Utilizing NVIDIA GPUs for Waveform Analysis for the Nab Experiment

D.G. Mathews, A.P. Sprow, C.B. Crawford

University of Kentucky

GPU Introduction

GPUs are composed of a large number of small computational cores compared to CPUs which are generally just a few larger cores. While the CPU excels at linear processes, the GPU excels at parallel tasks. For this project, the goal was to find a way to use the massive parallelism of a GPU to rapidly analyze waveform data from the Nab experiment.

GPU Functions and Coding

Nvidia's CUDA language was used for this project. While it based on C/C++, accessing the parallel nature of the GPU requires adjustments in how functions are defined. GPU functions use Blocks and Threads arranged in a Grid for identifying how the function will evaluate. The following function demonstrates this.

```
__global__ void addition(int *A, int *B, int *C){
    int id = threadIdx.x + blockIdx.x*blockDim.x;
    C[id] = A[id] + B[id];
}
```

This function simply adds the two arrays A and B and stores the result back in C. The number of blocks and threads used is dictated by the function call.

```
addition<<<5, 10>>>(A, B, C);
```

In this function call, 5 blocks were allocated, each with 10 threads for a total of 50 threads which means 50 values would be added. Any values past the 50th would not be affected by this function

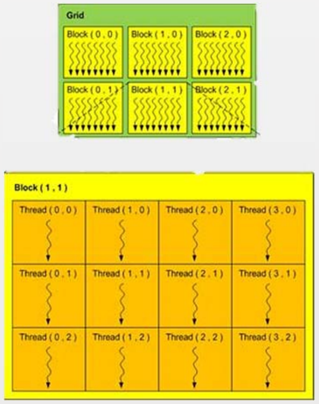


Diagram of the arrangement of Blocks and Threads within the Grid. The Grid is broken up into a 3 dimensional array of Blocks. The Blocks are broken up into a 3D array of Threads.

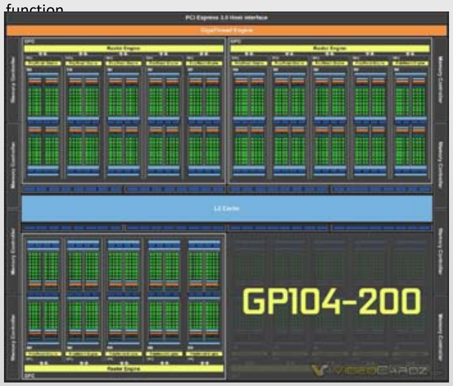
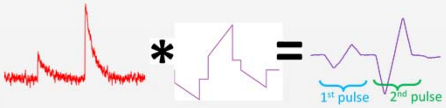


Diagram of the GP104 chip used in the GTX 1070 for testing. This chip features 1920 CUDA cores split between 15 SMs. There are 8 GB of VRAM associated with this card. For the experiment, 2 GTX 1080 TI cards will be used.

Why Convolution

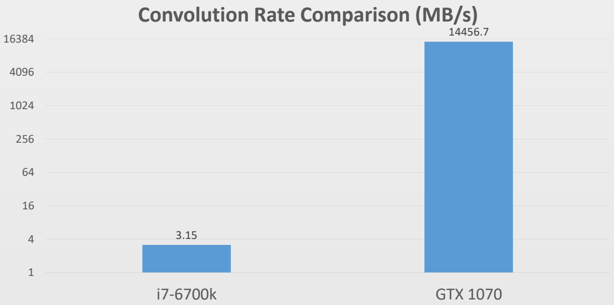


Example of a Double Trapezoidal Filter Convolution on a waveform with two pulses.

Convolutions like the one shown here reduce noise levels while maintaining data integrity. Different filter types can highly key features of the data that are difficult to identify otherwise.

Convolution Rate: GPU vs CPU

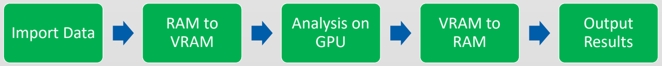
Simple codes were written to demonstrate the capabilities of a GPU compared to a CPU for convolutions. These rates only include the rate of the convolution itself. The CPU code only utilized one of its 8 cores so the result presented here has been multiplied by 8 to demonstrate its theoretical maxima.



Even with the adjustment for cores, the GTX 1070 from Nvidia was ~4600 times faster than the i7. This translates to a rate of 1.1 MHz of 2500 integer long waveforms, the standard size for the Nab experiment. The Nab data rate is predicted to be 150 kHz.

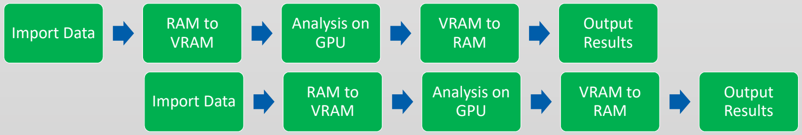
Overlapping Memory Transfers

One of the main hurdles in utilizing the GPU is memory usage. The GPU cannot directly access information stored on RAM. Instead it uses its own storage system VRAM. As such, all information needs to be transferred from the RAM to VRAM before it can be utilized. This process is shown below.



Import Data → RAM to VRAM → Analysis on GPU → VRAM to RAM → Output Results

This slowdown can be avoided by overlapping transfers with analysis steps. Most Nvidia GPUs can transfer and analyze at the same time with minimal impact on each process. Due to this, the following analysis pattern is more efficient.



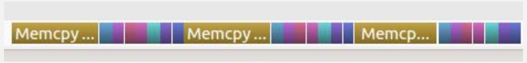
Import Data → RAM to VRAM → Analysis on GPU → VRAM to RAM → Output Results

Import Data → RAM to VRAM → Analysis on GPU → VRAM to RAM → Output Results

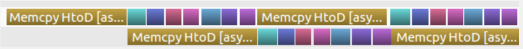
To layer the analysis process, cudaStream functionality is used. A cudaStream is analogous to a CPU thread and allows for certain processes to be overlapped.

cudaStream Results

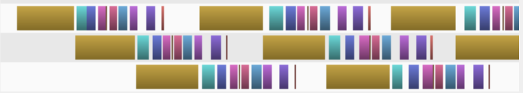
The following images are from NVidia's Visual Profiler tool. These represent the analysis process using different numbers of cudaStreams. These images do not share the same scale.



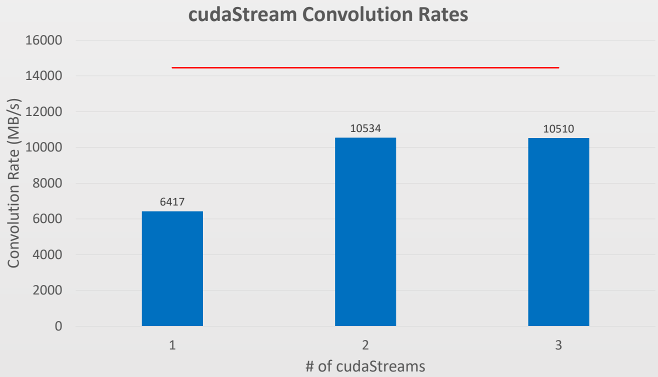
1 cudaStream: No overlap between memory transfer and analysis steps.



2 cudaStreams: Almost perfect overlap between memory transfers and analysis steps.



3 cudaStreams: Same level of overlap despite the extra stream. No performance gain over 2 cudaStreams due to this.



This graph shows the analysis rate of each of the processes shown above compared to the theoretical limit represented by the red line. By simply overlapping the memory transfers, the analysis rate increased by ~60 percent. This means that even with the extra memory transfers, the GTX 1070 is still around 3300 times faster than the Intel i7.

Applications to Nab Experiment

Traditionally waveform analysis has been handled in real-time by a DAQ system and then more robust offline analysis has been done by CPUs. A GPU powered server can be used to do real-time analysis with greater precision than a DAQ system. The FFT libraries provided by Nvidia use 32 bit floating point precision which is more precise than the integer arithmetic done by a DAQ system.

As the convolution rate is well above the real-time data speed, more time can be given to trigger analysis and event reconstruction. With multiple GPUs on a server, it is possible to do online analysis with as much precision as offline analysis.

References and Support

We gratefully acknowledge support of the DOE Nuclear Physics program under contract DE-SC0008107.

Block and Thread Diagram: http://cuda.ce.rit.edu/cuda_overview/cuda_overview.htm
GP104 Diagram: http://videocardz.com/60163/nvidia-releases-full-geforce-gtx-1070-specifications-and-first-specifications and first benchmarks
For more information on the CUDA Toolkit and using Nvidia GPUs for computational work: https://developer.nvidia.com/cuda-zone