

# ECEN 5593: ADVANCED COMPUTER ARCHITECTURE

## FINAL PROJECT

### Matrix Multiplication Research and Comparison

SANJANA KALYANAPPAGOL  
saka2821@colorado.edu

12<sup>th</sup> May, 2017

SPRING 2017

## ABSTRACT:

Utilizing graphics hardware for general purpose numerical computations has become a topic of considerable interest. The implementation of streaming algorithms, typified by highly parallel computations with little reuse of input data, has been widely explored on GPUs. The matrix multiplication is the key operation for many computationally intensive algorithms. It is a kernel operation used in many transform, image and discrete signal processing application as well as robotic applications. Matrix mathematics applies to several branches of science, as well as different mathematical disciplines. It is also used in the field of Computer Graphics. Nowadays, we can observe the results of matrix mathematics in every computer-generated image that has a reflection, or distortion effects such as light passing through rippling water, etc. Before computer graphics, the science of optics used matrix mathematics to account for reflection and for refraction and It also helps calculate the electrical properties of a circuit, with voltage, amperage and resistance. Because of its trending applications, new algorithms and new techniques are being developed on configurable devices too. Its regular data access pattern and highly parallel computational requirements suggest matrix-matrix multiplication as an obvious candidate for efficient evaluation on GPUs but, surprisingly near optimal GPU implementations are pronouncedly less efficient than current cache-aware CPU approaches

The aim of the project is to develop Matrix Multiplication that is implemented using NVIDIA's CUDA (Compute Unified Device Architecture). This project consists of the concepts of GPU, CUDA and C programming to implement Matrix Multiplication.

## INTRODUCTION:

A matrix is represented as a rectangular array of numbers which has one of the most useful and fundamental implementation in the field of mathematics and scientific computation. For matrix multiplication, we have two matrices that have 'i' rows and 'j' columns. We use the SIMT code on a CUDA enabled device. SIMT refers to 'single instruction, multiple thread', a model where many threads running in parallel to execute same instruction at the same time. The CUDA kernels are used for parallel execution and scale from a manycore GPU to massive distributed system consisting of hundreds of processors.

The threads in CUDA consists of their own program counters and registers. They have a 'global memory' and a 'shared memory' that is more limited in size. Within the same block the thread shares the instructions and execute them in parallel. They diverge to execute serially and once that execution is successfully implemented they converge to start parallel execution. The threads are grouped into Blocks. Each block can hold up to 512-1024 threads. Hence multiple blocks are used to compute the result of the matrix multiplication. The blocks can be 1 - Dimensional, 2 - Dimensional and 3 - Dimensional. To make it all system compactible we use the size of  $16 \times 32$ .

The thread hierarchy is explained below in the figures. In the CUDA memory hierarchy each thread has an access to private local memory for storing large data. All threads have access to shared memory which can be accessed by the threads of that block, whereas the global memory is accessible to all the threads. A kernel can be launched by specifying the grid dimensions, block

dimensions and the kernel function to run on the device. These functions act as an entry for GPU computations. At the time of kernel launch on the GPU, a grid of thread blocks is created and the blocks are queued to run on the GPU. The CUDA programming model requires that these blocks can compute in any order, that is, the programmer may make no assumptions about the order in which the GPU schedules and runs the blocks of threads. The program must run correctly regardless of the order in which the blocks are scheduled.

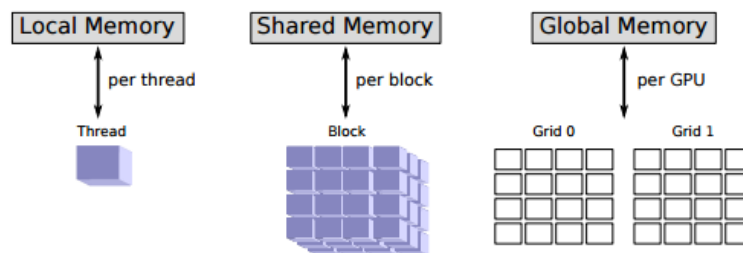


Fig: Memory hierarchy and its relation between thread hierarchy

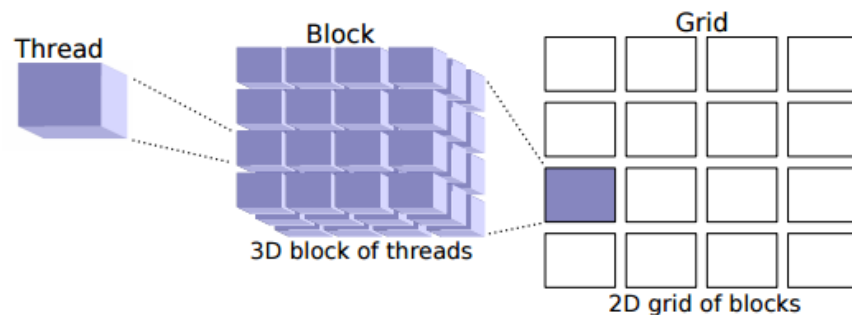


Fig: CUDA thread Hierarchy

## METHODOLOGY:

The kernel function is declared as `__global__` in the code and a global function on the device and passing parameters is used. The variables that are used can hold triple unsigned integers. An example syntax where the variables to which the threads have access is:

***GlobalFunction<<<variables that hold 3 integers>>> (parameters)***

### Implementation of Matrix Multiplication:

The matrix multiplication will consist of the following:

1. `__global__` keyword for declaring that it is an entry point function for running code
2. Declarations to reserve a register to hold the value where the product of the row and column entries.
3. Declaring the row and column of the matrix and its boundaries.

4. A condition check that terminated if the row and column is outside the boundary of the product matrix.
5. Loop to iterate over the values of the row and column of that matrices that must be multiplied to obtain the product matrix.

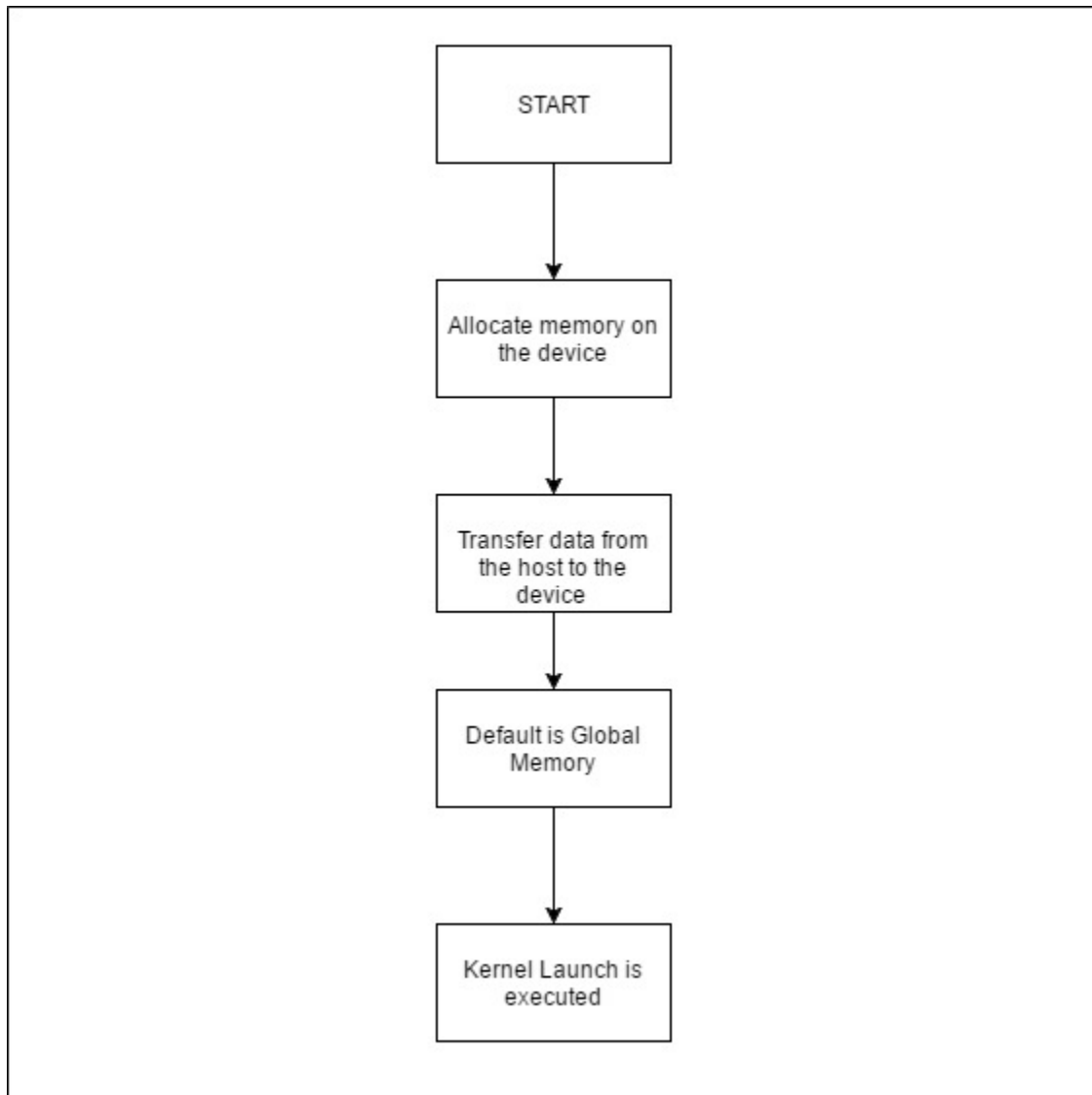


Fig 1 : Initial Start Diagram

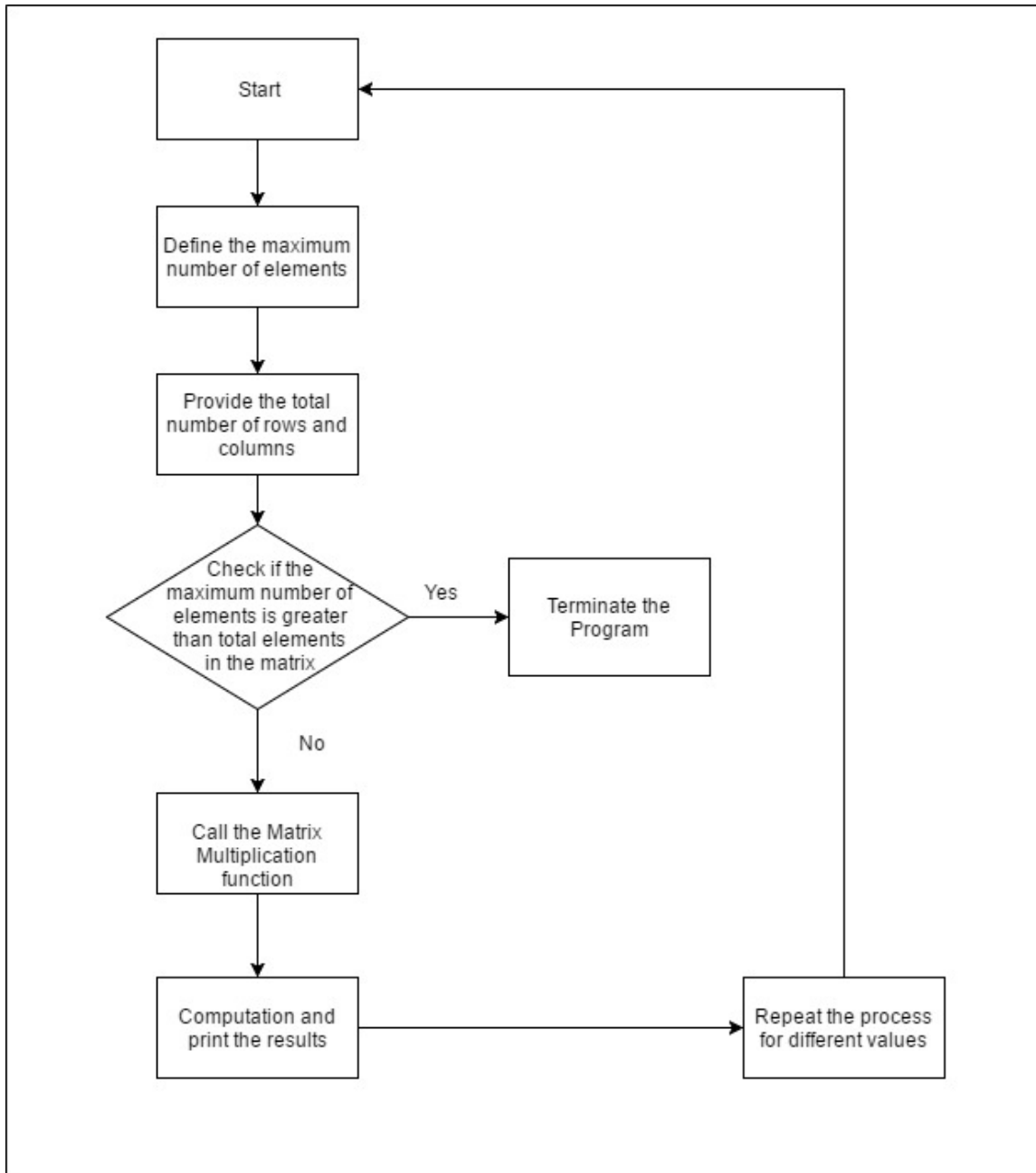


Fig 2 : Code flow diagram

The two matrices say A and B are stored in the global memory as a 1 - D array with the first row followed by the second row, and so on. Thus to find the index in this linear array of the (i, j) - entry of matrix A, for example, we compute  $(i \times \text{width of A})$  to find the starting index of the ith row, and then add j to go to the jth entry in that row. Finally, the last line of the kernel copies this product into the appropriate element of the product matrix C, in the device's global memory.

In this project, I have computed the CPU and GPU timing for different sizes of the matrix and compared them with one another. The variations have also been indicated.

## RESULTS AND ANALYSIS:

The computations for various matrix sizes 10\*10 to 300\*300 matrix were implemented and analyzed. The CPU and GPU times are presented in the table below supported by the screenshots.

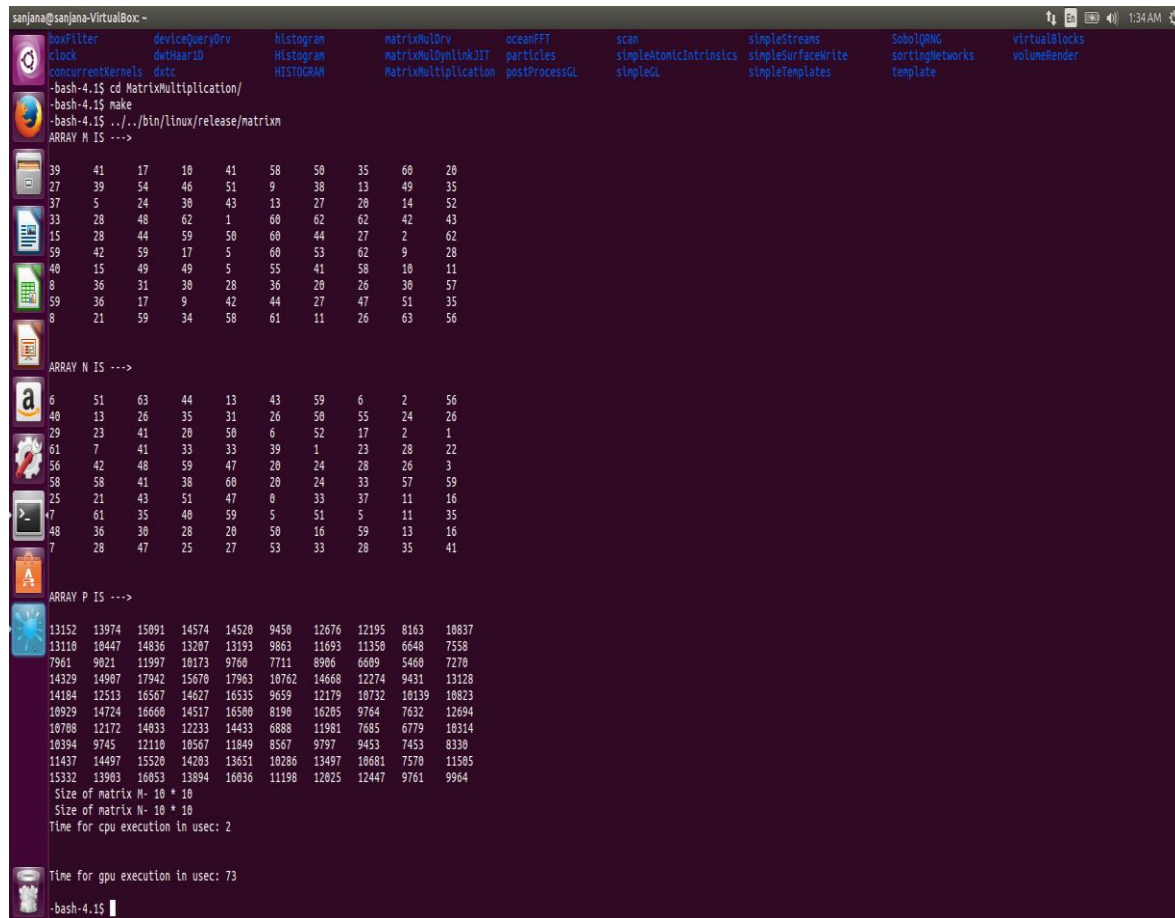


Fig 3 : 10\*10 Matrix Multiplication

```

sanjana@sanjana-VirtualBox: ~
ARRAY N IS --->
6 51 63 44 13 43 59 6 2 56 40 13 26 35 31 26 50 55 24 26
29 23 41 20 50 6 52 17 2 1 61 7 41 33 33 39 1 23 28 22
56 42 48 59 47 20 24 28 26 3 58 58 41 38 60 20 24 33 57 59
25 21 43 51 47 0 33 37 11 16 7 61 35 40 59 5 51 5 11 35
48 36 30 28 20 50 16 59 13 16 7 28 47 25 27 53 33 28 35 41
36 19 10 40 50 32 13 52 31 55 11 5 6 43 26 34 48 51 4 19
20 47 48 30 50 9 2 41 40 50 41 8 10 21 24 4 27 14 23 27
2 6 51 13 4 45 20 7 50 44 4 54 35 44 50 4 44 57 58 3
5 44 39 55 16 35 19 52 30 58 40 5 1 52 4 58 41 35 26 41
62 62 37 8 19 20 49 16 23 46 60 37 15 53 34 19 42 50 5 20
53 20 2 24 40 53 48 10 34 41 30 37 14 25 17 44 21 44 36 49
12 49 0 59 60 38 33 35 34 3 41 58 47 22 44 31 45 52 13 15
20 35 1 36 3 24 22 44 57 24 10 29 9 37 1 31 42 10 9 16
22 58 58 50 40 25 2 42 8 33 3 30 3 26 23 25 18 25 57 0
22 20 17 36 13 35 6 63 42 21 3 46 2 33 19 30 60 1 49 38
43 15 11 58 39 54 62 63 43 61 3 39 13 12 14 61 60 42 27 60
52 45 22 51 13 0 11 15 16 23 14 24 33 17 39 62 63 14 24 22
50 54 62 22 33 45 2 57 9 31 23 12 9 43 27 7 24 11 2 36
15 18 24 53 21 17 51 2 55 30 47 32 34 29 13 8 19 46 38 12
57 40 59 40 24 6 43 14 18 50 44 59 20 43 14 54 22 51 8 4

ARRAY P IS --->
20147 25732 23084 28704 20754 19952 18676 24569 21722 24397 19773 20763 14893 24702 17999 23377 26661 23649 20079 17649
25283 27506 24324 27914 20164 19892 20345 23128 18474 24428 17845 22893 15134 22506 18859 22480 26254 22701 17427 18666
29651 27902 23409 29094 25692 22124 22298 27344 21465 25482 20146 25722 17779 25754 23428 24191 29062 24587 17228 23321
18071 21067 22811 25902 20034 16581 17666 20254 18571 21069 16417 22778 14310 21777 19576 17125 24204 21809 16946 15753
21746 25627 25447 29013 19766 22833 23347 26082 23185 27822 20145 23442 15251 26258 17829 24499 28798 26389 20663 19536
25425 30237 28503 34199 25070 25047 22539 27105 23779 29245 24028 25744 18486 28966 23696 25878 31099 28745 24440 23331
20832 24793 25304 25092 20913 15422 17418 20874 14587 21145 17963 18771 13649 20334 18509 18743 22073 18826 16204 16489
16899 19414 17467 23029 18163 15665 14643 21217 18939 18694 14687 19209 14304 20711 18421 17780 25200 18425 15732 15060
20715 26556 25728 30071 23381 21892 21552 26993 18217 24244 19200 23380 15105 24944 20638 22891 28131 23612 19913 20566
17976 21418 20260 26212 19660 16322 17058 21821 18730 22966 16444 20013 12233 21440 16782 19887 25346 20204 14362 16872
11804 16804 16051 21956 16992 14013 14652 17774 14077 14544 14565 16125 12371 15669 14605 15624 18216 16203 14678 14914
26732 28673 23904 31092 23943 18254 22246 23813 21204 22681 23592 26586 18057 26805 22279 24493 27561 24326 19381 20703
20281 19978 16860 25520 21113 16775 19249 20417 19709 18299 18356 21042 15450 18752 16482 20727 21061 20856 17015 17260
18047 19072 20209 21231 16364 14603 14081 19363 15786 17191 12909 18203 13791 17004 17286 16764 20573 15797 17321 15747
18078 19021 18289 22107 16660 13668 15243 19723 12971 15147 13996 19534 12878 17695 16678 16447 22079 14355 12264 15909
24781 26109 22540 20157 19747 22117 20610 24048 21081 24310 17434 25209 15785 22549 20008 24400 29038 22915 21668 21313
17289 16458 16941 19632 17345 14104 13657 18193 14877 15037 14781 17435 11756 19011 17244 14249 19797 15331 14892 16146
19098 21821 20294 22398 19117 16621 14043 19566 15307 18633 18091 16531 11162 19020 15609 16419 18866 18207 15075 16936
17208 22990 21622 22758 16776 16083 15362 18735 10069 20271 18093 17729 12260 20631 15623 17739 21021 20077 16325 14378
20954 23393 22942 25965 20580 21179 20976 22401 16257 20350 17549 24601 16278 21506 20665 20058 25284 22627 19661 19660

Size of matrix N- 20 * 20
Size of matrix P- 20 * 20
Time for cpu execution in usec: 17

Time for gpu execution in usec: 79

-bash-4.1$

```

Fig 4: 20\*20 Matrix Multiplication

sanjana@sanjana-VirtualBox ~

1:37 AM

27462	33622	25444	33310	28691	36431	38925	30932	33480	31241	28668	36255	34438	32875	29555	29599	22458	31728	27520	27407	31890	31855	32854	36155	38024	2680
3	33752	34830	35195	33062																					
239533	31328	28900	32073	29678	31852	33584	28103	31960	31840	29198	34253	32601	34459	28293	28763	25831	31391	24394	28734	32321	30717	27574	32476	27756	2853
7	30761	31700	31945	32334																					
23391	30985	23404	26375	23180	27465	29008	26400	26954	25843	21368	29131	24399	27526	23439	22537	20521	24477	18223	23858	26790	23926	19555	25241	22929	2537
0	26708	26383	25475	25405																					
30354	38904	29091	36593	30232	38193	35434	30789	33128	39160	27131	33178	28784	33183	31405	32751	28792	34101	26127	32806	38004	33380	33819	34376	29717	3243
9	39005	31308	32742	32213																					
25152	29735	23723	28237	24727	29163	27244	22015	29850	32303	24934	27009	23037	27118	23193	26251	23105	25888	21448	24087	26105	26759	28936	25549	23114	2398
3	28123	25934	23661	27046																					
25240	35036	22923	29165	26410	35862	29476	26396	34392	33555	26394	34214	27562	33616	27006	29023	25521	29994	24312	25414	28755	33714	31290	30852	29656	2756
6	30832	29326	31032	30258																					
23654	31984	25387	29612	25970	30514	28247	27108	33727	28820	25259	30401	25229	33088	24177	24715	21768	28104	22797	23380	27847	29570	24620	28396	24682	2570
1	27653	29039	29189	28213																					
27078	35268	26468	33314	29273	34323	30063	27956	35695	30520	27257	32559	29721	33563	26727	30885	21987	33185	23937	28079	28959	31390	28349	34612	28819	2957
2	36181	30893	29019	33908																					
24041	34928	26644	29445	29036	28896	31283	25957	31778	29713	26997	30808	30097	30818	25256	27335	21723	29462	24230	26772	27641	27250	26677	30443	27027	3012
2	32403	29031	29121	30875																					
30507	38075	27246	36584	27294	38616	35146	32395	33795	33391	30264	36023	31103	36069	30905	29474	26349	30743	27297	27766	33582	34009	33619	34257	30035	3023
6	33132	36034	34692	32658																					
28176	34568	26622	28107	29591	31702	30210	30667	30961	32512	25292	34364	26281	36045	29039	27409	22192	28517	22527	27284	27781	31323	26897	33733	28619	3239
1	33778	35469	31250	30199																					
25673	29474	23291	25248	23900	27654	22570	21212	22518	26914	23447	28326	24118	28640	27060	24731	20634	26351	28215	24342	25232	24886	21089	25978	25328	2497
7	29951	29729	24799	23354																					
29409	39544	31330	32558	32554	35891	36118	30853	35593	37397	30903	39156	33710	35819	32091	30734	20216	34284	27450	29759	33720	33356	30801	34553	31441	3266
5	36276	34442	33727	34349																					
28941	34736	27317	33530	33473	35238	30884	30836	32739	36360	24546	35685	30941	33258	30670	28569	24599	26925	26010	28384	31148	34746	36496	32915	33069	3060
2	33899	37287	35534	33431																					
24057	35530	26351	31701	31437	35048	34820	28281	33329	32768	24261	34863	29071	33324	28222	30574	21495	33311	25181	29944	27517	30075	27572	35187	27778	2902
46	37450	34423	28242	30074																					
27283	37988	28067	34211	29570	35799	34669	28958	34957	34776	27592	36735	29338	35677	32034	29123	25224	34468	26766	28415	32944	33216	31669	34216	29837	3135
2	36159	33356	29860	30840																					
27661	33244	23968	25273	25500	31550	27305	26231	26501	30153	23564	29828	26921	31728	25411	25844	20235	25648	28974	24951	24035	28849	27891	26906	28357	2891
8	38048	28506	33115	27650																					
23577	38889	28397	34686	33116	36014	36731	33830	38158	34696	27971	36627	31820	37068	29304	31734	27400	34774	29336	31564	32002	35636	31353	36379	34452	3265
3	34497	36641	36803	35945																					
22553	31061	28815	28865	27141	26934	23761	30912	29473	28458	22612	31448	22770	30968	24857	24591	23671	25058	22363	21381	28575	32269	28091	30484	29289	2507
5	26076	31584	30300	28793																					
27575	34769	28083	30320	31299	37277	33831	30110	33777	36147	30482	34823	27990	34994	30661	32703	26224	36459	28245	28196	30131	35103	26422	32303	32723	2919
7	35112	31394	32203	34424																					
22690	29222	22136	25763	26181	31937	28186	22626	29922	33222	24180	30639	22002	29213	26939	26906	22152	30144	22121	24153	26684	30503	24378	30733	26867	2415
8	30073	27873	23678	26137																					
28515	36021	29527	34050	32097	37515	34232	28993	37428	40264	27482	34467	28698	33736	29091	31838	27824	32755	26070	31158	31189	35421	32873	34669	31268	3047
8	33728	34506	34141	33523																					
29616	36448	26573	36475	32252	33143	35551	37033	37114	30513	26018	37256	31713	34298	27289	32528	28584	32260	27321	31958	32453	33599	26425	36661	28172	3081
5	34005	37194	36115	35866																					

Size of matrix M- 30 \* 30  
Size of matrix N- 30 \* 30  
Time for cpu execution in usec: 56  
Time for gpu execution in usec: 99  
-bash-4.1\$

Fig 5: 30\*30 Matrix Multiplication



sanjana@sanjana-VirtualBox: ~

37738	34478	34418	38875	37453	33241	35376	34518	37634	40759	34453	38423	29961	39223	35888	38131	40114	37870	41530	31580	34136	35070	36765	37936	37722	4194
8	37322	36798	32737	39535	33407	42685	39377	34814	41288	36841	36527	42366	36477	33253											
44508	42284	37837	45274	45160	39151	50120	41901	47176	48857	38073	38513	35447	45198	44046	47346	45708	45361	47538	38821	43335	43111	40269	44378	46624	5239
3	47315	46529	44517	50262	42805	53263	48517	41692	47944	44604	47444	47038	41602	40189											
37567	35417	33849	40236	38599	32124	38864	39756	40906	46720	37722	33179	32149	36009	41599	48441	43109	39804	45348	32880	42430	38776	42472	39187	37387	5123
3	41833	41126	38475	44774	35568	44937	42267	35217	45833	42579	40283	43425	39541	40381											
33768	34635	30260	35446	35028	28966	35792	34989	34338	34197	33790	36379	25916	33958	34672	36911	33715	32535	41723	32241	34309	32985	32992	37847	32614	4478
5	35056	37602	31191	38354	33291	39182	34146	33104	32066	38626	35759	35874	32936	29951											
40656	41270	38228	43499	47088	38863	43772	39824	38202	47434	37898	38739	34636	44136	43872	44976	44602	44827	42580	41263	39782	40319	40520	40974	42113	4862
0	46732	43800	39781	48369	41185	48858	46013	35591	43658	41863	42234	45250	42415	37678											
40568	38269	43387	43712	43689	40136	43692	34441	42062	40860	41107	34049	32899	43778	38447	44678	42927	39316	46570	37393	38839	38358	42528	41381	41837	5099
9	36980	41886	38533	48149	37499	46236	40680	43028	45910	37634	42481	48827	39578	34717											
40754	41264	38882	45547	40975	37775	38182	40630	40539	41939	34724	38547	30884	40487	41637	46682	44174	37691	47860	37723	39854	37859	39886	45974	44277	5297
2	42443	47416	37783	46549	36755	43973	41339	38182	43513	43757	40587	44810	43236	36236											
38987	32574	33471	33246	39688	32669	36323	38286	33380	39662	33791	30710	32031	33155	42247	38976	37565	42598	43591	32486	36017	34553	32637	36966	36575	4431
0	38511	37238	35538	39917	33880	41124	39349	36360	40416	39025	36620	37611	34973	34961											
34372	36487	35064	37414	39958	36929	38323	43086	43555	42423	37364	34010	31166	36712	38218	43308	39116	38351	46026	33937	39619	36886	36318	39631	38205	4917
8	40847	39321	37731	44738	38121	43990	36691	37541	39615	40763	43453	43725	35373	35382											
42568	39220	39750	42829	40336	36580	36264	35839	37946	46641	37438	37553	35164	43050	42210	47159	45228	41077	46651	36945	37590	42175	43107	42293	44447	5084
5	39985	45250	38543	45052	40607	45731	45280	36323	46851	41519	38829	45829	43847	38245											
40339	38797	39350	44781	43758	40975	41417	37926	41168	47579	35460	38836	32496	41267	39396	44240	43812	40393	47569	48215	40574	37190	37896	43546	42605	4784
9	44951	46903	39316	51344	39412	49685	39726	39998	39224	44706	43582	47334	39795	37165											
36806	36941	32995	36241	39394	33258	33553	35932	30883	45925	38075	32580	30871	36219	35764	39479	38977	37695	41142	33065	31898	34422	38217	36776	36780	4535
0	40356	37231	33138	38293	37498	40963	38030	31505	39492	38765	35777	39400	39078	38229											
33738	32029	29591	35819	32719	29594	31982	28514	37188	34342	29208	38765	28230	32777	30378	35424	33714	38889	38540	28377	33897	32696	31066	34287	35353	3980
2	29548	36386	29484	37031	38676	36898	29060	32739	33327	33834	34397	37034	26156	28459											
36668	36686	32551	41431	36079	29789	34804	35083	30758	38496	32939	34089	27751	34351	38786	37831	37856	35655	40174	33029	33763	34192	38855	39241	36971	4194
7	38959	40700	35551	40844	33010	42773	40822	31343	42139	37495	38866	39673	37689	34643											
43444	36992	37665	46127	44613	36858	41274	41947	43985	46724	40105	46027	34764	43448	37974	47991	41993	43798	51782	38471	42646	44496	40472	46072	42834	4880
9	47349	44146	42459	49821	44661	51250	45577	39043	43474	46083	43522	50695	42801	39825											
40155	43391	40852	44953	44913	38897	38444	43051	43215	49355	41638	36842	37201	44920	42025	46924	47089	44429	45504	37569	41041	42613	43359	45647	45782	4954
7	50795	42969	38444	40796	42971	51077	46453	37423	47587	43329	45543	49848	44893	38216											
35689	35211	38821	40462	41918	33640	35836	33921	38199	39186	40149	34846	34579	37165	36677	40282	37772	37791	39952	34859	38286	34111	40616	43830	40787	4421
6	41574	39050	34459	45062	35558	42996	37827	35240	38864	39958	42855	45942	35790	32738											
42386	35597	36789	39902	42767	36413	36211	39452	38512	42271	36957	35197	34227	39218	41571	47234	43341	41045	48288	36261	39063	38408	40591	40274	38301	4897
3	41281	42192	37793	47362	37986	43866	42138	37379	43690	42986	39088	46039	40781	39893											
40803	40823	34279	41829	41182	35272	36819	40690	43406	43410	35215	37142	35284	39434	41361	45314	41086	41169	45228	37671	39022	39307	41349	45178	42130	4659
8	40822	44958	39079	46321	43013	47513	37193	36084	39444	42392	42154	46045	39109	36348											
36974	39027	33899	39735	38805	33623	38499	31914	34466	40173	37074	40210	32865	34666	40126	40446	35209	37922	43939	36786	36143	38062	39030	38322	39351	4524
0	35657	41338	31878	39058	37927	41674	40678	32583	40285	37957	36764	39848	34925	35197											
41852	43271	41486	48625	43912	36955	38119	42277	45766	50891	42465	37355	36456	46814	40933	47645	40764	43817	47372	36062	45263	44786	46169	48188	43067	5160
3	43014	46429	36751	50354	39041	40607	40859	41526	43723	44048	44395	53523	40943	39545											
37495	36509	39303	39069	40328	35324	35718	39181	41131	38346	40957	31620	36089	39146	41307	44291	40118	39007	43438	33421	37365	36676	38592	41589	41726	5085
4	41200	36755	37091	45014	39286	40631	41296	34232	44114	40094	43690	45969	37908	32325											
42340	43163	41960	42805	45694	37623	39602	35905	43163	46134	44784	38514	39140	40872	46398	43872	39802	41668	49418	40634	38955	37476	48155	41589	43653	5078
6	40402	42894	39168	49055	41429	47408	40931	39398	46684	39989	42999	43736	39724	37047											

Size of matrix N- 40 \* 40  
Size of matrix N- 40 \* 40  
Time for cpu execution in usec: 134  
Time for gpu execution in usec: 162  
-bash-4.1\$

Fig 6: 40\*40 Matrix Multiplication

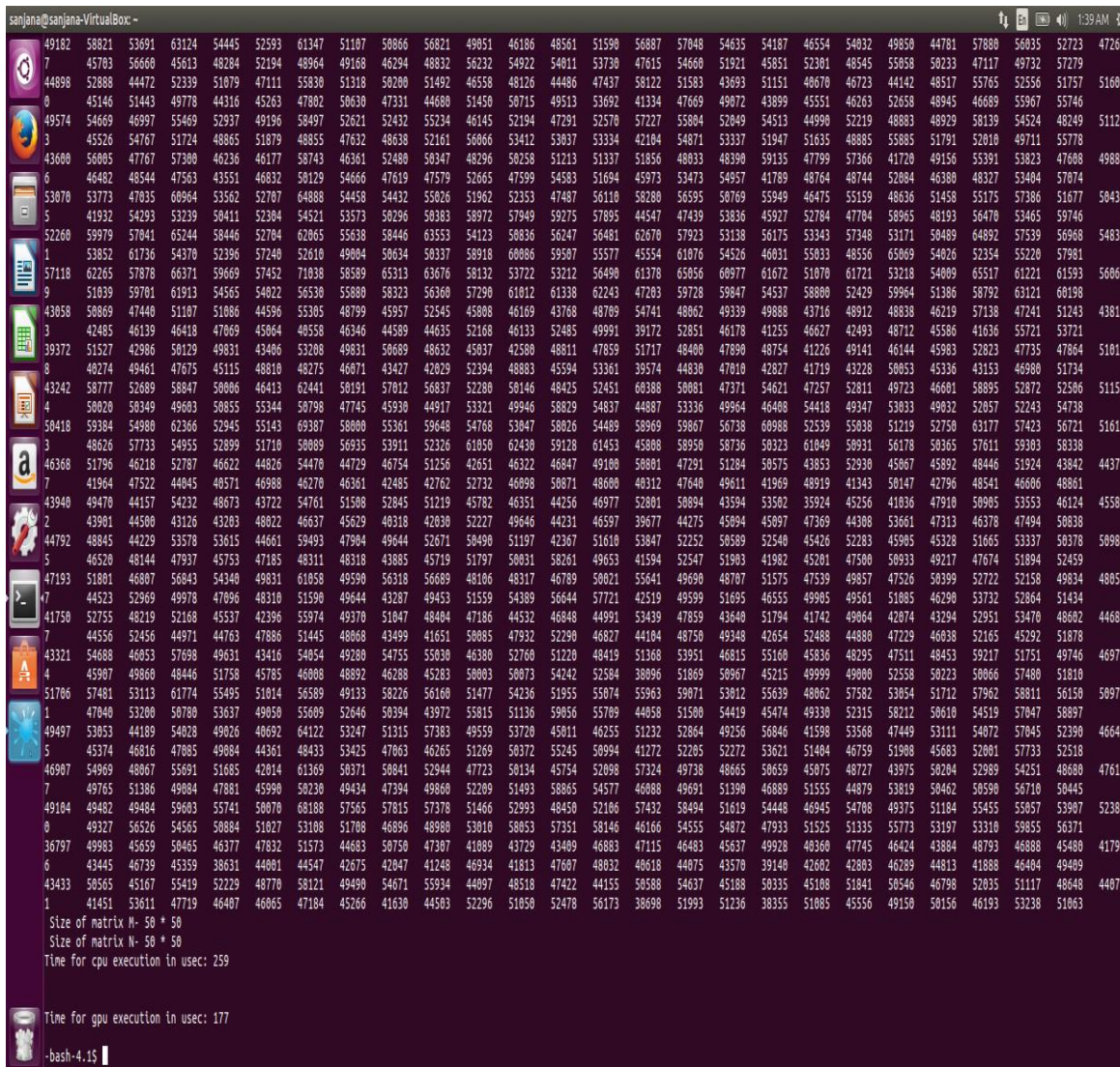


Fig 7: 50\*50 Matrix Multiplication



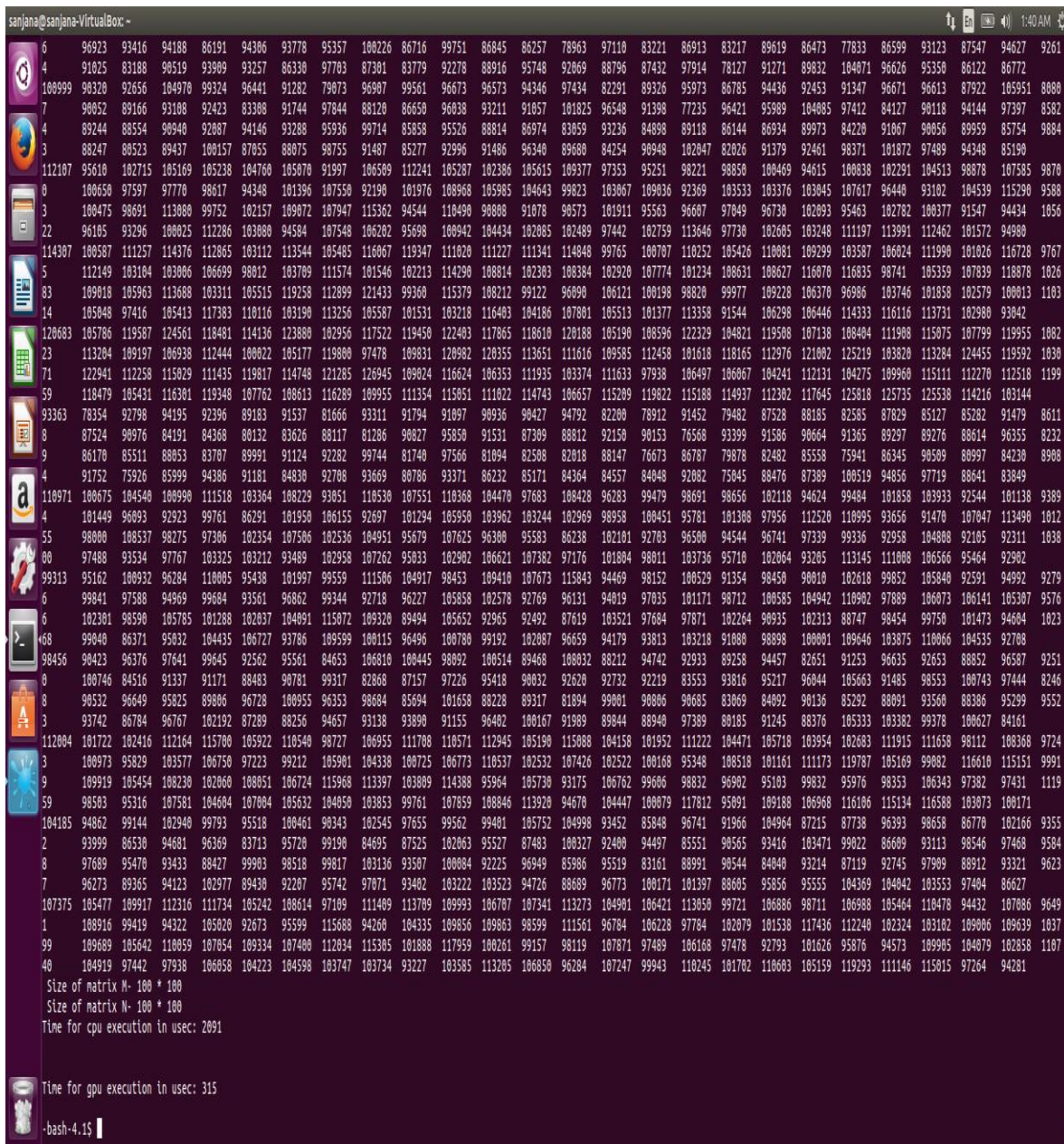


Fig 8: 100\*100 Matrix Multiplication



sanjana@sanjana-VirtualBox: ~

92	289183	303567	281269	276519	277625	306315	272565	277207	282523	290626	288140	272820	274569	276324	287231	286904	258810	260447	291665	270621	275700	293650	281395	287179	2703
76	276408	256777	274318	292942	289672	286287	279949	282706	288064	307011	297985	303268	285055	277664	275743	280610	287424	296485	294219	289043	279998	272433	287184	264655	2933
88	291016	276207	280871	280908	296689	282773	287597	275024	280070	289601	289325	291665	305087	283070	282305	290929	278234	293681	300400	281244	286082	303271	279699	274301	2707
02	273711	279306	268966	276214	279523	297136	279166	288357	275749	287836	285973	274964	279958	270662	294083	296213	263222	283753	286237	302677	289568	278413	290112	274603	2766
65	271194	282951	285043	279299	266517	284815	267884	283493	289712	277034	293744	283244	285023	291682	263111	285621	283378	287334	276409	276482	283944	318838	286289	266432	2666
13	286589	288408	285958	285644	262892	286211	297351	286467	278636	266238	271829	302506	293468	315731	280342	295871	281664	278769	261750	286753	294875	291836	270263	276475	2799
87	265574	287344	292444	280642	300359	290968	293956	301262	293132	281797	281935	275424	280152	261502	293324	285757	279047	280498	288770	277481	290636	287537	290750	277541	2829
05	253155	285892	260651	284156	281399	300606	279486	280437	284146	282501	284472	290919	274990	271641	292695	265911	288519	280635	284304	283099	290895	278685	302283	274232	2942
40	281417	277626	275870	276159	281867	281387	275691	296792	275063	287313	278486	287323	261582	280341	276664	278131	307386	290418	290235	293141	293334	293986	285565	277062	2960
66	284186	289554	282956	276491	275074	290300	286333	265542	288877	285865	277620	268309	311056	282834	280756	297975	281169	280170	301956	279954	286671	303809	278070	293082	
320582	301514	323386	288492	306335	286589	296136	330844	299307	296280	301544	317934	314249	291727	282904	300923	310160	290456	312743	301410	281350	294873	313486	307686	288546	3234
90	280484	306415	285086	285013	280404	280858	297723	302625	280471	299308	296922	305006	299129	303919	304655	312954	308222	301152	320797	303085	321022	294473	311311	293530	3030
69	315605	329661	310681	315439	290588	318232	284074	301254	290903	316887	312836	301083	305073	290230	309372	308777	274072	277357	315030	283569	304724	313599	280137	287569	2878
43	295731	297047	292353	303566	314120	292329	297241	313411	296777	317413	321239	324372	300714	285831	290746	291087	295230	306001	319906	319395	297512	301909	294722	288888	3093
90	325192	299248	303869	290875	312014	291206	305832	299335	303874	309304	304437	306661	322363	316209	300599	298009	309670	308432	314968	301556	297915	321471	290676	293345	2831
81	297353	300895	293221	288204	292163	300893	297498	320713	290067	304565	319151	298863	300077	279145	304703	320937	289253	290667	297274	313222	290026	290447	318855	317624	3012
50	280111	302750	299366	299544	283372	300871	303014	302132	306075	302225	300465	310638	289913	304529	289422	298001	312317	307441	306109	293511	290866	313078	301808	289388	2887
69	307679	313490	302165	304286	307439	305923	303319	306213	299704	295578	298542	321966	301572	330051	301361	320268	307041	299141	292846	295767	314502	312887	283998	311318	2982
60	283195	307159	300802	299696	319772	300657	315859	319154	312528	307962	295718	295398	311502	291110	290633	299054	289793	289623	301364	310949	305037	296322	302290	295803	3041
75	270343	310947	303547	299731	290181	319426	294280	294792	290232	304457	309205	315393	302571	296502	321010	291563	310156	308000	301032	300128	307058	290165	312020	307114	3138
41	303721	293409	299012	279065	291893	296800	293083	325471	299240	319323	287215	304172	288883	296082	285731	285904	327906	317907	311670	297210	326284	302701	290641	299172	3163
08	305531	303237	311959	307574	293646	300589	312116	305451	308302	299072	296043	300112	328718	304847	289549	315126	293032	288883	326944	286573	290595	315396	297581	301332	
309009	298524	311763	285505	305408	285574	286153	323891	286972	288162	301640	300883	298709	303036	270609	296307	314225	292583	302626	285510	279197	299261	299934	307834	289990	3026
03	285234	307612	290267	291743	283744	285728	303377	288866	296238	290610	307220	301157	287264	307264	291476	316752	302689	304115	314331	294225	290937	291158	302342	280176	3031
45	300805	325088	289513	283385	284542	319953	281206	300186	290027	302171	315081	290799	286785	296328	304473	295297	272959	268092	300745	289974	289458	303907	285310	274309	2882
55	290246	281641	281553	297418	298164	293199	300167	300180	288047	313993	314082	306640	302767	293130	297770	292779	304453	307490	308964	303693	295503	280385	300233	297144	3006
26	301692	293445	291375	301609	300052	285323	296144	290329	299004	306317	290734	310433	310209	291800	290811	290201	296822	325891	307973	286693	297592	308953	290975	282449	2847
85	292034	303317	286072	291103	295721	309457	293552	310829	290544	299980	300678	293024	285996	272001	301532	307514	265927	305416	297951	313268	300381	282923	300259	304206	3076
63	271019	293937	301584	293722	281888	296211	300575	303330	291524	291318	306856	293064	289993	306126	275307	296285	309092	295357	301829	293489	311494	332332	295568	284360	2814
44	308348	297148	299245	309362	290131	301003	306061	292210	300236	287696	282267	319754	305630	325449	291810	303926	304749	294800	277724	296940	302771	290965	286355	296922	3025
19	285515	298705	310782	294225	314622	294527	306339	303681	307220	285726	288757	290511	292711	281379	305154	283453	289066	286681	293536	305280	290921	280357	301015	296502	2993
75	267235	301517	300057	294088	289994	319564	277471	284690	292810	300664	313274	297590	294326	282770	310250	277563	297206	314606	290863	293780	301104	300621	314274	290900	2882
97	292968	292449	282581	270685	300438	290266	295333	315299	303505	309910	283195	293676	281116	290616	269705	277819	327701	290330	309333	309112	312273	292965	292518	286035	3165
46	296522	299543	303167	287148	294073	294798	295801	280524	304418	290011	289775	285185	308690	295316	293977	307491	295780	296080	320159	293129	300769	311711	290504	294390	
329270	319836	327497	297356	310134	297090	313263	325946	296963	303649	316700	313998	307012	303838	277436	313632	318529	300074	321561	295907	293132	304372	316819	314052	296137	3165
60	296872	304305	299810	304082	282606	300482	303850	302513	304393	307770	307980	313124	290439	323603	317988	319594	315110	311749	322994	312955	309609	303107	300839	309602	3160
53	315615	339488	306210	290764	301030	330841	296084	309633	296332	317479	313379	298377	291953	304329	307031	316423	301481	274683	315398	288791	309729	319362	294023	283755	2952
78	300103	285994	301650	322852	315562	304851	313610	316158	304161	324390	320820	324082	320285	301169	301352	300608	317286	314279	319661	322491	306433	304555	307700	304099	3163
76	328255	315937	290026	312468	327350	294601	307005	295746	303373	316596	309940	324231	333049	313401	299014	301027	295006	335223	320833	310527	310600	335253	315219	290138	2846
26	303951	323157	315001	305065	312760	309320	300153	318060	300800	309760	326038	309140	319445	293345	305986	325314	283913	315804	325518	329366	313734	302879	327199	315683	3118
32	297731	305389	300129	300133	303532	317801	300365	297644	301401	315281	313453	306996	301909	320072	295355	305673	320903	307715	311603	300915	326073	339755	305449	286732	3013
67	318043	314365	310530	315168	296088	314426	321363	312968	317441	293347	308578	319171	315060	334914	292114	313119	327398	299790	289806	308469	310945	310157	290847	319585	3012
43	296337	315511	319075	312755	315071	315832	327135	340049	328628	309230	312897	304416	315042	293210	309060	317208	294603	293794	322747	310197	319035	293622	320910	304019	3101
11	277781	309064	304365																						

Table 1: Execution times for different sizes of the matrix

<u>SIZE OF MATRIX</u>	<u>CPU EXECUTION TIME IN MICROSECONDS</u>	<u>GPU EXECUTION TIME IN MICROSECONDS</u>
<b>10*10</b>	2	73
<b>20*20</b>	17	79
<b>30*30</b>	56	99
<b>40*40</b>	134	162
<b>50*50</b>	259	177
<b>100*100</b>	2091	315
<b>300*300</b>	63950	1637

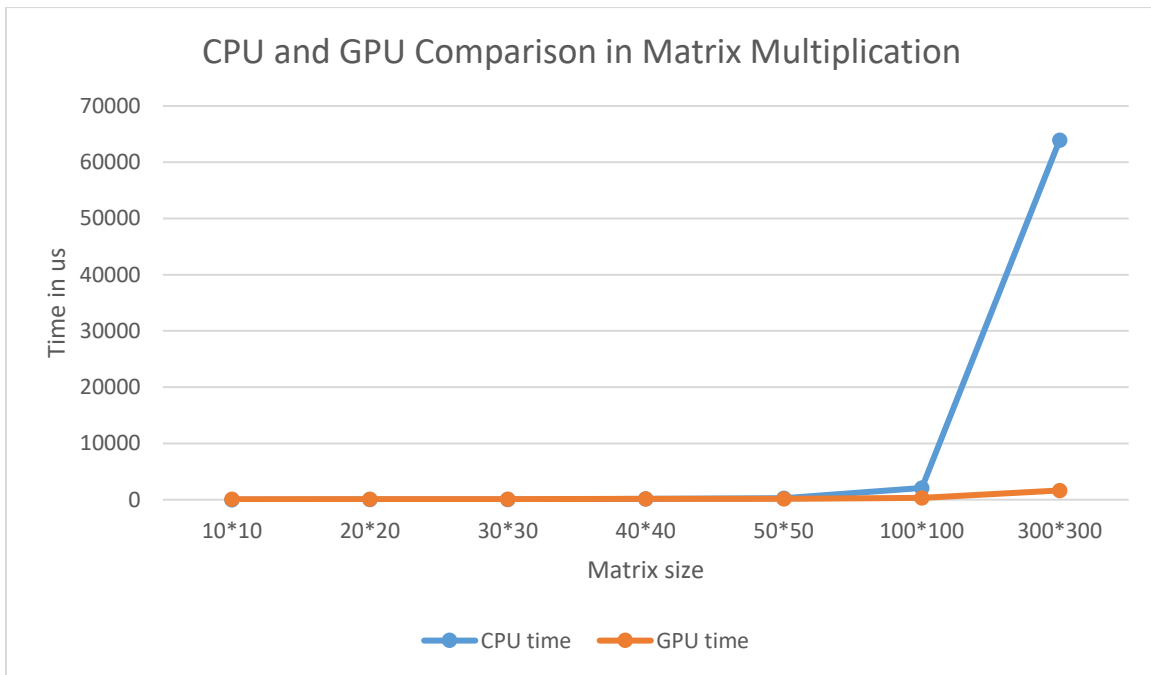


Fig 10: Graphical Representation of the CPU and GPU execution time for different sizes

## APPLICATIONS:

**There are various applications of Matrix Multiplication. Few of them are listed below:**

1. Numerical Analysis.
2. Pattern Recognition.
3. Scientific Computing.
4. Used in the Computer Graphics industry involved in Video Gaming domain where construction and manipulation of realistic animation of a polygonal figure.

## CONCLUSIONS:

1. CPU and GPU execution time varies with respect to the size of the matrix and it is observed that the CPU execution time changes widely as the size varies compared to the GPU execution time.
2. Using CUDA to implement scientific applications on the GPU we can fully exploit SIMD programming to populate with work the cores the GPU possess.
3. Floating point bandwidth from the closest cache GPU is several times lower than the current CPU.
4. There is a utilization of 17% of arithmetic resources on the NVIDIA hardware.

## FUTURE SCOPE:

1. The code can be extended to bigger sizes and the execution times of CPU and GPU can be verified.
2. In terms of application wise it can be implemented for security which involve facial recognition or finger print recognition.
3. It can be extended to be implemented for performing mathematical computations for 3 – Dimensional figures.

## REFERENCES:

- <https://graphics.stanford.edu/papers/gpumatrixmult/gpumatrixmult.pdf>
- <https://people.eecs.berkeley.edu/~sangjin/2013/02/12/CPU-GPU-comparison.html>
- <http://www.nvidia.com/object/cuda-home.html>
- <http://www.shodor.org/media/content/petascale/materials/UPModules/matrixMultiplication/moduleDocument.pdf>
- The GPU on Matrix - Matrix Multiply: Performance Study and Contributions by *Jose Maria Cecilia, Jose Manuel Garcia, Manuel Ujaldon*.
- [http://www.nvidia.com/content/nvision2008/tech\\_presentations/Game\\_Developer\\_Track/NVISION08-Image\\_Processing\\_and\\_Video\\_with\\_CUDA.pdf](http://www.nvidia.com/content/nvision2008/tech_presentations/Game_Developer_Track/NVISION08-Image_Processing_and_Video_with_CUDA.pdf)
- <http://supercomputingblog.com/cuda/advanced-image-processing-with-cuda>
- <http://homepages.math.uic.edu/~jan/mcs572/devmemdata.pdf>
- <http://cuda-programming.blogspot.in/2013/01/shared-memory-and-synchronization-in.html>