The CUDA programming model is a heterogeneous model in which both the CPU and GPU are used. In CUDA, the host refers to the CPU and its memory, while the device refers to the GPU and its memory. Code run on the host can manage memory on both the host and device, and also launches kernels which are functions executed on the device. These kernels are executed by many GPU threads in parallel. Below is the sample code to perform addition of two arrays with a million elements each. It supports both Linux and windows OS.  
  
Getting started with CUDA: <https://lnkd.in/gG8P8NVD>



Here is a CUDA example to find out the GPU information in the system using CUDA Toolkit.



Profiling CUDA kernels using **nvprof** utility:  
  
The nvprof profiling tool enables you to collect and view profiling data from the command-line. **nvprof** enables the collection of a timeline of CUDA-related activities on both CPU and GPU, including kernel execution, memory transfers, memory set and CUDA API calls and events or metrics for CUDA kernels. Profiling options are provided to **nvprof** through command-line options. Below is the screen shot of sample cuda code and its profiling data.

