Nuclear Instruments and Methods in Physics Research A 662 (2012) 49–54

Contents lists available at SciVerse ScienceDirect

Nuclear Instruments and Methods in Physics Research A

journal homepage: www.elsevier.com/locate/nima

Fast online triggering in high-energy physics experiments using GPUs

G. Collazuol [a,1], G. Lamanna [b,2], J. Pinzino [b], M.S. Sozzi [b,*]

[a] INFN Sezione di Pisa, L.go Pontecorvo 3, 56127 Pisa, Italy
[b] Department of Physics, University of Pisa, L.go Pontecorvo 3, 56127 Pisa, Italy

ARTICLE INFO

Article history:
Received 24 March 2011
Received in revised form
28 September 2011
Accepted 29 September 2011
Available online 14 October 2011

Keywords:
Triggering
GPUs
Online computing

ABSTRACT

We discuss an approach for using commercial graphic processors (GPUs) at the earliest trigger stages in high-energy physics experiments, and study its implementation on a real trigger system in preparation. Latency and processing rate measurements on several state-of-the-art devices are presented, and potential issues related to processing time jitter and data transfer throughput are discussed. GPUs might act as the missing link to allow present implementations of large DAQ systems to be entirely based on commodity devices.

1. Introduction

The use of commercial Graphic Processing Units (GPUs) for scientific computing purposes grew enormously in recent years and is now rather widespread (see Ref. [1] for an entry point to a vast bibliography). These devices, designed for handling on-screen graphics on Personal Computers (PCs) and manufactured in huge numbers with the video games market as a target, are basically massively parallel SIMD (Single Instruction, Multiple Data) multiprocessors with very fast access to large on-board memory, communicating to the host system via a high-bandwidth standard bus.

The peculiarity of GPUs with respect to general-purpose processors (CPUs) lies in a different architecture, which devotes much more silicon area to actual computing units rather than to control structures, resulting in more specialized devices, which can provide large amounts of raw computing power for highly parallelizable tasks.

In past decades High-Energy Physics (HEP) trigger and Data AcQuisition (DAQ) systems often involved custom-built processors at the leading edge of technology. The explosive spread of mass-market electronic devices changed that framework, and since a long time commercial hardware manufacturers have the lead in the development of the most performing digital computing devices. Furthermore, the costs involved in the development of devices using the latest and fastest silicon technology are now unaffordable for all but the larger hardware manufacturing firms, while the cost for the final user has been constantly decreasing.

As a consequence, the steady trend in the scientific community is towards the use of commercial solutions instead of custom electronic systems for most of the DAQ chain, with the possible exception of the most specific front-end electronics. Similarly, the exponential growth of the computing power of CPUs resulted in a constant shift of the level of data handling at which commodity personal computers are used: if 10 years ago PCs were used only at the last stage of online processing and for data logging, modern experiments often have only the first trigger level based on custom electronics, with all higher level triggers implemented in software, on PC farms (see e.g. Ref. [2] for a recent overview and source of extensive information). The obvious advantages of such trend lie in an optimization of the cost, installation and maintenance issues, as well as the possibility of easy upgrade, since more powerful devices become available every year at the same or lower street price.

Experiments with relatively large number of channels and high event rates, such as those in HEP, so far could not reach the goal of implementing their entire trigger and DAQ system on commodity processors (so-called "triggerless" approach), because the size of the required computer farms would be in most cases impractically large.

In this paper we describe ongoing work in the investigation of the use of GPUs to close the missing gap between front-end electronics and commodity devices.

2. Real-time GPUs?

While the raw power of GPUs is being used in a growing number of scientific computing applications (see e.g. Ref. [3]),

* Corresponding author. Tel.: +39 0502214258; fax: +39 0502214317.
E-mail addresses: gianmaria.collazuol@cern.ch (G. Collazuol), gianluca.lamanna@cern.ch (G. Lamanna), marco.sozzi@df.unipi.it (M.S. Sozzi).
[1] Present address: Department of Physics, University of Padova, via Marzolo 8, 35131 Padova, Italy.
[2] Present address: CERN, Meyrin, Geneve 23, Switzerland.

0168-9002/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.nima.2011.09.057

this is usually done by building large farms, with fast interconnect links among them, to implement powerful, massively parallel supercomputers. Both main GPU manufacturers (NVIDIA [4] and ATI Technologies, now AMD Graphics Products group [5]) developed in recent years programming frameworks (NVIDIA's CUDA and AMD's ATI Stream) which expose the computing power of GPUs for general-purpose tasks, abstracting from the computer-graphic specific processing; a portable standard language also exists (OpenCL [6]), which is well suited to high-level GPU programming but also supports standard CPUs. Computing power (or rather computing power normalized to power consumption) is the only issue in the above mentioned scientific computing applications, while deterministic time response is not crucial; indeed GPUs are not designed for low latency response, since their target application has only to deal with video frames at rates usually below a hundred Hz.

A first-level HEP trigger system must instead handle event rates which can reach into the MHz range, and requires a well defined (maximum) latency: in a deadtimeless system this parameter determines the size of the buffer memories in which detector data must be stored, waiting for a trigger response, before being overwritten. In recent large HEP experiments such latencies range from a few μs to above 10 μs [2], but examples of past experiments with latencies exceeding 100 μs exist [7]. The natural trend for the future is towards an increase in these figures, as allowed by the use of cheap digital memory buffering, and we expect that values in the several hundreds of μs range will be commonplace in future HEP experiments.

The speed of GPUs is increasing at such a fast pace that their intrinsic time latencies are now approaching values compatible to the requirements of low-level HEP trigger system, making such devices an interesting option for moving towards a "triggerless" DAQ system. Furthermore, being raw computing devices running no Operating System (OS), their intrinsic time response is in principle fully predictable.[3] As a matter of fact, the only source of variability in the overall time latency for the completion of a task is related to the control of the GPU and the transfer of code and data to/from it, both normally performed by a host CPU.

The data (and code) transfer is performed on the system interconnect bus, which for modern devices is invariably PCI-express (PCIe): this is based on point-to-point serial links, thus introducing no additional time variability due to bus arbitration between devices, so that the host CPU driving the transactions is again practically the only source of variable latency. Such variability in the time response could be eliminated entirely if GPUs were controlled by a custom PCIe master device emulating the CPU commands, or even by a CPU not running any OS; alternatively, the variability could be limited by running a real-time OS on the controlling CPU. All these possibilities are not easy to implement at present, since information on the low-level control of GPUs is largely undisclosed, and therefore such devices can almost only be used through the software drivers provided for the most widespread OSs; furthermore modern GPUs are rather complex devices with thousands of control registers and a low-level hardware architecture tailored to graphics processing, which can be difficult to exploit for general purposes without a vendor-provided interface layer to the hardware. In any case the use of a GPU in a highly customized and non-standard way would be somewhat contrary to the very idea of adopting standard, cheap, off-the-shelf technology for implementing a high-performance scalable system. We therefore focused on understanding the real-time performances which can be obtained from current GPUs used in a quasi-standard PC environment.

The computation of trigger "primitive" objects might be reduced to either pattern recognition or pattern fitting problems. In both cases a GPU-based trigger architecture allows fast, high-resolution, single-precision floating point arithmetics implementation with great advantages over custom hardware implementations (typically based on FPGAs) in terms of cost, maintenance, scalability, flexibility, ease of programming and debugging.

Moreover, a key point to be appreciated concerning the usefulness of a GPU-based approach for HEP triggering is the fact that whether or not trigger algorithms can be parallelized, the intrinsic nature of a DAQ system, providing a stream of completely independent events, defines another scope for the kind of parallel processing in which GPUs excel.

3. Use case: NA62 RICH ring-finding

A modern medium-scale HEP experiment such as NA62 [8], currently in preparation at CERN, searching for a ultra-rare decay, and thus requiring extremely high particle rates and correspondingly strong trigger rejection, represents the ideal test bench for the investigation of the above approach. The total number of readout channels in the experiment does not exceed 10^5, and the maximum first-level trigger latency was chosen to be 1 ms.

As a first application for the use of GPUs in a first-level trigger we studied the case of single ring identification and fitting in the Ring-Imaging Cherenkov (RICH) detector [9] of the NA62 experiment. The detector (Fig. 1) consists of a 17 m long vessel (4 m diameter) filled with Neon at atmospheric pressure and room temperature as radiator. Cherenkov light produced by charged particles in the 15–35 GeV/c momentum range of interest is focused onto two circular regions of 36 cm radius by a composite mirror. Each region is equipped with $\sim$1000 18 mm diameter Photo-Multipliers (PMs) arranged in a tightly packed honeycomb grid.

In the standard NA62 trigger configuration [10] the RICH will provide the positive condition and the time reference in the first-level trigger (L0), with O(100 ps) resolution per track. The rate of tracks is expected to be around 10 MHz, and the average number of firing PMs per track, measured on a prototype detector, is close to 20 for a π^+ of 25 GeV/c momentum. Additional information on the position and radius of the Cherenkov ring would be useful to implement more selective conditions at this earliest trigger stage.

The goal of the work described in this paper was to test the feasibility and the performances of a GPU-based system for fast ring-finding, in the "real time" conditions required by the online L0 trigger selection of NA62.

We tested a few recent devices by NVIDIA, namely the Tesla C1060 ("T10" architecture, released 2008) and C2050 ("Fermi", released 2009), and by AMD, namely Radeon HD5970 ("Evergreen", released 2009). Nominal parameters for these devices are summarized in Table 1. A cheaper GPU found in desktop PCs (NVIDIA Quadro 600) was also included in the set for comparison.

The processor cores of a GPU are grouped in multi-processor structures (eight processors for NVIDIA "T10", 32 for NVIDIA "Fermi", 80 for ATI "Evergreen"). In each multi-processor the instruction buffer is shared among the cores, all of them executing concurrently the same instructions. The individual cores also share some amount of fast on-chip memory, which is a crucial element for performance considerations.

Five non-iterative algorithms were implemented to test different uses and capabilities of the GPU.[4]

[3] In this context predictable means that the wall clock time required to complete a task can be fully determined from the algorithm and the data on which it operates.

[4] In general non-iterative methods are better suited to the required real-time approach [11].