Real-time heterogeneous stream processing with NaNet in the NA62 experiment

Real-time heterogeneous stream processing with NaNet in the NA62 experiment

R Ammendola[2], M Barbanera[4], A Biagioni[1], P Cretaro[1], O Frezza[1], G Lamanna[3,4], F Lo Cicero[1], A Lonardo[1], M Martinelli[1], E Pastorelli[1], P S Paolucci[1], R Piandani[4], L Pontisso[1], D Rossetti[5], F Simula[1], M Sozzi[3,4], P Valente[1] and P Vicini[1]

[1] INFN Roma,P.le A.Moro,2 - 00185 Roma - Italy
[2] INFN Roma Tor Vergata, Via della Ricerca Scientifica,1 - 00133 Roma - Italy
[3] Pisa University, Largo B.Pontecorvo,3 - 56127 Pisa - Italy
[4] INFN Pisa, largo B.Pontecorvo,3 - 56127 Pisa - Italy
[5] nVIDIA corp., 2701 San Tomas Expressway Santa Clara, CA 95050 - USA

E-mail: luca.pontisso@roma1.infn.it

Abstract. The use of GPUs to implement general purpose computational tasks, known as GPGPU since fifteen years ago, has reached maturity. Applications take advantage of the parallel architectures of these devices in many different domains.

Over the last few years several works have demonstrated the effectiveness of the integration of GPU-based systems in the high level trigger of various HEP experiments. On the other hand, the use of GPUs in the DAQ and low level trigger systems, characterized by stringent real-time constraints, poses several challenges.

In order to achieve such a goal we devised NaNet, a FPGA-based PCI-Express Network Interface Card design capable of direct (zero-copy) data transferring with CPU and GPU (GPUDirect) while online processing incoming and outgoing data streams. The board provides as well support for multiple link technologies (1/10/40GbE and custom ones).

The validity of our approach has been tested in the context of the NA62 CERN experiment, harvesting the computing power of last generation NVIDIA Pascal GPUs and of the FPGA hosted by NaNet to build in real-time refined physics-related primitives for the RICH detector (i.e. the Cerenkov rings parameters) that enable the building of more stringent conditions for data selection in the low level trigger.

1. Introduction

The introduction of heterogeneity into the computing landscape reflects the effort in focusing the diversity of our computation abilities on achieving a global better performance in solving specific tasks. Being the architectures of CPU-GPU-FPGA vastly different, achieving such a collaborative computing requires reckoning with the varied characteristics of all devices.

We devised a heterogeneous approach with the aim of studying the usage of Graphic Processing Units (GPUs) in the Low Level trigger systems of High Energy Physics experiments. Such systems are real-time environments, requiring low and (almost) deterministic latency; their standard implementation is mostly on dedicated hardware (ASICs or FPGA only board).

In order to reduce and control the latency due to data transfer from network to GPU memory, a dedicated FPGA-based Network Interface Card (NIC) has been designed and developed within the INFN-funded project NaNet.

Our integrated system — a GPU-based trigger — consists of a server equipped with an x86-based CPU (managing the minimal software stack for setup of the configuration), hosting the GPU to perform the computing, and the NaNet board, which manages the data transfers and their online processing.

This configuration allows the exploitation of parallel algorithms tailored specifically on the architecture of GPUs to improve the performance in events selection in the Low Level Trigger.

In the following we give details of the system and report on results obtained with such system along with the implemented algorithm.

2. NaNet architecture

The purpose of bridging front-end electronics and the software trigger computing nodes of HEP Experiments requires a low-latency, high-throughput data transport mechanism.

NaNet features a modular design based on a low-latency PCIe RDMA NIC supporting different network link technologies. This layout is functional for a straightforward deployment in multiple scenarios: standard GbE (1000BASE-T) and 10GbE (10GBase-KR) plus a custom 34 Gbps APElink [1] and 2.5 Gbps deterministic latency KM3link [2].

Its key characteristics are i) the management of custom and standard network protocols in hardware, in order to avoid OS jitter effects and guarantee a deterministic behaviour of communication latency while achieving maximum capability of the adopted channel; ii) a processing stage which is able to reorganize data coming from detectors on the fly, in order to improve the efficiency of applications running on computing nodes; iii) a DMA engine performing zero-copy data transfers to or from application memory in order to avoid bounce buffers.

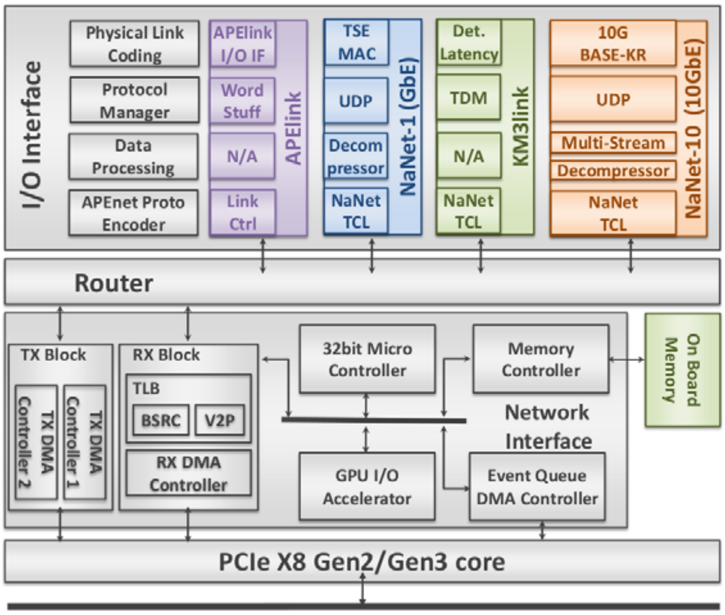


Figure 1: NaNet architecture schematic.

NaNet GPUDirect RDMA capability, inherited from the APEnet+ 3D torus NIC dedicated to HPC systems, extends its application range into the world of general-purpose computing on graphics processing units (GPGPU).

The I/O Interface module performs a 4-stages processing on the data stream: according to the OSI Model, the Physical Link Coding stage implements the channel physical layer (e.g. 10GBase-KR) while the Protocol Manager stage handles data/network/transport layers (e.g. UDP); the Data Processing stage implements application-dependent manipulation on data streams (e.g. compression/decompression) while the APEnet Protocol Encoder performs protocol adaptation. This is carried out encapsulating inbound payload data into the APElink packet protocol (used in the inner NaNet logic) and decapsulating outbound APElink packets before re-encapsulating their payload into the output channel transport protocol.

2.1. Software stack

The NaNet board is capable of writing data coming from a network source directly into the GPU memory through the PCIe bus, using GPUDirect RDMA (see figure 2, left).

Following such write, a "receiving done" event is also DMA-written in a memory region called "event queue" and trapped in kernel-space by a device driver which sends a signal to the user application. This latter launches the consuming application on GPU.

Once the processing is completed, the results can eventually be sent via NaNet board to a remote device. Data are DMA-read directly from GPU memory; the kernel device driver — invoked by the user application on the host machine — instructs the NIC by filling a "descriptor" into a dedicated DMA-accessible memory region called "TX ring".

The presence of a new descriptor is notified over PCIe to NaNet by writing on a doorbell register so that the board can issue a "TX done" completion event in the "event queue".

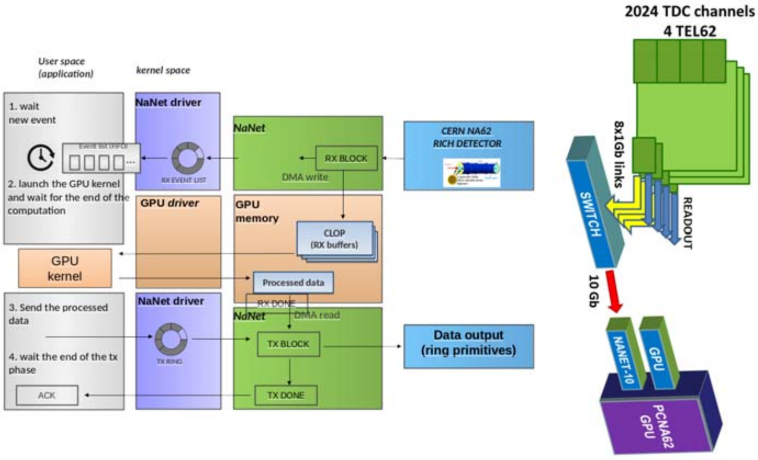


Figure 2: NaNet software stack (left) and GPU-RICH readout system (right).

2.2. NaNet-10

NaNet-10 is a PCIe Gen2/3 x8 network adapter implemented on the Terasic DE5-net board equipped with an Altera Stratix V FPGA featuring four SFP+ cages.

The board features a zero-latency full UDP/IP network stack on the 10GbE channels, as well as a high throughput PCIe core towards the host with GPUDirect capabilities [3].

3. Physics case: heterogeneous processing in the NA62 Low Level Trigger

The NA62 experiment at CERN is designed to measure the branching ratio of the ultra-rare decay of the charged kaon into a pion plus a neutrino-antineutrino pair.

The sub-detectors are located along the trajectory of the K^+ beam with the purpose of identifying decay particles and to measure their momentum and energy.

Due to an expected ∼10 MHz rate of these decays, a cascade of three trigger levels is needed to reduce the rate by three orders of magnitude to manage the stream of raw data [4].

The low-level trigger (L0) performs the early selection of events with a time budget of $\sim$1 ms. It is a synchronous real-time system implemented in hardware through FPGA boards.

The GPU-based processing stage is inset between the RICH readout and the L0 Trigger Processor (L0TP) with the task of computing in real-time physics-related primitives (i.e. centers and radii of Čerenkov ring patterns on the photomultipliers arrays), in order to improve the low level trigger discrimination capability.

Data from the detector photomultipliers (PMTs) are collected by four readout boards (TEL62) sending primitives to GPU as UDP datagrams on a dedicated GbE link (see figure 2, right) connected to a multiport GbE/10GbE switch. Packets are then received over the NaNet-10 10GbE channel. The onboard FPGA performs zero-copy of the raw data onto the GPU after decompressing and coalescing the event fragments scattered among the four streams.

Data are gathered and arranged in the GPU memory as a Circular List Of Persistent buffers (CLOP), according to a configurable time window which must be shorter than the total processing time to avert the overwriting of the buffers before they are consumed.

A system such as this, displaying improved processing capabilities and flexibility and based on different computing units, can be regarded as a heterogeneous "smart detector".

3.1. Overview of the histogram-based algorithm

In order to identify the ring patterns in the PMT hit map, we developed a specific pattern recognition histogram-based algorithm to leverage the massively parallel architecture of the GPU. Obviously, working in a real-time environment, this must be fast enough to cope with the input events rate while remaining within the L0TP 1 ms time budget.

Another constraint on the design is exerted by the need to be seedless, as data coming from other detectors are not available at this stage.

Preliminary to the ring reconstruction stage there is the indexing of events gathered and merged by NaNet into a CLOP. Many threads launched by a single Event Finder kernel look for event delimiter patterns in the buffer and the resulting vector of memory addresses is then passed to the processing kernel.

On a general level, the computing is arranged so that each event is processed by a whole block of threads, leveraging the GPU per-block shared memory faster than the global one.

Taking into account that $\sim 95\%$ of K^+ decays have only one charged particle in the final state, resulting in only one ring in the RICH detector, the overall processing of events is sped up using the histogram-based algorithm only for multi-rings events while a faster single-ring fit algorithm like Crawford [5] (a least square method, since iterative ones are not easily implemented in a parallel way) is employed for the most frequent cases.

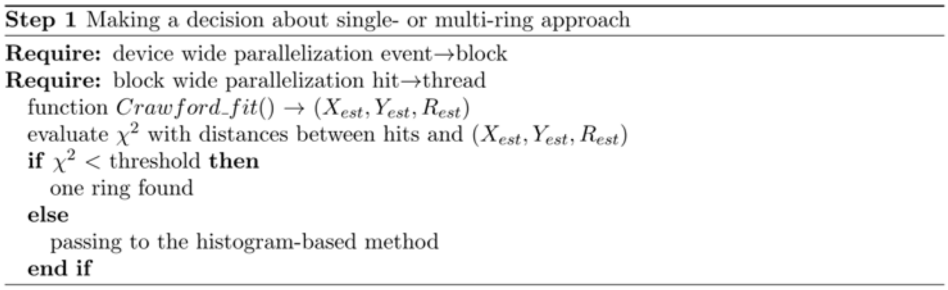
The decision about which is the right one to use is based on a χ^2 test involving distances between the hits.

At a block level (i.e. for each event), every thread holds the coordinates of a single PMT hit. A χ^2 test between center and radius estimated with the Crawford algorithm using hits and their center of gravity is performed. If the revenue is less than a threshold value, the hypothesis of a single ring is accepted. If the test fails, the event is passed to the histogram-based method (details in figure 3).

A grid of 8×8 squares is established, associating each element to a thread for a total of 64 of them. For each square, the distances between its center and every hit are computed. These values are inserted into a histogram. The bins with the higher number of occurrences (above a threshold) are selected.

A comparison between the values of neighbouring grid elements makes us choose the local maxima.

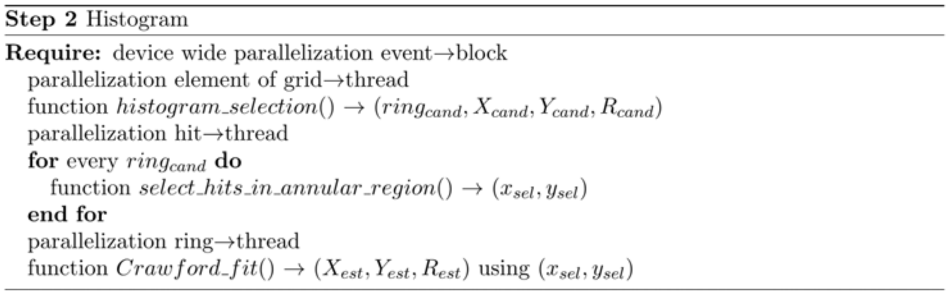
These are the centers for the candidate rings. For each one we select the hits whose distances fall within an annular area (defined by maximum and minimum physics-related radii). For each



```
Step 1 Making a decision about single- or multi-ring approach
Require: device wide parallelization event→block
Require: block wide parallelization hit→thread
  function $Crawford\_fit() \rightarrow (X_{est}, Y_{est}, R_{est})$
  evaluate $\chi^2$ with distances between hits and $(X_{est}, Y_{est}, R_{est})$
  if $\chi^2 <$ threshold then
    one ring found
  else
    passing to the histogram-based method
  end if
```

Figure 3: Illustrative code dealing with different paths in processing single or multi-ring events.

set of hits (associated to a candidate ring) we apply again the Crawford algorithm to evaluate center and radius with a better accuracy — see figure 4.



```
Step 2 Histogram
Require: device wide parallelization event→block
  parallelization element of grid→thread
  function $histogram\_selection() \rightarrow (ring_{cand}, X_{cand}, Y_{cand}, R_{cand})$
  parallelization hit→thread
  for every $ring_{cand}$ do
    function $select\_hits\_in\_annular\_region() \rightarrow (x_{sel}, y_{sel})$
  end for
  parallelization ring→thread
  function $Crawford\_fit() \rightarrow (X_{est}, Y_{est}, R_{est})$ using $(x_{sel}, y_{sel})$
```

Figure 4: Schematic about parallel implementation of the histogram algorithm.

4. Experimental results

The whole GPU-RICH system equipped with the NaNet-10 board is currently installed at the NA62 site (working in a completely parasitic mode with respect to standard trigger), collecting data during the 2017 run.

The testbed is made up of a HP2920 switch, a NaNet-10 PCIe board plugged into a server made of X9DRG-QF dual-socket motherboard populated with Intel Xeon E5-2620 @2.00 GHz CPUs (i.e. Ivy Bridge architecture), 32 GB of DDR3 RAM, and a Kepler-class NVIDIA K20c GPU (until October, a NVIDIA Pascal P100 was employed thereafter).

Latencies of the stages corresponding to GPU processing using the K20c (event indexing and ring reconstruction) and sending UDP packets with the results to the L0TP are shown in figure 5. Data were taken during single bursts with beam intensity 18×10^{11}pps and a gathering time for NaNet-10 of 350 μs (dashed line on plots). The red area represents the latency of event indexing in the CLOP buffer (almost constant), the cyan area shows the ring reconstruction kernel latency and the blue is the sending stage.

On the left, data acquisition is performed processing only every eighth event (i.e. downscaling factor of 8) and the overall latency is below the time limit posed by gathering time. The plot on the right displays instead results with a downscaling factor of 4. The ring reconstruction time increases with the number of processed events and in this case the total latency exceeds constantly the time budget. This is an unwanted situation, leading to loss or corruption of data.

Having identified the ring-fitter kernel as the culprit for this condition, we investigated the possibility of increasing the computing speed with the availability of the new Pascal NVIDIA architecture. Our tests show (figure 6) a speedup factor of 5 on processing buffers with 2000 events: from ∼500 ns/evt on K20c to ∼100 ns on P100. This suggests that with a new GPU the system downscaling can comply with the time budget request also in a no downscaling mode.

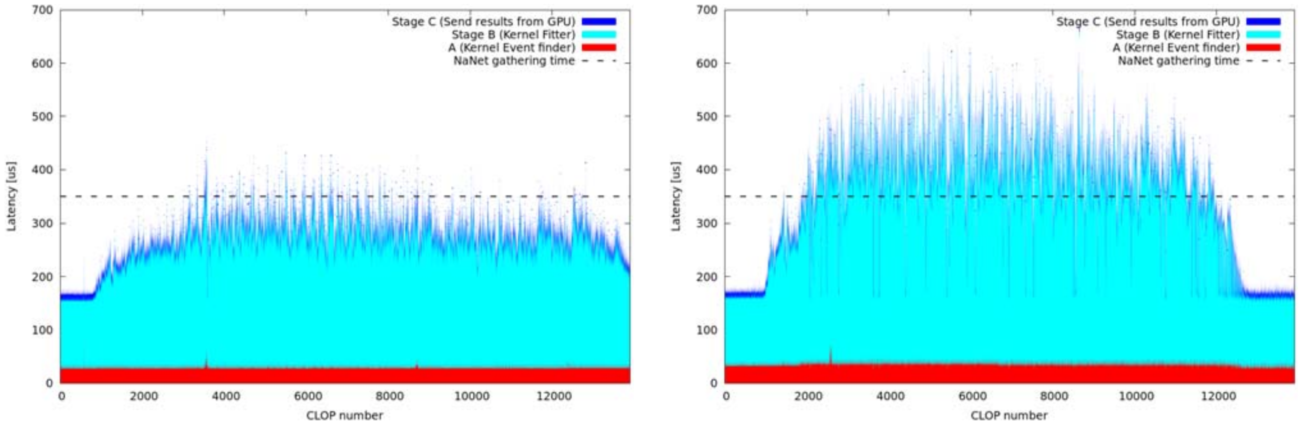


Figure 5: GPU processing latency.

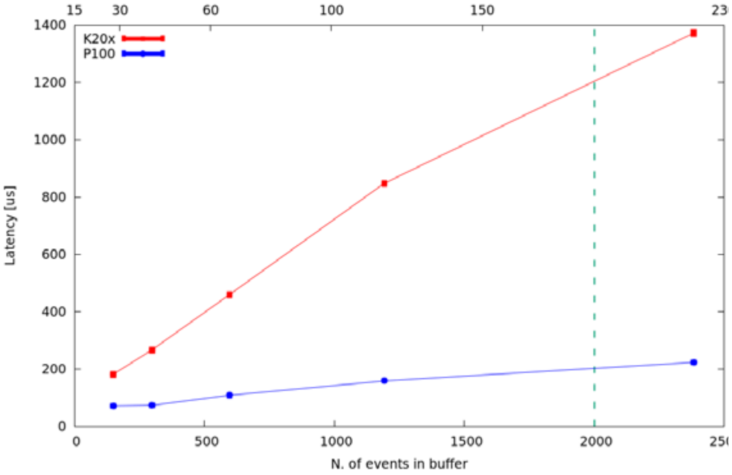


Figure 6: Ring reconstruction latency: NVIDIA Pascal (P100) vs K20c.

5. Conclusions

The widespread availability of heterogeneous architectures led us to assess their potential in the challenging real-time environment of low level trigger systems for High Energy Physics experiments. The whole NaNet-based framework turns out to be able to cope with the strict requirements for the NA62 RICH Low Level Trigger with respect to events rate and time budget.

The hurdle of the latency in the ring reconstruction might be overcome, according to our tests, with a GPU upgrade. We plan to install the new NVIDIA P100 in order to keep the process latency below the threshold without any downscaling factor and to prove the ability of the NaNet-based system to sustain the full detector throughput.

References

[1] R. Ammendola, A. Biagioni, O. Frezza, A. Lonardo, F. Lo Cicero, P. Paolucci, D. Rossetti, F. Simula, L. Tosoratto, and P. Vicini, "APEnet+ 34 Gbps Data Transmission System and Custom Transmission Logic," *Journal of Instrumentation*, vol. 8, no. 12, p. C12022, 2013.

[2] A. Aloisio, F. Ameli, A. D'Amico, R. Giordano, V. Izzo, and F. Simeone, "The NEMO experiment data acquisition and timing distribution systems," in *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2011 IEEE*, pp. 147–152, Oct 2011.

[3] R. Ammendola et al., "Nanet-10: a 10GbE network interface card for the GPU-based low-level trigger of the NA62 RICH detector.," *Journal of Instrumentation*, vol. 11, no. 03, p. C03030, 2016.

[4] B. Angelucci et al., "The fpga based trigger and data acquisition system for the cern na62 experiment," *Journal of Instrumentation*, vol. 9, no. 01, p. C01055, 2014.

[5] J. Crawford, "A non-iterative method for fitting circular arcs to measured points," *Nuclear Instruments and Methods in Physics Research*, vol. 211, no. 1, pp. 223 – 225, 1983.