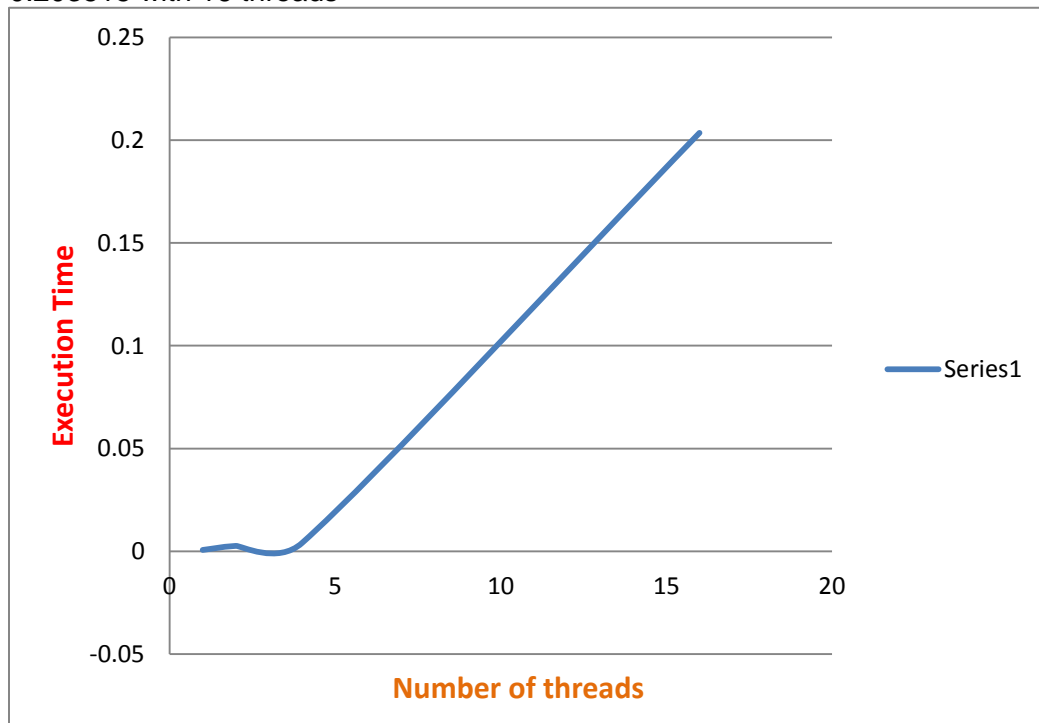
The following results were obtained when the number of threads were increased :

0.000708 with 1 thread

0.002680 with 2 threads

0.004106 with 4 threads

0.203515 with 16 threads



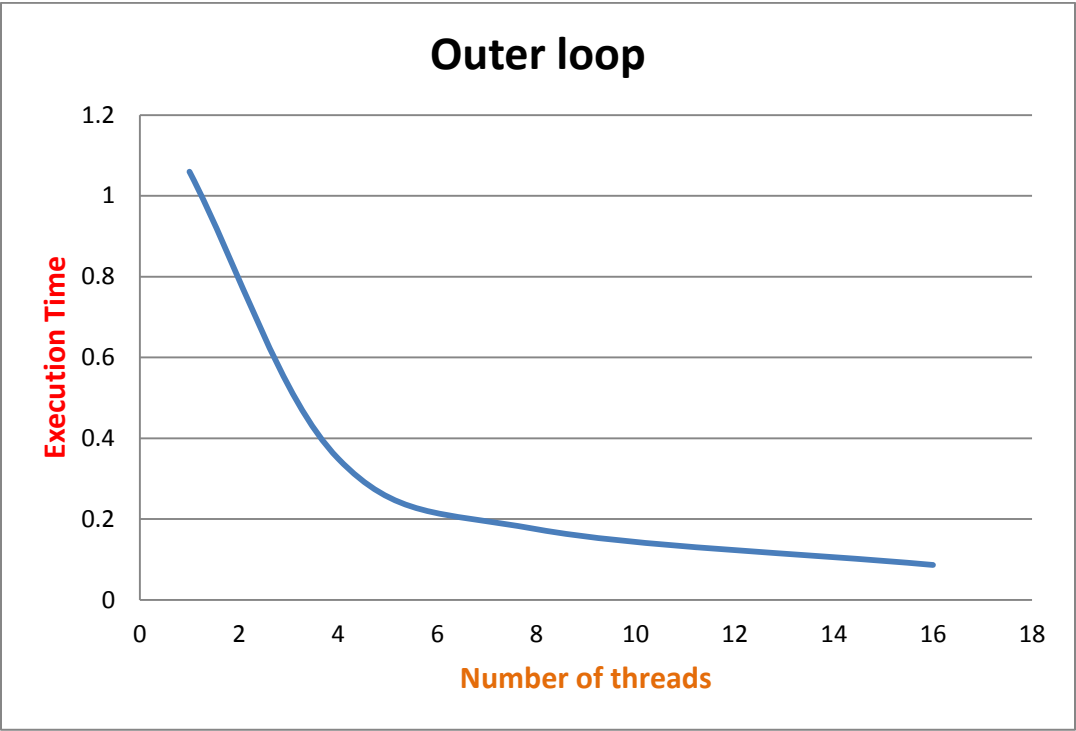
Hence increasing the number of threads is actually not improving the performance as most of the work is being done by only two threads!!!!!!!  
 The execution time was increasing linearly.....  
 Therefore more time is wasted on thread creation and join!!!!

#### MATRIX MULTIPLICATION:

The matrix multiplication was executed for sizes 50 , 500 and 2000

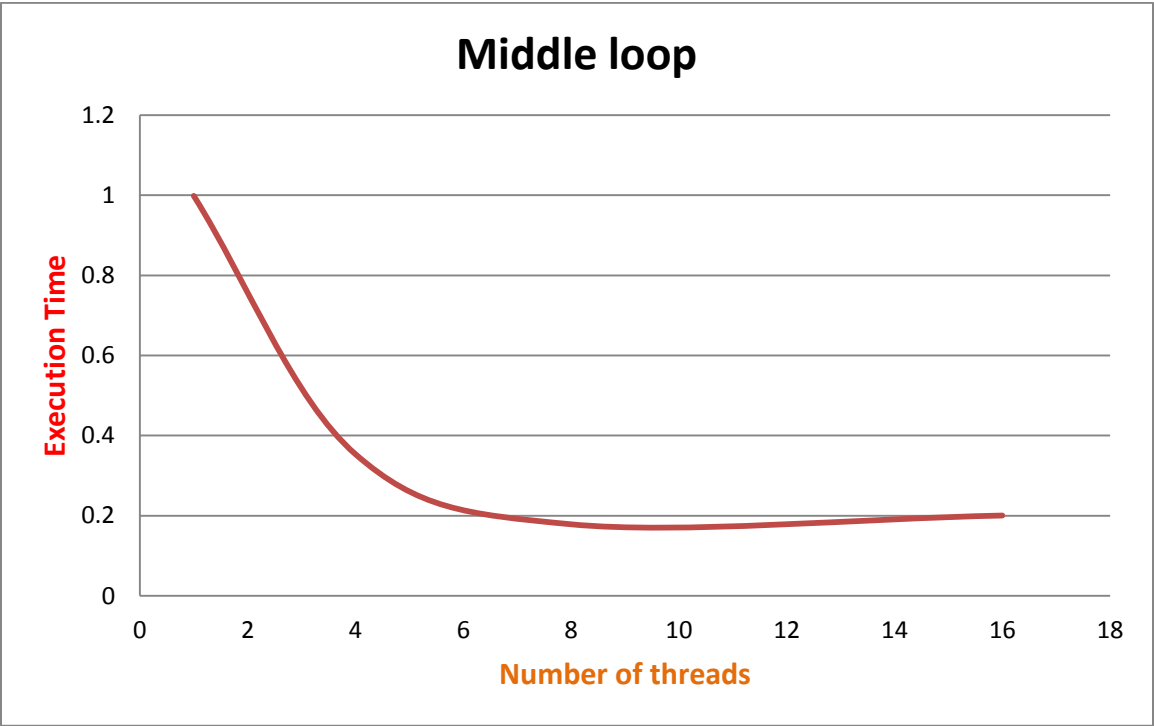
The results are shown below:

	Matrix Multiply				
	<b>Size 500</b>				
	<b>outer loop</b>				
	1 thread	4 threads	8 threads	16threads	
	1.068974	0.349947	0.172787	0.086703	
	0.976212	0.34875	0.175638	0.086534	
	1.052904	0.34999	0.175517	0.086196	
	1.151666	0.350008	0.175607	0.086478	
	1.129208	0.349537	0.175569	0.086477	
	1.047258	0.349744	0.172783	0.086456	
	0.986338	0.349826	0.175532	0.085895	
	1.019279	0.349855	0.175234	0.086738	
	1.019686	0.349771	0.175584	0.086596	
	1.146694	0.3488	0.17587	0.086543	
<b>Average</b>	1.0598219	0.3496228	0.1750121	0.0864616	
	THREADS	TIME			
	1	1.0598219			
	4	0.3496228			
	8	0.1750121			
	16	0.0864616			



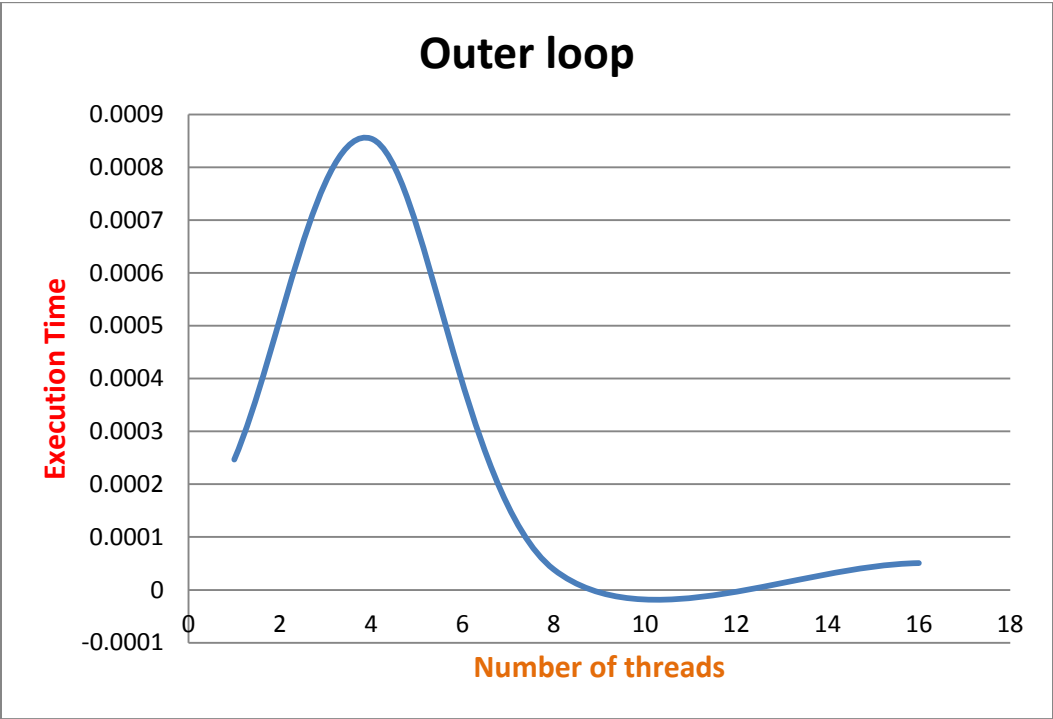
Size 500						
middle loop						
1 thread	4 threads	8 threads	16 threads			No Parallel
0.956809	0.35291	0.177552	0.22536			0.994878
0.989787	0.353135	0.178595	0.182458			1.102907
1.055734	0.353157	0.178441	0.214779			1.082362
1.066249	0.352813	0.178486	0.192603			1.03423
0.97851	0.352984	0.180637	0.180982			0.951382
0.922628	0.353102	0.177592	0.178629			0.918436
0.962398	0.353416	0.178684	0.178662			0.956998
1.033455	0.351551	0.178343	0.268589			0.978684
0.99474	0.353156	0.180255	0.180891			1.01107
1.02229	0.359511	0.177888	0.202638			1.000037
0.99826	0.3535735	0.1786473	0.2005591			1.0030984
	THREADS	TIME				
	1	0.99826				
	4	0.3535735				

	8	0.1786473				
	16	0.2005591				



	<b>Size 50</b>			
	<b>outer loop</b>			
	1 thread	4 threads	8 threads	16threads
	0.000252	0.000067	0.00004	0.000048
	0.000246	0.000067	0.000039	0.000055
	0.000246	0.000071	0.000041	0.000052
	0.000246	0.000065	0.00004	0.000053
	0.000246	0.000063	0.000037	0.000056
	0.000246	0.000069	0.000039	0.000051
	0.000247	0.000067	0.000038	0.000042
	0.000246	0.007935	0.000037	0.000046
	0.000245	0.000065	0.000037	0.000054
	0.000247	0.000072	0.000039	0.00005
<b>Average</b>	0.0002467	0.0008541	0.0000387	0.0000507
	THREADS	TIME		
	1	0.0002467		

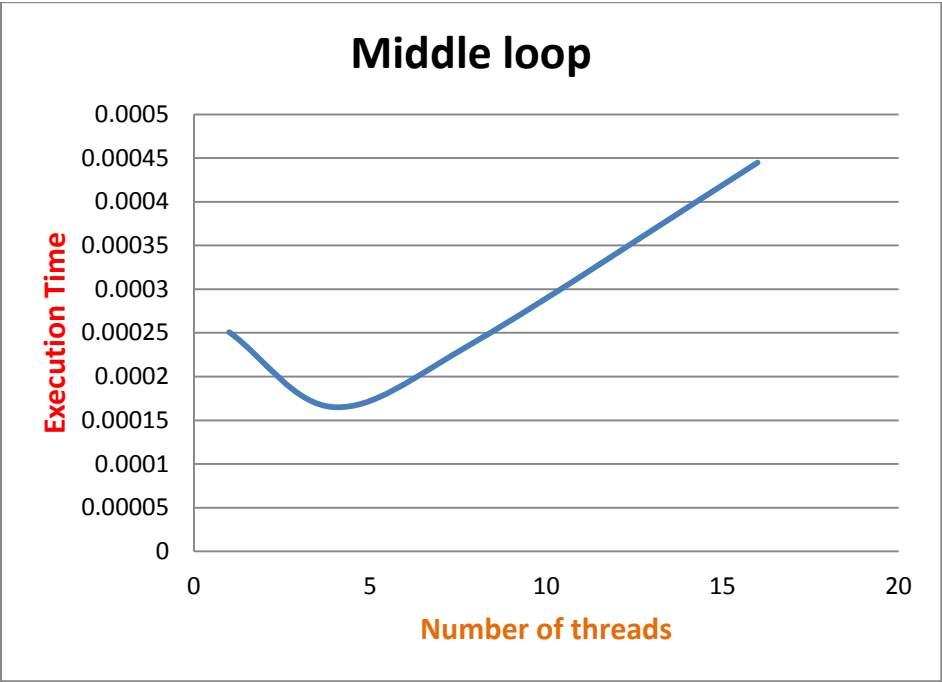
	4	0.0008541		
	8	0.0000387		
	16	0.0000507		



<b>Size 50</b>						
<b>middle loop</b>						
1 thread	4 threads	8 threads	16 threads			No parallel
0.000252	0.000176	0.000209	0.000431			0.000237
0.000251	0.000176	0.000234	0.000454			0.00025
0.000251	0.000116	0.000236	0.000442			0.000238
0.000251	0.000181	0.000263	0.000435			0.000238
0.000251	0.00017	0.000253	0.000432			0.000237
0.000251	0.000118	0.000222	0.000445			0.000237
0.00025	0.000187	0.00023	0.000466			0.000237
0.00025	0.000162	0.00025	0.000453			0.000237
0.00025	0.000186	0.000238	0.000436			0.000238
0.000251	0.000178	0.000268	0.000455			0.000237
0.0002508	0.000165	0.0002403	0.0004449			0.0002386

	THREADS	TIME				
	1	0.0002508				

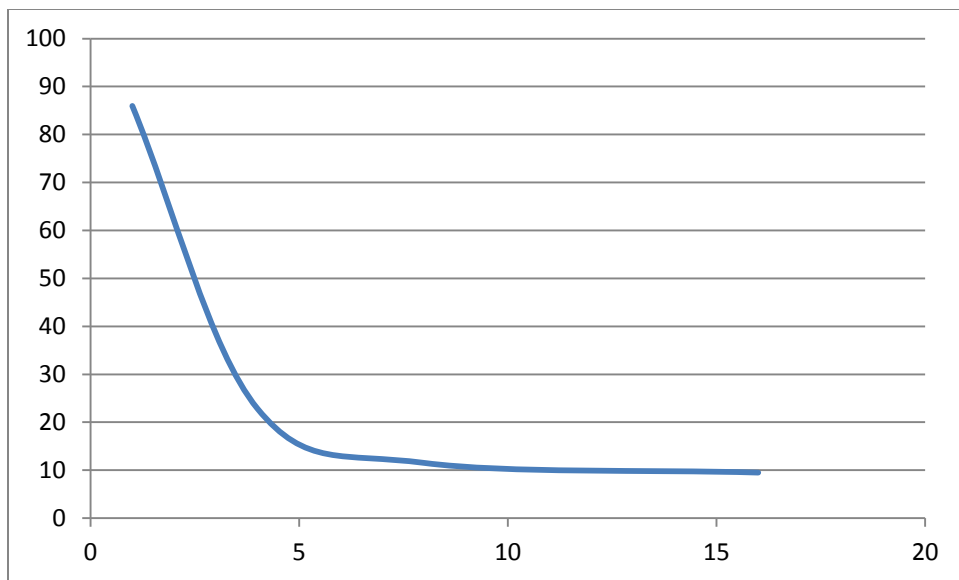
	4	0.000165				
	8	0.0002403				
	16	0.0004449				



**Size 2000**

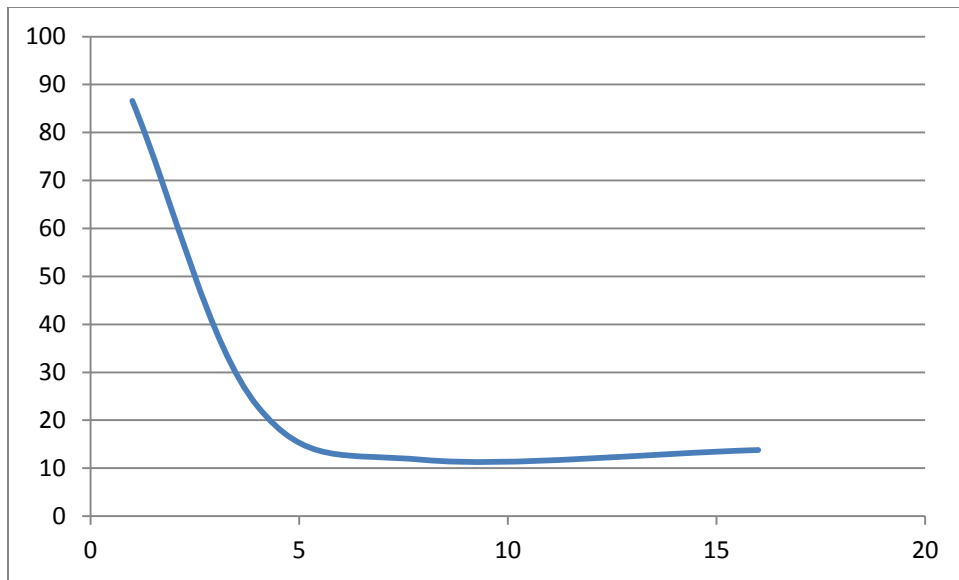
**Outer loop**

THREADS	TIME
1	85.95526
4	22.83168
8	11.51947
16	9.506124



### Middle loop

THREADS	TIME
1	86.59042
4	22.92648
8	11.73231
16	13.79196



When we increase the number of threads the performance has improved drastically.

The graphs when middle and outer loop was parallelized were similar.

In the case of size 50 matrix we could not observe much performance since thread creation overhead is more in comparison to the size of the matrix.

But in the case of 2000 matrix the parallelization was evident since it gave better performance as the number of threads were increased. The granularity in this case was more compared to size 50 matrix and hence increasing the number of threads led to more parallelization and hence we observed better performance!!!!!!!!!!