GPU accelerated programing

Amaro Jr. and Rafael A. N.
29/07/2020

Engenharia - UFJF

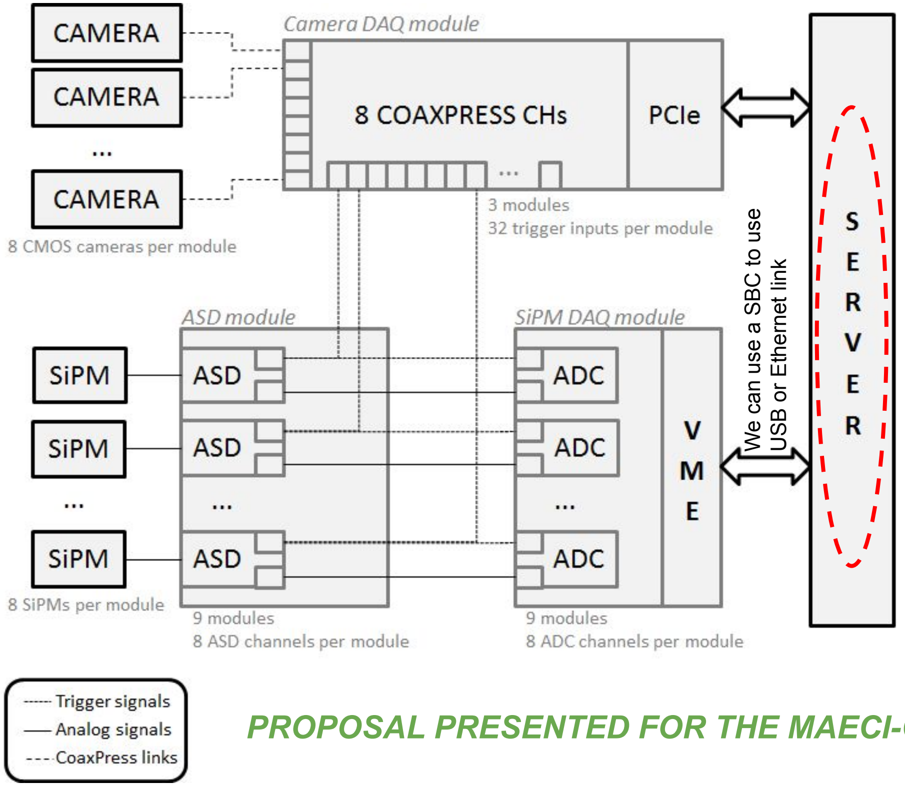
Universidade Federal de Juiz de Fora (UFJF)
Juiz de Fora, MG

Universidade Federal de Juiz de Fora

CYGNO system overview

- *The detector is composed of 18 readout regions, each equipped with 1 sCMOS sensor and 4 Photomultipliers.*

→ TOTAL: 18 sCMOS sensors and 72 Photomultipliers.



The need of Trigger Decisions based on tracking images might make it necessary to use GPU processing

This presentation serves to kick-start the search for a GPU-based solution

It is important to start it now as the selection of the readout hardware might be influenced by it

PROPOSAL PRESENTED FOR THE MAECI-CONFAP CALL

Why parallel computing on a GPU?

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

- Optimized for data parallel operations
    - Particularly matrix calculations

- Very fast compared with procedural computing

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

Frame grabbers support

- To increase speed, some framegrabbers have PCI access to the GPU internal memory
- Technologies:
  - DirectGMA (AMD)
    - FireBird frame grabber are compatible with GPUDirect
  - GPUDirect (NVIDIA)
    - FireBird and Phoenix frame grabbers are compatible with GPUDirect

EXAMPLE

FireBird CoaXPress Frame Grabber (4xCXP12-3PE8)

Part Number: AS-FBD-4XCXP12-3PE8

The **FireBird Quad CXP-12 3PE8** frame grabber is a high-performance four-link board, compliant to CoaXPress v2.0. This 8-lane Gen3 PCI Express frame grabber supports CoaXPress speeds up to 4 x CXP-12, resulting in data rates of up to 12.5 Gbps per link. It is fitted with Micro-BNC connectors and offers I/O access on the end bracket.

GPU hardware

Possibilities should be investigated...but

- We could buy GPU cards to assembly custom computers using cards like those on the bottom
- Or buy GPU servers with PCIe slots like this one on the right (just an example)

FEATURES

- Four DisplayPort 1.4 Connectors[3]
- DisplayPort with Audio
- 3D Stereo Support with Stereo Connector[3]
- NVIDIA GPUDirect™ Support
- NVIDIA NVLink Support[1]
- Quadro Sync II[4] Compatibility
- NVIDIA nView® Desktop Management Software
- HDCP 2.2 Support
- NVIDIA Mosaic[5]
- Dedicated hardware video encode and decode engines[6]

SPECIFICATIONS

GPU Memory	32 GB HBM2
Memory Interface	4096-bit
Memory Bandwidth	Up to 870 GB/s
ECC	Yes
NVIDIA CUDA Cores	5,120

GPX XT8-22S1-16GPU 2U

Supports:

- Intel Xeon Scalable
- 3TB DDR4 ECC RDIMM ← *RAM memory*
- 2 2.5" NVMe Hot-Swap
- 16 PCIe 3.0 x16 ← *Cameras*
- NVMe
- GPU-Optimized
- Dual 10-Gigabit Ethernet ← *Photomultipliers (or via USB)*

STARTING PRICE $7,089.00 CONFIGURE

Code development

Universidade Federal de Juiz de Fora

- **CUDA**
  - C, C++, Fortran
  - Can be used with python with Pycuda (3rd party)

- **OPENCL**
  - C, C++
  - Can be used with python with Pyopencl (3rd party)

Code development

- Although you could use python for analysis, calculation modules could still be written in C/C++
- Python hill hinder the speed of analysis

Portability

- Python code used nowadays could potentially be parallelized with GPU acceleration
- A port to a faster language should be discussed

Accessibility

- **NVIDIA**
  - Well documented
  - Most cards have CUDA support
  - Drivers are proprietary

- **AMD**
  - Poorly documented
  - Most cards have OPENCL
  - Only FirePro cards have DIRECTGMA
  - Open source drivers

Simple example:

Simple kernel that adds two values, 2^20 numbers

```
// function to add the elements of two arrays
__global__
void add(int n, float *x, float *y)
{
    int index = threadIdx.x;
    int stride = blockDim.x;
  for (int i = index; i < n; i += stride)
      y[i] = x[i] + y[i];
}
```

The setup:

First with a single thread and memory block:

```
[amaro@Skynet cuda_test]$ nvprof ./add1
==247863== NVPROF is profiling process 247863, command: ./add1
Max error: 0
==247863== Profiling application: ./add1
==247863== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:  100.00%  272.87ms         1  272.87ms  272.87ms  272.87ms  add(int, float*, float*)
      API calls:   63.52%  272.89ms         1  272.89ms  272.89ms  272.89ms  cudaDeviceSynchronize
                   34.98%  150.29ms         2  75.144ms  551.47us  149.74ms  cudaMallocManaged
                    1.24%  5.3426ms         1  5.3426ms  5.3426ms  5.3426ms  cudaLaunchKernel
                    0.17%  737.82us         2  368.91us  321.21us  416.62us  cudaFree
                    0.05%  198.24us        97  2.0430us     189ns  84.215us  cuDeviceGetAttribute
                    0.02%  82.698us         1  82.698us  82.698us  82.698us  cuDeviceTotalMem
                    0.01%  40.885us         1  40.885us  40.885us  40.885us  cuDeviceGetName
                    0.00%  3.6010us         1  3.6010us  3.6010us  3.6010us  cuDeviceGetPCIBusId
                    0.00%  1.8940us         3     631ns     288ns  1.0230us  cuDeviceGetCount
                    0.00%     888ns         2     444ns     281ns     607ns  cuDeviceGet
                    0.00%     374ns         1     374ns     374ns     374ns  cuDeviceGetUuid

==247863== Unified Memory profiling result:
Device "GeForce 930MX (0)"
   Count  Avg Size  Min Size  Max Size  Total Size  Total Time  Name
       6  1.3333MB  512.00KB  2.0000MB  8.000000MB  5.081312ms  Host To Device
      96  128.00KB  4.0000KB  0.9961MB  12.00000MB  7.032640ms  Device To Host
Total CPU Page faults: 48
```

The setup:

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

Then one memory block with 256 threads:

```
[amaro@Skynet cuda_test]$ nvprof ./add_cuda
==248128== NVPROF is profiling process 248128, command: ./add_cuda
Max error: 0
==248128== Profiling application: ./add_cuda
==248128== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:  100.00%  1.9201ms         1  1.9201ms  1.9201ms  1.9201ms  add(int, float*, float*)
      API calls:   95.07%  168.92ms         2  84.461ms  609.36us  168.31ms  cudaMallocManaged
                    2.99%  5.3088ms         1  5.3088ms  5.3088ms  5.3088ms  cudaLaunchKernel
                    1.08%  1.9261ms         1  1.9261ms  1.9261ms  1.9261ms  cudaDeviceSynchronize
                    0.41%  727.80us         2  363.90us  329.18us  398.62us  cudaFree
                    0.26%  462.53us        97  4.7680us     588ns  188.72us  cuDeviceGetAttribute
                    0.13%  230.53us         1  230.53us  230.53us  230.53us  cuDeviceTotalMem
                    0.05%  97.540us         1  97.540us  97.540us  97.540us  cuDeviceGetName
                    0.00%  5.6500us         3  1.8830us     804ns  3.9670us  cuDeviceGetCount
                    0.00%  4.2140us         1  4.2140us  4.2140us  4.2140us  cuDeviceGetPCIBusId
                    0.00%  3.1210us         2  1.5600us     606ns  2.5150us  cuDeviceGet
                    0.00%  1.1670us         1  1.1670us  1.1670us  1.1670us  cuDeviceGetUuid

==248128== Unified Memory profiling result:
Device "GeForce 930MX (0)"
   Count  Avg Size  Min Size  Max Size  Total Size  Total Time  Name
       6  1.3333MB  512.00KB  2.0000MB  8.000000MB  5.073760ms  Host To Device
      96  128.00KB  4.0000KB  0.9961MB  12.00000MB  7.046080ms  Device To Host
Total CPU Page faults: 48
```

The setup:

Then ~4 thousand blocks with 256 threads:

```
[amaro@Skynet cuda_test]$ nvprof ./add_grid
==248226== NVPROF is profiling process 248226, command: ./add_grid
Max error: 0
==248226== Profiling application: ./add_grid
==248226== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:  100.00%  467.69us         1  467.69us  467.69us  467.69us  add(int, float*, float*)
      API calls:   95.86%  164.57ms         2  82.287ms  479.30us  164.09ms  cudaMallocManaged
                    3.10%  5.3136ms         1  5.3136ms  5.3136ms  5.3136ms  cudaLaunchKernel
                    0.44%  752.45us         2  376.22us  323.61us  428.83us  cudaFree
                    0.28%  473.92us         1  473.92us  473.92us  473.92us  cudaDeviceSynchronize
                    0.19%  333.29us        97  3.4360us     383ns  137.45us  cuDeviceGetAttribute
                    0.09%  155.83us         1  155.83us  155.83us  155.83us  cuDeviceTotalMem
                    0.04%  66.107us         1  66.107us  66.107us  66.107us  cuDeviceGetName
                    0.00%  4.3400us         1  4.3400us  4.3400us  4.3400us  cuDeviceGetPCIBusId
                    0.00%  3.8510us         3  1.2830us     565ns  2.5660us  cuDeviceGetCount
                    0.00%  1.7300us         2     865ns     483ns  1.2470us  cuDeviceGet
                    0.00%     802ns         1     802ns     802ns     802ns  cuDeviceGetUuid

==248226== Unified Memory profiling result:
Device "GeForce 930MX (0)"
   Count  Avg Size  Min Size  Max Size  Total Size  Total Time  Name
       6  1.3333MB  512.00KB  2.0000MB  8.000000MB  5.065184ms  Host To Device
      96  128.00KB  4.0000KB  0.9961MB  12.00000MB  7.009408ms  Device To Host
Total CPU Page faults: 48
```

Conclusions

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

- We are starting to study the possibilities and potential of using GPU in the trigger processing part
- A decision on the use of **DirectGMA (AMD)/GPUDirect (NVIDIA)** must be made, as it influences the choice of the readout electronics and computing hardware
- The **code environment (C++,Python,etc)** and **GPU hardware** should also be chosen but it is not strongly related to the readout electronics.
- *Is there any initial orientations/requirements related to the subject to serve as a guide for making those decisions?*