ASSIGNMENT 1

Problem Statement: Design Parallel algorithm to add two large vectors.

Objectives: To learn CUDA Architecture and programming concepts.

Requirements:

1. OS : Fedora 20/Ubuntu 64 bit
2. CUDA API
3. Nvcc compiler
4. NVIDIA GPU
5. RAM 4GB
6. HDD : 500GB

Theory:

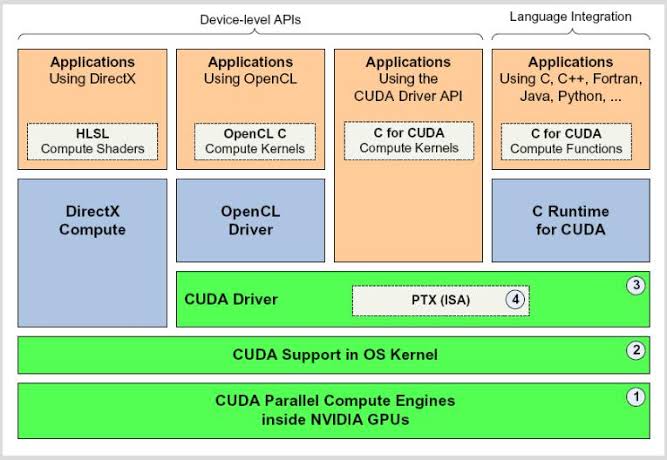
The CUDA Architecture The CUDA Architecture consists of several components, in the green boxes below:

1. Parallel compute engines inside NVIDIA GPUs

2. OS kernel-level support for hardware initialization, configuration, etc.

3. User-mode driver, which provides a device-level API for developers

4. PTX instruction set architecture (ISA) for parallel computing kernels and functions



Advantages of CUDA over the traditional approach to GPGPU computing:

* Programming interface of CUDA applications is based on the standard C language with extensions, which facilitates the learning curve of CUDA
* CUDA provides access to 16 KB of memory (per multiprocessor) shared between threads, which can be used to setup cache with higher bandwidth than texture lookups
* More efficient data transfers between system and video memory
* No need in graphics APIs with their redundancy and overheads
* Linear memory addressing, gather and scatter, writing to arbitrary addresses
* Hardware support for integer and bit operations

Main limitations of CUDA:

* No recursive functions
* Minimum unit block of 32 threads
* Closed CUDA architecture, it belongs to NVIDIA.

Time Analysis:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Operation | Input Size | Sequential  time | Parallel  Time | Efficiency |
|  | N=256 | 0.01 | 0.02 | 0.5 |
| Vector Addition | N=1024 | 0.01 | 0.02 | 0.5 |
|  | N=2048 | 0.02 | 0.01 | 2.0 |