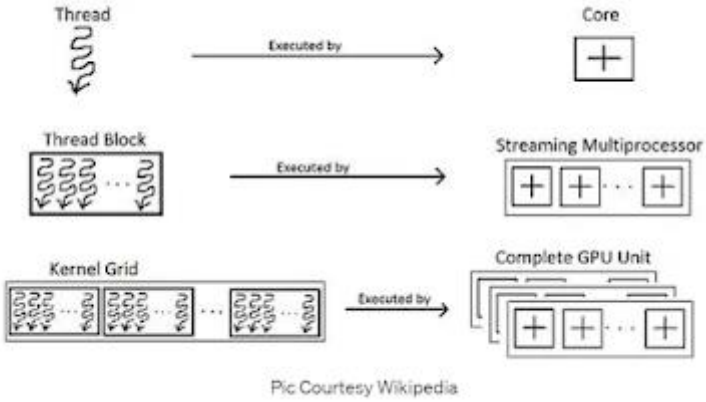
Hello All,

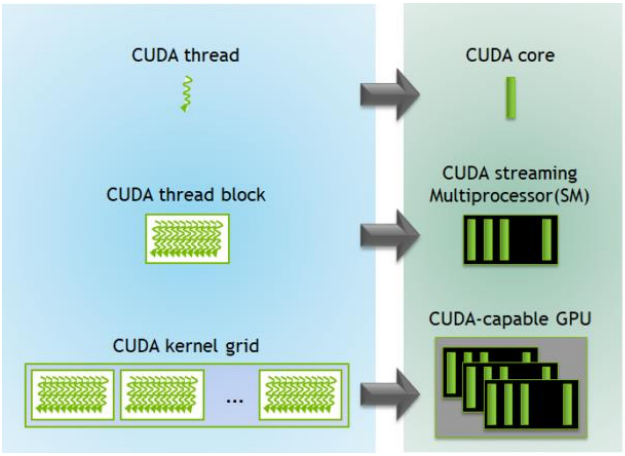
In my previous post related to Introduction to GPU programming using NVIDA CUDA Tool Kit(link in comment section) I have explained about how to write a simple program(Performing addition of two arrays) using CUDA.

In this post let us understand how the CUDA kernel will launch with provided block dimension and grid dimensions parameters.

A GPU will follows a single instruction multiple thread(SIMT) architecture it means that the multiple threads are issued for processing the same instruction.

These threads are organized in to blocks and blocks are organized in to grids.



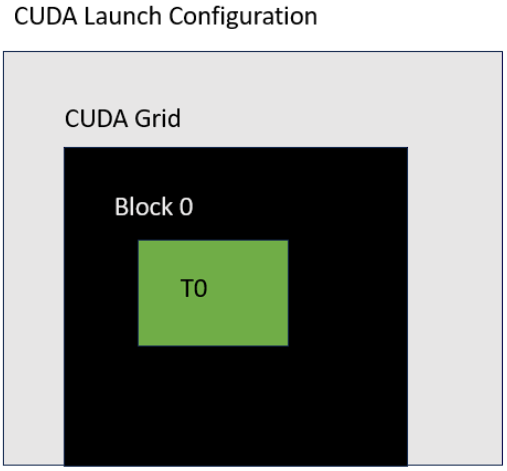


Let us consider the example of CUDA Hello World Program where we launch the CUDA kernel with total number of threads in a block as 1 and there is 1 such block in a grid.

HelloWorld.cu

```
//Pre-processor directives
#include <stdio.h>
#include "cuda_runtime.h"
#include "device_launch_parameters.h"//Device code
__global__
void cuda_kernel()
{
    printf("Hello World!");
}//Host code
int main()
{
    cuda_kernel <<< 1, 1 >>> ();  //CUDA kernel launch
    cudaDeviceSynchronize();
    cudaDeviceReset();
    return 0;
}
```

In the above code to launch the CUDA kernel two 1's are initialized between the angular brackets. The First parameter indicates the total number of blocks in a grid and second parameter indicates the total number of threads inside a block. Thus in the above code total number of threads in a block is 1 and there is 1 such block in a grid.



Implicit variables initialized by CUDA runtime:

threadIdx

1. It is a dim3 variable and each dimension can be accessed by threadIdx.x, threadIdx.y, threadIdx.z.

2. Refers to the thread ID with in a block and it starts from 0. So, if number of threads in X dim in a block is 32, then threadIdx.x ranges from 0 to 31 in each block.

blockIdx

1. It is a dim3 variable and each dimension can be accessed by blockIdx.x, blockIdx.y, blockIdx.z.

2. Refers to the block ID in a grid and it starts from 0.

blockDim

1. It is a dim3 variable.

2. Refers to the maximum number of threads in a block in all the dimension and it starts from 1.

3. All thread blocks have the same dimension.

gridDim

1. It is a dim3 variable.

2. Refers to the maximum number of blocks in a grid in all the dimension and it starts from 1.