GPU for Real Time processing in HEP trigger systems

M.Bauce[a,b], **A.Biagioni**[b], **R.Fantechi**[c,f], **M.Fiorini**[d], **S.Giagu**[a,b], **E.Graverini**[e,f], **G.Lamanna**[*f], **A.Lonardo**[b], **A. Messina**[a,b,c], **F.Pantaleo**[e,f], **P.S.Paolucci**[b], **R.Piandani**[f], **M.Rescigno**[b], **F.Simula**[b], **M.Sozzi**[e,f] **and P.Vicini**[b].

[a] *University, Rome "Sapienza", Italy*
[b] *INFN, Rome, Italy*
[c] *CERN, Geneve, Switzerland*
[d] *University & INFN, Ferrara, Italy*
[e] *University, Pisa, Italy*
[f] *INFN, Pisa, Italy*

We describe a pilot project for the use of Graphics processing units (GPUs) for online triggering applications in high energy physics (HEP) experiments. Two major trends can be identified in the development of trigger and DAQ systems for HEP experiments: the massive use of general-purpose commodity systems such as commercial multicore PC farms for data acquisition, and the reduction of trigger levels implemented in hardware, towards a pure software selection system (trigger-less). The very innovative approach presented here aims at exploiting the parallel computing power of commercial GPUs to perform fast computations in software both at low- and high-level trigger stages. General-purpose computing on GPUs is emerging as a new paradigm in several fields of science, although so far applications have been tailored to the specific strengths of such devices as accelerator in offline computation. With the steady reduction of GPU latencies, and the increase in link and memory throughputs, the use of such devices for real-time applications in high-energy physics data acquisition and trigger systems is becoming very attractive. We discuss in details the use of online parallel computing on GPUs for synchronous low-level trigger with fixed latency. In particular we show preliminary results on a first test in the NA62 experiment at CERN. The use of GPUs in high-level triggers is also considered, the ATLAS experiment (and in particular the muon trigger) at CERN will be taken as a study case of possible applications.

The European Physical Society Conference on High Energy Physics
18-24 July, 2013
Stockholm, Sweden

[*]Speaker.

1. Introduction

The trigger system of any HEP experiment has a crucial role deciding, based on limited and partial information, whether a particular event observed in a detector is interesting and must be recorded or not. Every experiment is characterised by a limited amount DAQ bandwidth and disk space for storage, therefore the use of real-time selections to reduce data throughput selectively, rejecting uninteresting events only, is fundamental to make an experiment affordable and at the same time maintaining its discovery potential.

This paper describes the idea to use GPUs for triggering in HEP experiments. This work is included in a project wider in scope, named GAP (GPU Application Project) concerning the use of GPUs for advanced scientific computation in real-time application. The Project covers both applications in HEP for event selection and in medical imaging (CT, PET and NMR), however this paper focuses only on recent results on real-time triggering in HEP.

2. The use of GPUs in low level triggers

A typical trigger system is organised in a cascaded sets of computation levels. The low-level trigger is usually implemented in hardware, based on custom electronics and normally distinct from the DAQ system. The following high-level trigger (HLT) is nowadays implemented in software, using dedicated farms of commodity PCs. These systems are very flexible and scalable, since they make use of commercial technology, computers, and networks.

The use of GPUs in HEP trigger systems provide a large computing power on a single device, therefore allowing complex decisions to be taken with a sufficient speed to match high event rates. In a multi-level trigger architecture, GPUs can quite naturally be exploited in high level triggers, as discussed in Section 4. On the other hand, using GPUs in low-level triggers requires a careful assessment of their real-time performances. Fundamental requirements for a low-level trigger are a low total processing latency and it's time stability, which however are not crucial parameters in the typical applications GPUs have been originally developed for. The main limiting issue here is due to the fact that in its typical use as graphics co-processor in a PC, the GPU receives data and sends back processed results through the PCI-express bus. In addition to that, to get the best performances out of a GPU, its computing cores should be saturated, requiring computation on a significant number of events after a buffering stage. The largest fraction of the GPU processing time comes from the latency in data transfering. In order to mitigate this problem we are considering two different approaches: the use of a dedicated NIC device driver with very low latency (PFRING, by NTOP[1]) and the use of a direct data transfer protocol from a custom FPGA-based NIC to the GPU (NaNet). These two approaches will be described in detail in the following.

2.1 Driver DNA-PFRING

Standard drivers do not guarantee the highest packet capture and transmission speed, especially with small-size packets and high rates. The copy of packets from the NIC to the system buffers and then to the "userland", is done by the CPU twice. PFRING is a new kind of socket that works in connection with the DNA (Direct NIC Access) driver (both developed by NTOP) in order

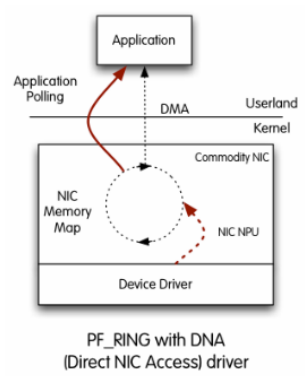

Figure 1: Scheme of data processing using DNA driver (*courtesy of NTOP*).

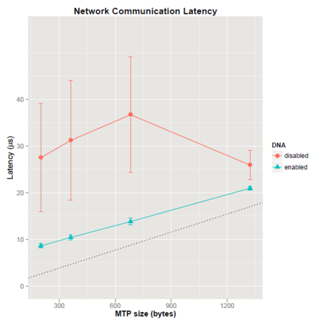

Figure 2: Comparison between standard data transmission and by using PFRING socket.

to allow direct copy of packets from the NIC's FIFO to the memory through DMA (Direct Memory Access). A schematic drawing of the PFRING working principle can be found in Fig.1: using the DNA driver, data transfer is managed by the NIC's NPU (Network Processing Unit) processor through circular buffers that are directly accessible by any application, and therefore for copying to the GPU's memory. The scheduler managing data transfer from the NIC to the GPU's memory has been implemented in a partially preemptive way in parallel streams. This allows to "hide" data transfer latency by exploiting concurrent copy-execution enabled in last generation GPUs. To ensure adequate latency we have measured, with microsecond resolution, the transfer time along an ethernet link. Such a measurement is quite difficult due to different time reference between the two systems (transmitter and receiver). For reliable measurement we used a readout board (TEL62) to send input data and a PC equiped with a standard Intel I350T2 NIC and a NVDIA Tesla K20 GPU as receiver. The TEL62 sends packets containing simulated events: each packets can containes one or more events according to the MTP factor. The instant in which the packet is sent is indicated by the TEL62 with a TTL signal that can be read with an oscilloscope ("start"). When the packet is received in the PC, the "stop" signal for the oscilloscope is produced using the LPT port. The delay due to the parallel port has been measured to be at the level of about 1 μs. Additional "stop" signals are produced in different stages of the GPU calculation: after the copy of data in the GPU and after the GPU computing. Figure 2 shows the comparison between data transfer based on standard intel driver (in red) and on PFRING (in blue). The transfer time improves by more than a factor of 2 and, most importantly, the fluctuations are reduced to a negligible level. In order to hide the latency and to fully exploit the GPU power, the data have to be buffered before the computing. Figure 3 (a) shows the total processing time as a function of the number of buffered events (GMTP) including data transfer from host to device, computing time for a simple kernel, and copy of results from device to host. From fig.3 (b) it's evident that the latency per event decreases increasing the number processed events. Since the processing time is relatively small, the latency depends on the gathering time needed to buffer events (shown in Fig. 4 (a)). For this plot the start signal corresponds to the first event of the GMTP buffer, while the end is produced when all the events in the buffer are processed. Figure 4(b) shows the component of the latency due to the GPU system only, by excluding the gathering time (considering as "start" the recording of the last packet of the GMTP

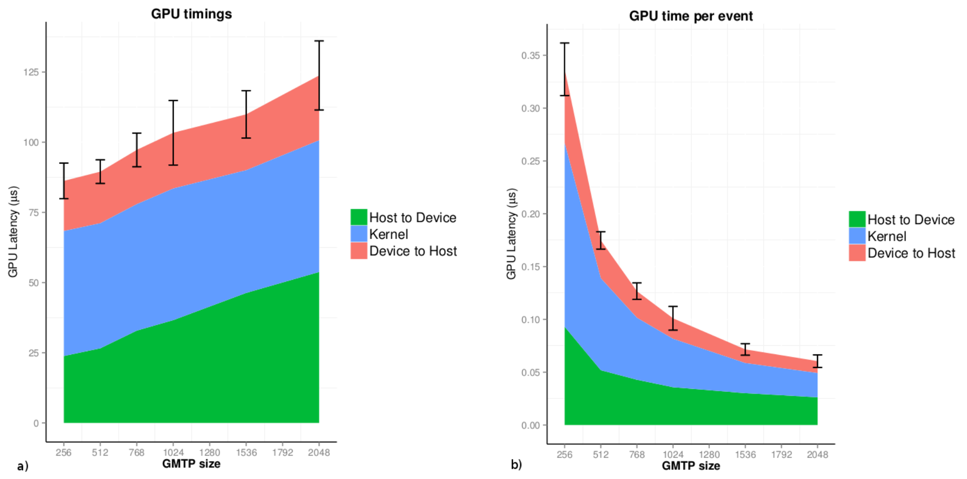

Figure 3: a) Transfer time between host and device and processing time inside the GPU, as a function of GMTP. b) Total time (transfer+processing) per event.

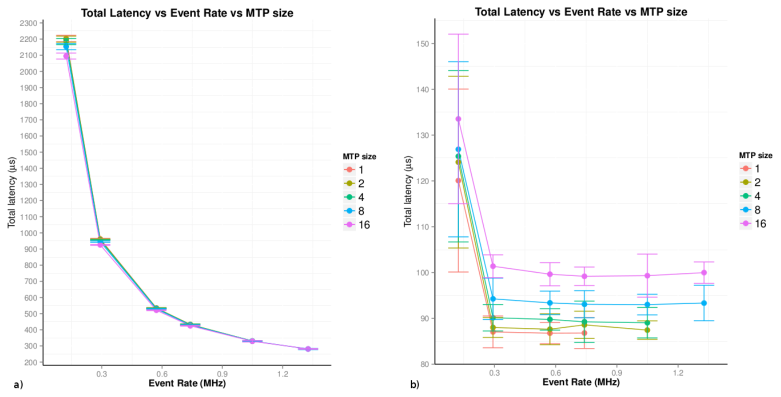

Figure 4: a) Total latency (including transfer times and computing) for a buffer of 256 events. b) Total latency of the system. Gathering time excluded.

buffer).

2.2 NANET

The NaNet board [3] is a customized version of the APEnet+ [2] NIC designed to be integrated in the GPU-based low level trigger system of the NA62 RICH detector (Section 3). Adding to the APEnet+ design the logic to manage a standard GbE interface, NaNet is able to exploit the GPUDirect P2P capabilities of NVIDIA Fermi/Kepler GPUs allowing a hosting PC to directly inject into their memory an UDP input data stream from the detector front-end, with rates compatible with the low latency real-time requirements of the trigger system. The communication tasks are entirely offloaded to a dedicated UDP protocol-handling block directly communicating with the P2P logic: this allows direct data transfer (no data coalescing or staging is performed) with low and predictable latency on the GbE link $\rightarrow$ GPU data path.

Current NaNet implementation provides a single 32-bits wide channel; it achieves 6.4 Gbp/s

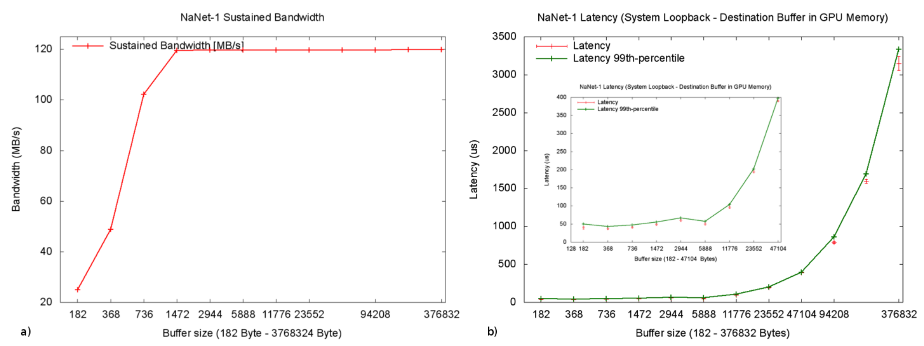

Figure 5: a) Average and 99-percentile NaNet latency vs. buffer size. In the inner box, a magnification of the small packets size section. b) Sustained NaNet bandwidth vs. buffer size.

at the present operating frequency, 6 times greater than what is required for a GbE channel. The embedded ALTERA Nios II subsystem executes the InterNiche TCP/IP stack to setup and tear down UDP packet streams which are processed in hardware at the maximum data rate achievable over the GbE network. In order to limit the Nios II subsystem response jitter, a first hardware optimization was implemented in this release re-designing the interface between NaNet receiver logic and the Nios II requests handler.

The synthesis performed on a Stratix IV achieves the target frequency of 200 MHz (in the current APEnet+ implementation, the Nios II subsystem operates at the same frequency). We use a “system loppback” configuration to measure the system latency and its fluctuations. Figure 5(a) shows the measurements of the whole path latency of the UDP stream (from packets generation to the receiving host). These results match with the NA62 RICH detector (Section 3) requirements for read-out data transmission time budget. In the one GbE link L0 GPU-based trigger processor prototype the sweet spot between latency and throughput is in the region of 70-100 Kb of event data buffer size, corresponding to 1000-1500 events. Saturation of the GbE channel is achieved, with $\sim$ 120 MB/s of sustained bandwidth (see Figure 5(b)). We foresee several improvements on the NaNet design. In particular we will implement a custom logic dedicated to destination address calculation for the data receiving circular buffer in GPU memory, to further improuve total latency. In addition we will design logic units FPGA blocks in order to support low latency data handling of minimum sized packets (less than 200 Bytes). We will also implement a 10 GbE data link interface to sustain the higher bandwidth demands of future read-out systems.

3. NA62

The NA62 experiment at CERN [4] has the goal of measuring the branching ratio of the ultra-rare decay of the charged kaon into a pion and a neutrino-antineutrino pair. The main interest in this decay is linked to the very high precision in the theoretical prediction of its branching ratio, available at the level of few percent, since it is almost free of non-parametric theoretical uncertainties. Because of this, a precise measurement with a sample of 100 events would be a stringent test of the Standard Model, being also highly sensitive to any new physics particle. The trigger is a key system to attain such a result. In its standard implementation, the FPGAs on the readout

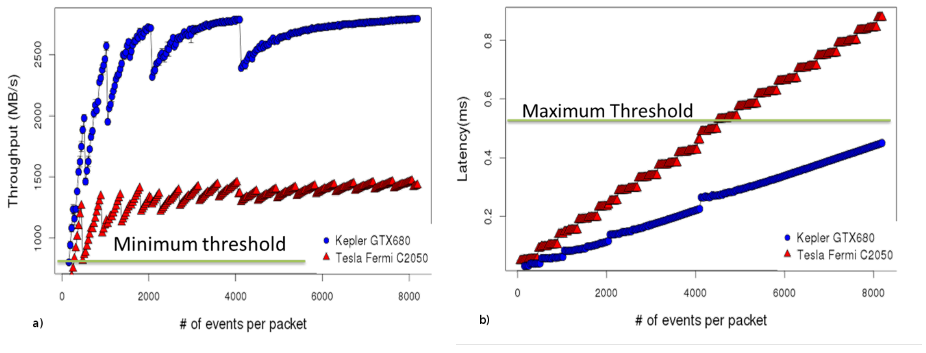

Figure 6: a) Throughput as a function of number of events for last generation GPUs. b) Total latency (including data transfer and computing).

boards of each sub-detector participating to the L0 trigger compute simple trigger primitives. The maximum latency allowed for the synchronous L0 trigger is related to the maximum data storage time available on the data acquisition boards; its value was chosen to be rather large in NA62 (up to 1 ms). Such a time budget allows in principle the use of more complex but slower trigger implementations at this level, i.e. the use of GPUs. This would have the evident benefit of increasing the trigger selectivity for the $K^{\pm} \to \pi^{\pm}\nu\bar{\nu}$ process, thus opening the road to trigger on additional physics processes within at.

As a first use case, we studied the possibility of reconstructing using GPUs the ring-shaped hit patterns in a RICH Cerenkov detector [5]. The use of the information provided by such detector would allow to implement highly selective algorithms.

As described in reference [6] the "math" algorithm, based on a simple coordinate transformation of the hits which reduces the problem to a least square procedure, was found to be the best one in terms of computing throughput (for single rings). This algorithm has been implemented and tested on different GPUs, such as the NVIDIA Tesla C1060, Tesla C2050 and GeForce GTX680 (in increasing order of processing core generation). The computing performance of the C2050 and GTX680 proved to be a factor 4 and 8 higher than that of the C1060. In figure 6(a) we show the computing throughput for these devices as a function of the number of events processed in one batch. The effective computing power is seen to increase with the number of events to be processed in one go. Figure 6(b) shows instead the total latency (for NVIDIA Tesla C2050 and GeForce GTX680 devices), which includes data transfer times to and from the GPU and the kernel execution time. The significant reduction of the latency for the newer GTX680 GPU is due to the faster data transfer thanks to the presence of the gen.3 PCI express bus. Also in this case the maximum latency allowed by the NA62 application is seen to be attainable when a reasonable number of events is processed in one batch.

4. The use of GPUs in high-level triggers

HLT systems, in particular those of LHC experiments, offer a complementary environment with respect to the one discussed so far for NA62 for rate, bandwidth, and latency. Typical values are latencies from ms up to seconds, input rates of few hundred KHz, and bandwidths of hundreds of Gb/s. HLT systems are nowadays implemented as customized software algorithms executed on farms of commodity PCs. The LHC upgrade with the consequent increase of luminosity and pile-up, poses new challenges for the HLT systems in terms of rates, bandwidth and signal selectivity. To exploit more complex algorithms aimed at better performances, higher computing capabilities and new strategies are required. Moreover, given the tendency of the computing industry to move away from the current CPU model towards architectures with high numbers of small cores well suited for vectorial computation, it is becoming urgent to investigate the possibility to implement higher level of parallelism in the HLT software.

The GAP project is studying the deployment of GPUs for the HLT in LHC experiments, using as a study case the ATLAS muon HLT. The ATLAS trigger system is organized in 3 levels [9]. The first-level trigger is built on custom electronics, while the second-level (LVL2) and the event-filter are implemented in software algorithms executed by a farm of about 1600 PCs with different Xeon processors each with 8 to 12 cores. Currently, a first upgrade is foreseen in 2018 [10], when real-time tacking capabilities will also be available, followed by a complete renovation of the trigger and detector systems in 2022. We intend to explore the potential improvements attainable in the near future by deploying GPUs in the ATLAS LVL2 muon trigger algorithms. Such algorithms are now implemented as approximated solutions of complex primitives; the high computing capabilities of GPUs would allow the use of refined algorithms with higher selection efficiency, and thus to maintain the sensitivity to interesting physics signals even at higher luminosity.

Acknowledgment

The GAP project is partially supported by MIUR under grant RBFR12JF2Z "Futuro in ricerca 2012". This work is partially supported by the EU Framework Programme 7 project EURETILE under grant number 247846.

References

[1] *http://www.ntop.org*

[2] R. Ammendola, A. Biagioni, O. Frezza, F. Lo Cicero, A. Lonardo, P. S. Paolucci, D. Rossetti and F. Simula *et al.*, J. Phys. Conf. Ser. **396** (2012) 042059.

[3] A. Lonardo and others,"Building a Low-Latency Real-time GPU-based stream processing system", *http://on-demand.gputechconf.com/gtc/2013/presentations/S3286-Low-La tency-RT-Stream-Processing-System.pdf*

[4] G. Lamanna, J. Phys. Conf. Ser. **335** (2011) 012071.

[5] B. Angelucci, G. Anzivino, C. Avanzini, C. Biino, A. Bizzeti, F. Bucci, A. Cassese and P. Cenci *et al.*, Nucl. Instrum. Meth. A **621** (2010) 205.

[6] G. Collazuol, G. Lamanna, J. Pinzino and M. S. Sozzi, Nucl. Instrum. Meth. A **662** (2012) 49.

[7] *The Atlas Collaboration*, JINST **3** (2008) S08003.

[8] S. Gorbunov *et al.* [ALICE Collaboration], IEEE Trans. Nucl. Sci. **58** (2011) 1845.

[9] *The Atlas Collaboration*, JINST **3** (2008) P08003.

[10] *The Atlas Collaboration*, -CERN-LHCC-2011-012 (2012).