Title: Multiplication of matrix and vector using CUDA C.

Outcome: At the end of this seesion students will be able to:

1) Understand CUDA structure.

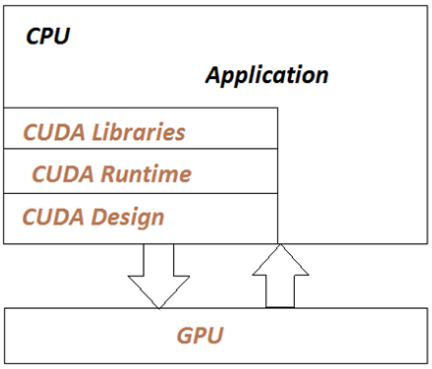
2) Understand programming structure of CPU & GPU.

3) Able to write the code & test it for result.

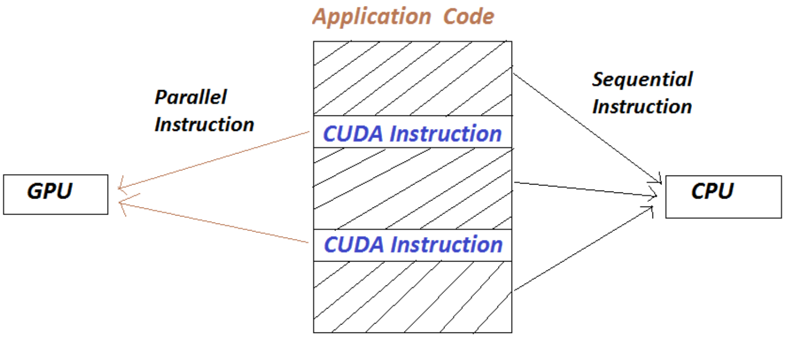
4) Compare execution time for sequential & parallel programs.

Theory:

- CUDA Architecture:



- Programming Structure of GPU & CPU:



- CUDA Kernel:

The function which are executed on GPU are called as kernels.Kernels are full program or function invoke by the CPU and executed on GPU.A kernal is executed N number of times in parallel on GPU by using N number of threads.

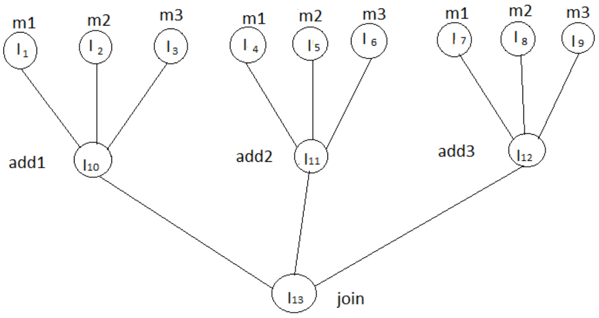
Invocation: kernel_name<<<grid,block>>>(argument,list);

kernel is defined as:

```
_global_voidkernel_name(arguments)

{

.........

}
```

$$\begin{bmatrix} 3 & 1 & 2 \\ 4 & 2 & 1 \\ 3 & 2 & 7 \end{bmatrix} * \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} = \begin{bmatrix} 47 \\ 53 \\ 100 \end{bmatrix}$$

Task graph:



Procedure:

1) Write a program using text editor, name the source code with .cu extension.

2) Compile the program using nvcc compiler.

3) Execute the program.

4) Verify the result.

Theory

Conclusion:

Execution time for parallel and serial for multiplication of matrix and vector is compared. Performance of parallel program is more as compared to sequential addition.