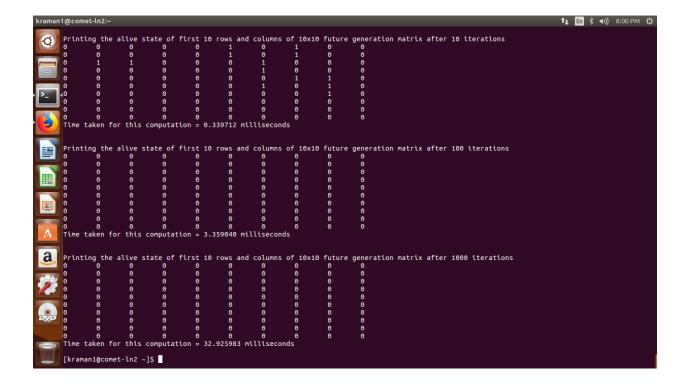
RESULT REPORT

I have used different matrix sizes to evaluate the performance of my parallel algorithm. The various sizes of the matrices are 10x10, 50x50, 100x100, 500x500 and 1000x1000. The following table shows the runtime of various sized matrices for 10, 100 and 1000 iterations.

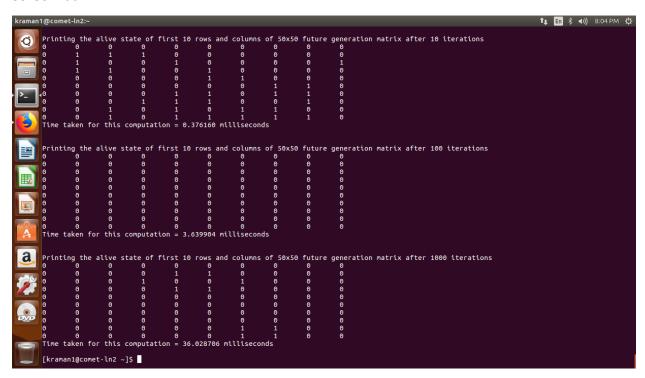
MATRIX SIZE	TIME TAKEN FOR 10 ITERATIONS	TIME TAKEN FOR 100 ITERATIONS	TIME TAKEN FOR 1000 ITERATIONS
	(in milliseconds)	(in milliseconds)	(in milliseconds)
10x10	0.339712	3.35904	32.925983
50x50	0.37616	3.639904	36.028706
100x100	0.486688	4.675232	46.450401
500x500	3.776256	39.59811	386.737091
1000x1000	11.04448	114.800034	1021.810364

Following screenshots show the output for each matrix size while using parallel computation.

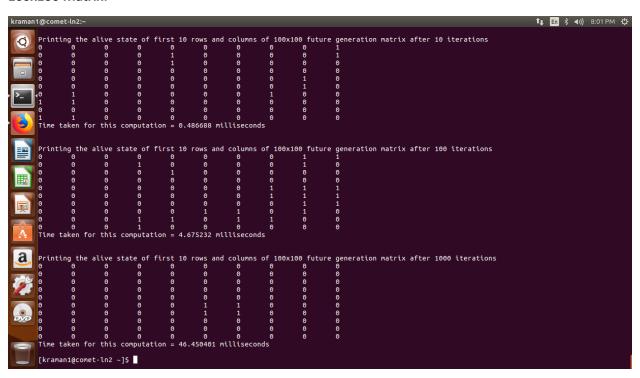
10x10 Matrix:



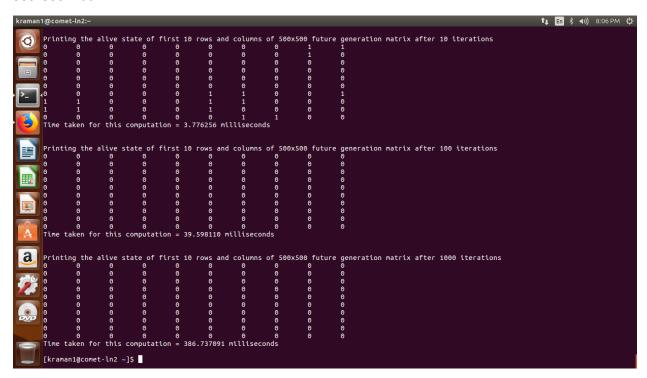
50x50 Matrix:



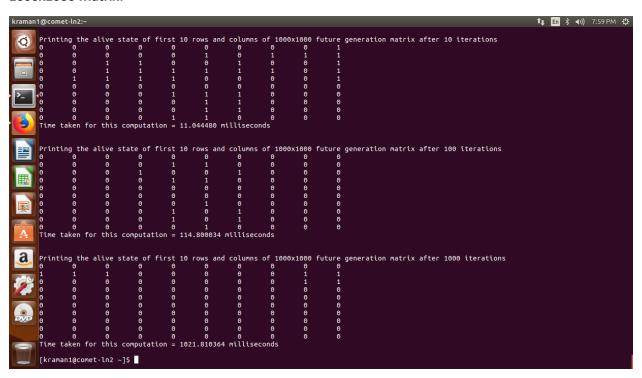
100x100 Matrix:



500x500 Matrix:



1000x1000 Matrix:



Now let us examine the need for CUDA by comparing the performance of Serial algorithm with Parallel algorithm using CUDA.

Following table shows the computation times for both serial and parallel computation.

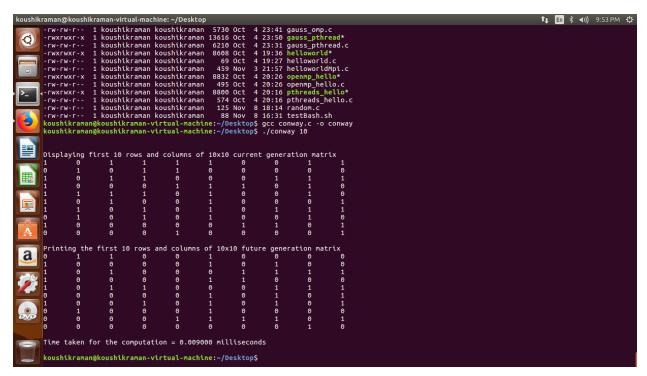
MATRIX SIZE	SERIAL COMPUTATION TIME FOR 1	PARALLEL COMPUTATION USING CUDA
	ITERATION	FOR I ITERATION
	(in milliseconds)	(in milliseconds)
10x10	0.009	0.05824
50x50	0.18	0.04144
100x100	0.654	0.04336
500x500	16.349	0.141984
1000x1000	68.466	0.525952

The above table clearly shows that when the matrix size is considerably small, serial computation is effective. This is shown when matrix size is 10x10.

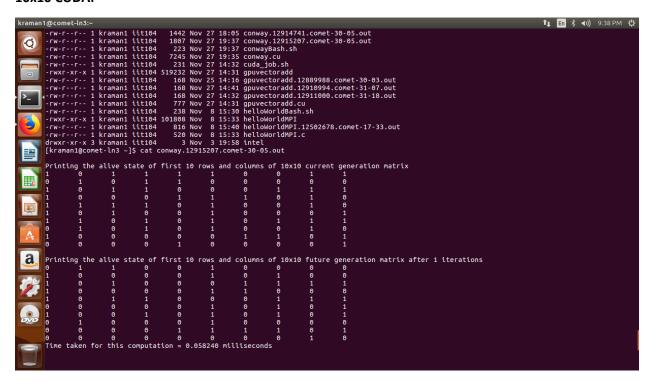
But when the matrix size increases, the serial computation time keeps on increasing at an alarming rate. However, the parallel computation while using CUDA shows very short computation times as compared to serial computation.

Following are the screenshots for the above table.

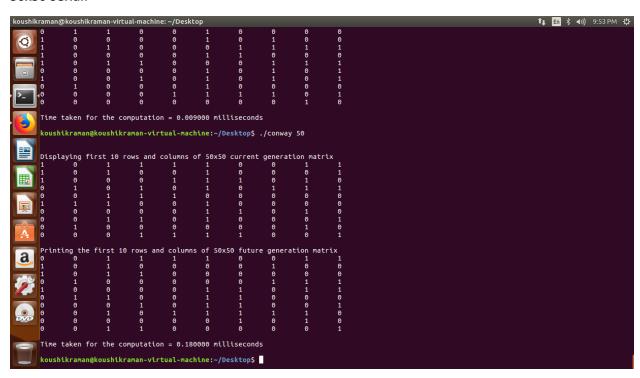
10x10 Serial:



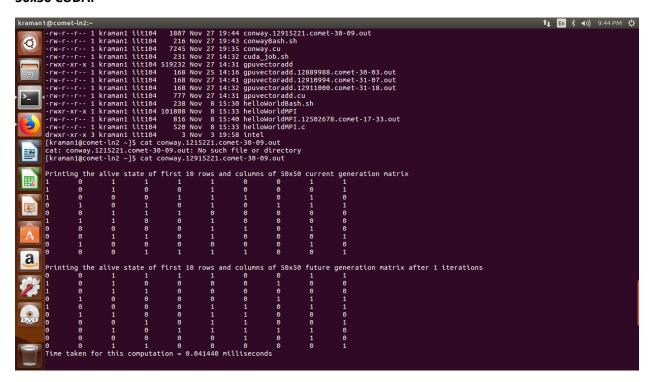
10x10 CUDA:



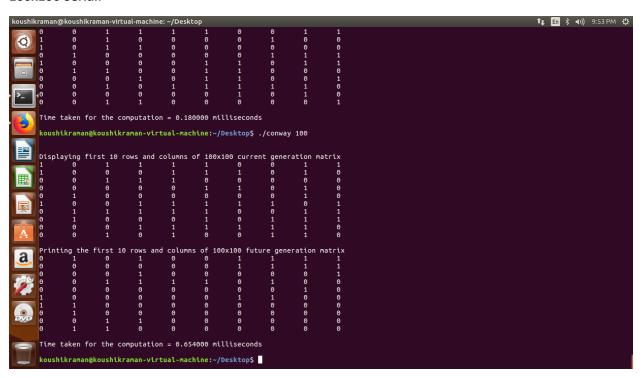
50x50 Serial:



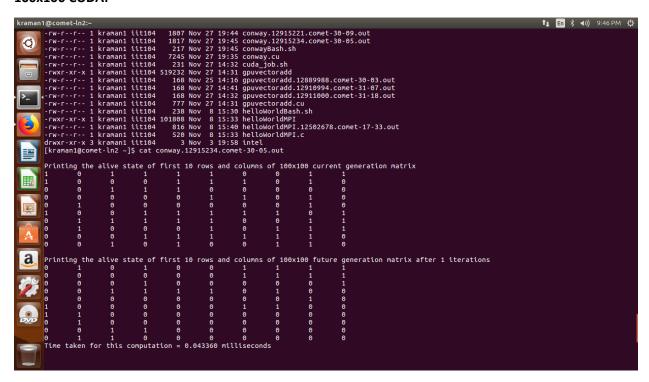
50x50 CUDA:



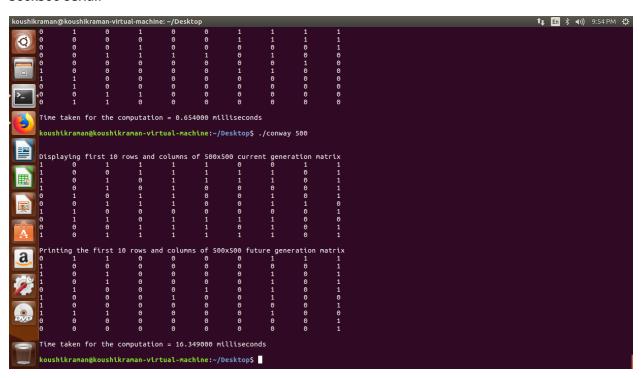
100x100 Serial:



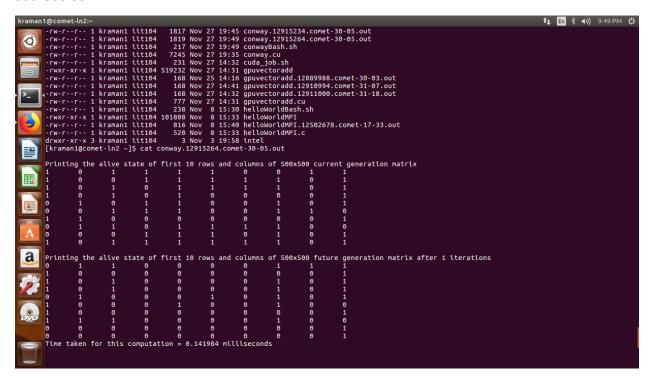
100x100 CUDA:



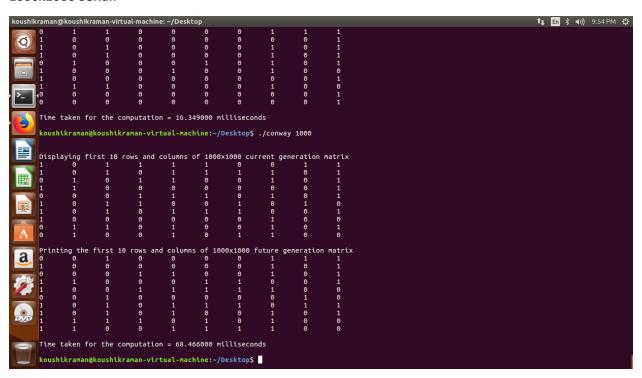
500x500 Serial:



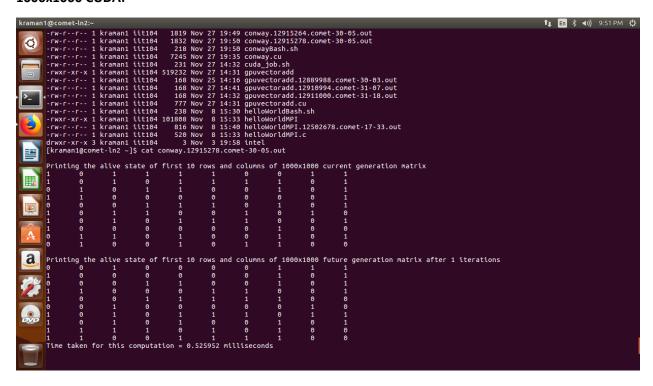
500x500 CUDA:



1000x1000 Serial:



1000x1000 CUDA:



CONCLUSION:

These screenshots and the table clearly shows that CUDA is the better option than serial computation when the input matrix size is large.

Thus, I have learnt from this assignment that CUDA is one of the most efficient ways to implement parallel computation.