Parallel Neutrino Triggers using GPUs for an underwater telescope

Bachir Bouhadef [1], *Mauro Morganti*[1,2], *Giuseppe Terreni*[1]
for the KM3Net-It Collaboration

[1] INFN, Sezione di Pisa, Polo Fibonacci, Largo B. Pontecorvo 3, 56127 Pisa,Italy
[2]Accademia Navale di Livorno, viale Italia 72, 57100 Livorno, Italy

DOI: will be assigned

Graphics Processing Units are high performance co-processors originally intended to improve the use and the acceleration of computer graphics applications. Because of their performance, researchers have extended their use beyond the computer graphics scope. We have investigate the possibility of implementing and speeding up online neutrino trigger algorithms in the KM3Net-It experiment using a CPU-GPU system. The results of a neutrino trigger simulation on NEMO Phase II tower and a KM3-It 14 floors Tower are reported.

1 Introduction

A neutrino telescope is a tool used to increase our knowledge and to answer fundamental questions about the Universe. Following the success of the IceCube experiment [1], which is a km^3 size telescope in the ice at south pole, and of the ANTARES experiment [2], an underwater telescope with a volume of 0.4km^3, the European scientific community is going to construct a neutrino telescope similar to but larger than IceCube called Km3Net in the Mediterranean Sea. The NEMO [3] and NESTOR [4] are R&D experiments for the same purpose. All these optical telescopes use a Photomultiplier Tube (PMT) or a group of it as a Detection Unit (DU). The NEMO collaboration have already deployed a tower of 32 single PMT DUs. For the much larger Km3Net Telescope, thousands of DUs will be used to detect the muons passage produced by undersea neutrino interactions. This large number of sensors will lead to a huge amount of data to be filtered by any trigger algorithm. In the case of the ANTARES telescope the amount of data acquired in a second is 0.3-0.5 GB, and many CPUs are used for such a task. The general strategy of data analysis for an online trigger is that each CPU works on a Time Slice of the data coming from the underwater telescope [3], and this is a kind of parallel work. In the present work, we describe the study of using a Graphical Processor Unit (GPU) to implement a trigger algorithm and to simplify it. In addition, the use of GPU-CPU leads to power, hardware and time saving. The parallel version of the trigger algorithm is shown to be suitable for an online muon track selection, and was tested on simulated data of the NEMO Towers of 32 and 84 (KM3Net-It tower) single PMT DUs.

2 DAQ system of NEMO Phase II

During March 2013, the NEMO collaboration have deployed a tower (See Fig. 1) at the Capo Passero site, south east of Sicily island [3]. Since than, it was taking data until the power shutdown in the beginning of August 2014 for upgrading. The tower is 450m high, and it is composed of 8 floors with 4 DUs each, and two hydrophones. The distance between floors is 40m, where the length of the floor is 6m. The tower is kept up thanks to an appropriate buoyancy on its top. The tower is hosted at depth 3400m nearly 100km off the Capo Passero harbour, each floor equipped with a Sea Floor Control Module (SFCM) that collects data streams from the four PMTs as well as the two hydrophones and sends it to its twin card on-shore called Earth FCM (EFCM), via Electro-Optical-Cable (EOC) which permits 2.5Gbs of bandwidth. During data acquisition and low bioluminescence activity we have a rate of 50-55kHz by PMT. The PMT hit size is 28 bytes in average, and a total 20% of the bandwidth is used.

The data streams from all EFCMs are grouped in to two streams and routed to the first stage of CPUs data processing called Hit Managers (HM) via Gbit link, which means the tower has two HMs as in Figure 2. Each HM merges together PMTs data streams and divides in consecutive time intervals of 200ms called time-slices. All time slices from HMs within same time interval are sent to one Trigger CPU (TCPU) where the trigger will be applied, and successive time slices are sent to different TCPUs. A detailed description of the DAQ system can be found in [3]. Once trigger conditions are satisfied, a $6\mu s$ time window of the tower data (centred at the time of the trigger seed) are sent to Event Manager to be save for a more off-line filtering and muon track reconstruction.

3 Muon trigger strategy

A muon passing through water, faster than the speed of light at that medium produces Cherenkov photons. These photons have an angle θ_C with respect to the trajectory of the muon are detected by distributed DUs. The arrival times of muon Cherenkov photons at PMTs are correlated in time and space which is not the case of the background photons (mostly are photons generated by ^{40}K decays). Hence, looking for coincidences of multiple PMTs within a certain time window will reduce the background rate.

In fact the time difference of the arrival time of photons generated by a muon are distributed in maximum $3\mu s$ interval, and a time-space correlation between PMTs hits within a time interval less than $3\mu s$ (time needed for a muon to traverse the detector) can be used to filter out the background hits. In addition to the time-space correlation, a hit charge over threshold trigger can be applied on all hits. These type of triggers can be parallelized because are applied on data streams independently.

Even though these triggers are simple, the amount of background data hits in NEMO phase II, is about 1.7 Mhits/s/tower (with respect to a roughly a few hits/s of a muon track) and it will be 4.6 Mhits/s for 14 floors Tower.

In addition to these standard triggers, we have studied a level 0 trigger, easy to implement on GPU, based on N hits in a fixed Time Windows (we chose N7TW1000 with N=7 and TW=1000ns) to reduce the background rate [5], instead of looking first for all possible time coincidences and charge over threshold which is time consuming. After this level 0 trigger, we apply the other triggers, and the background rate is drastically reduced.

4 CPU-GPU

The aim of our work is to replace the TCPU by a TCPU-GPU system (TGPU). The work in the TGPU is done into three steps: first prepare 1 second of data to be sent to GPU, then send data to GPU for trigger selection, and last save the selected events.

Every PMT hit in the NEMO Phase II tower contains an header, with the GPS time and geometry information, followed by a sampled charge waveform with a variable size. To apply triggers in GPU, the raw hits are converted to a fixed size structure that contains all needed information for trigger algorithms: charge, time, DU identification (DU_ID), trigger flags and rates. We have optimized the work in CPU and GPU to minimize the trigger searching time.

The main work of CPU is to convert 5 consecutive time slices of 200ms to an unique time slice of one second to be sent to GPU (one fixed size memory buffer). This is done by running 5 CPU threads, and each thread will fill dedicated memory zone. A CPU thread will convert the 200ms time slice to N time slices to be used by GPU threads. The number of threads N is chosen to have in average 100 hits per thread at a nominal rate of 50kHz/PMT, and the total number of threads (NTHRD) in GPU will be NTHRD=5xN.

In our code, we have also take into account the edge effect between threads, in such way that when the trigger algorithm reaches the last hit of the current thread, it proceeds on few hits from the next time slice. The data structure to be sent to GPU is 1 second time interval, and its size takes into account the maximum rate of each PMT(5 times the nominal rate). Hence, in the presence of bioluminescence at two PMTs for example at rate of 1Mhits/s, and assuming a nominal hits number per thread of 100 (55 kHz/PMT), we expect an increase of 40%(140 hits/thread) of the number of hits.

The work in GPU is done in two steps, first we sort the hits in time order using classical sorting methods (Shell, Quick, and Merge sort algorithms), and than we apply the all needed trigger algorithms. Figure 3 shows the performance of Quicksort and Shellsort algorithms on a uniformly distributed time values. The time values are in structure of size 1 and 7 floats (one float is used for the time value), clearly the Quicksort algorithm shows a good performance, and the measured times for all cases remain below 100ms for 100 elements per thread.

In the case of NEMO tower data, the Shellsort algorithm shows a better performance over the others less than 50ms, the reason is by the fact that our data within a GPU thread are not completely random, but are time ordered for single PMT within a GPU thread.

After time sorting we apply the following trigger algorithms:

- time-space correlation (for NEMO Phase II tower):
  - looking for 7 hits in TW = 1000ns (N7TW1000): $hit[i].time - hit[i+7].time < TW$
  - looking for 7 hits from different PMTs within TW = 1000ns (DN7TW1000)
  - Simple Coincidence (SC): a coincidence between two hits occurred on two adjacent PMTs within 20ns.
  - Floor Coincidence (FC): a coincidence between two hits occurred on two PMTs in same floor within 100ns.
  - Adjacent Floor Coincidence (AFC): a coincidence between two hits occurred on two PMTs located at two consecutive floors within 250ns .
- charge threshold (QTRIG) looking for charge hit over a threshold: $hit[i].Charge > QTRIG$

- combination between the above triggers.

The N7TW1000 and DN7TW1000 have the same efficiency for muon track selection, however DN7TW1000 is more efficient in the presence of bioluminescence activities for background reduction.

To select muon events we combine between these trigger seeds. Once the candidate hits satisfied the trigger seeds all hits within $\pm 3\mu s$ are tagged to be saved with respect to our trigger seeds. After that, the data are sent back to CPU to save the tagged hits.

Even though there is a difference in selected events rate between applying N7TW1000 before or after time coincidences, and applying time coincidences after N7TW1000, we have chosen N7TW1000 as level 0 trigger because the trigger time searching is less and the muon track selection efficiency remains the same [5].

5 Results and perspectives

For our simulation we have used Tesla20c50 (448 CUDA cores) and GTX Titan (2688 CUDA cores), in addition to 2 PCs one was used as a Hit Manager for time slice sending and the other PC was used with GPU cards for trigger algorithms. The measured time of the CPU to prepare 1 second of data ranges from 200ms (for 32 PMTs tower) to 300ms (for 84 PMTs tower), the CPU-GPU data memory transfer time is included.

Our first aim was to see whether the TGPU can cope with data streaming and the online triggers for muon track selection within left time (less 700ms). First step was to simulate the actual NEMO phase II tower with 32 DUs at rate 55kHz and we apply the following triggers: charge trigger, SC, FC, AFC, N7TW1000 and N7TW1000 AND [(SC OR FC) AND AFC]; the last is used to tag the muon track candidate. In the case of KM3Net-It tower of 84 DUs we have used N7TW500 (the inter-floor distance is 20m instead of 40m).

Table (1) shows measured Tesla-GPU times for 1 second data from 32 and 84 PMTs using 8x10^3, 2x10^4 and 4x10^4 threads, respectively. By adding 200-300ms of the CPU time for 1 second time slice preparation to GPU time trigger, we verify that Tesla 20c50 GPU card is able to cope with online trigger algorithms of NEMO Phase II tower as well as of KM3Net-It tower (with GPU threads >= 20000).

We have also compared the performance of our triggers in Tesla20c50 and GTX Titan. We have tested only the GPU work without including the 1 second time slice preparing as well as data memory transfer. The results are shown in Table (2). For a sufficient number of GPU threads both GPU cards can handle the online trigger of 84 PMTs data.

The work done in HMs can also be included in CPU-GPU work, given that we were using only 40% of the CPU resources, and leads us to group the work of HMs and TCPUs in a TGPU system, with adequate network structure.

Our conclusion is that both GPU cards can be used in the TGPU system for the online muons trigger. We propose a new DAQ system based on TGPU structures (Figure 4) for the 8 towers of KM3Net-It, where each TGPU (1-8) looks for all trigger seeds in 1 second of raw data of corresponding tower and send all trigger seeds to the TCPU for muon tracks selection. Once the TCPU selected the candidate trigger seeds for muon tracks, send back the corresponding times to all TGPUs to send their data time window to Event Manger, and free the corresponding memory buffer.

This new DAQ of CPU-GPU is a huge simplification of the classical CPU DAQ system and can be used for both online trigger and even the muon track reconstruction which is our next step.

NTHRDs	32 PMTs	84 PMTs
8000	250	950
20000	230	600
40000	190	500

Table 1: Measured time triggers (ms) in Tesla 20c50 for 1 second of Nemo towers.

NTHRDs	32 PMTs	84 PMTs
8000	160 / 90	500 / 450
20000	100 / 50	200 / 140
40000	50 / 45	150 / 110

Table 2: Measured time triggers (ms) in Tesla20c50 / GTX TITAN for 1 second of NEMO towers.

References

[1] A. Karle *et al.*, arXiv:hep-ex/14014496 (2014).

[2] S. Mangano *et al.*, arXiv:astro-ph/13054502 (2013).

[3] T. Chiarusi *et al.*, JINST 9 C03045 (2014).

[4] P.A. Rapidis, Nucl. Instr. and Meth. A, 602 (2009).

[5] B. Bouhadef *et al.*, *Trigger Study for KM3Net-Italia*, 6th Very Large Volume Neutrino Telescope Workshop, Stockholm, Sweden (2013).

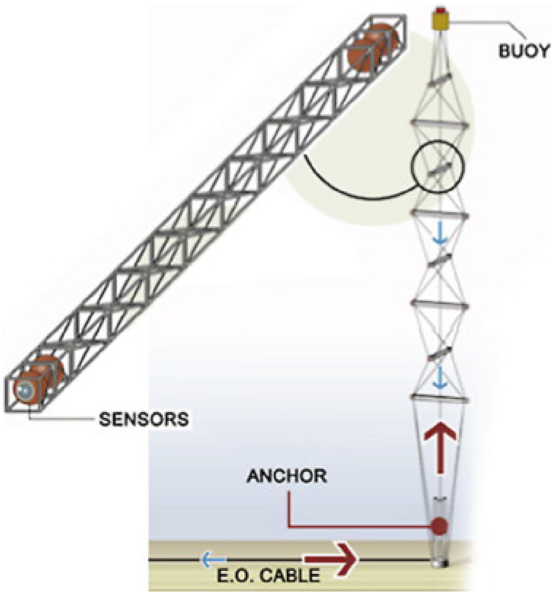

Figure 1: NEMO Tower.

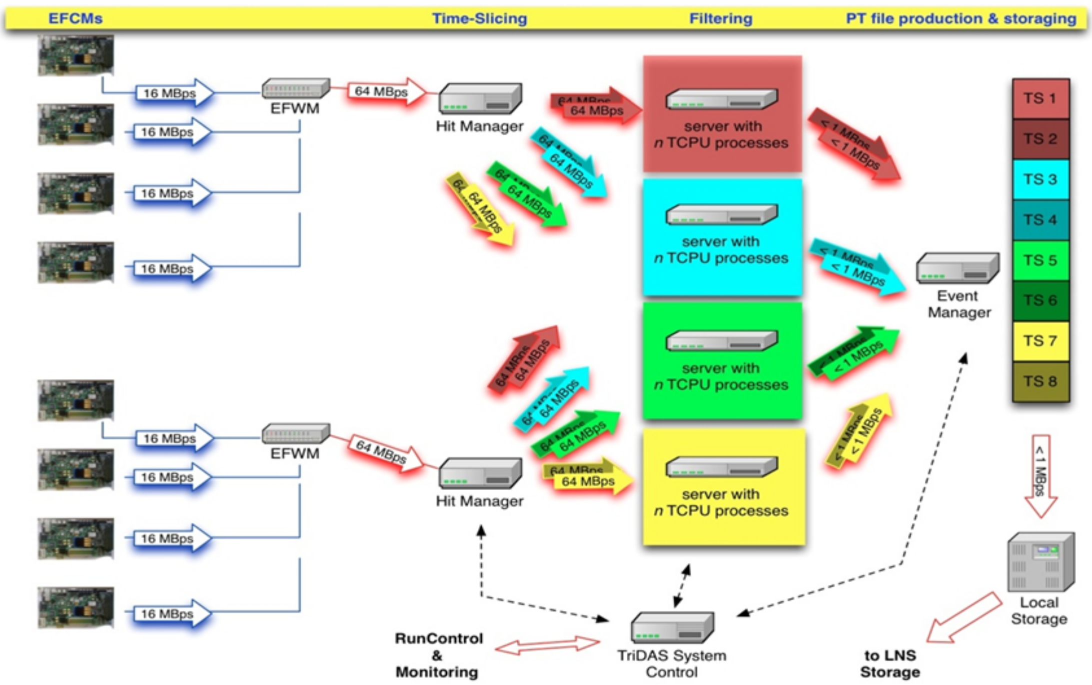

Figure 2: DAQ System for NEMO Phase II and the Trigger scheme.

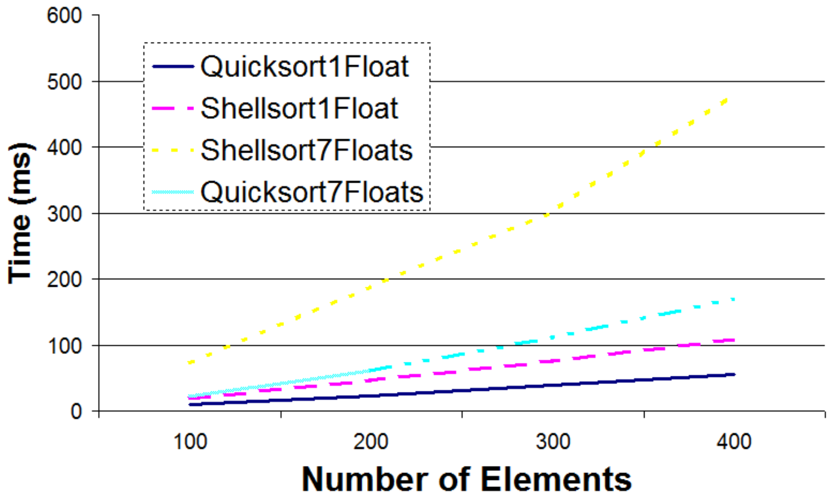

Figure 3: Measured sorting time (ms).

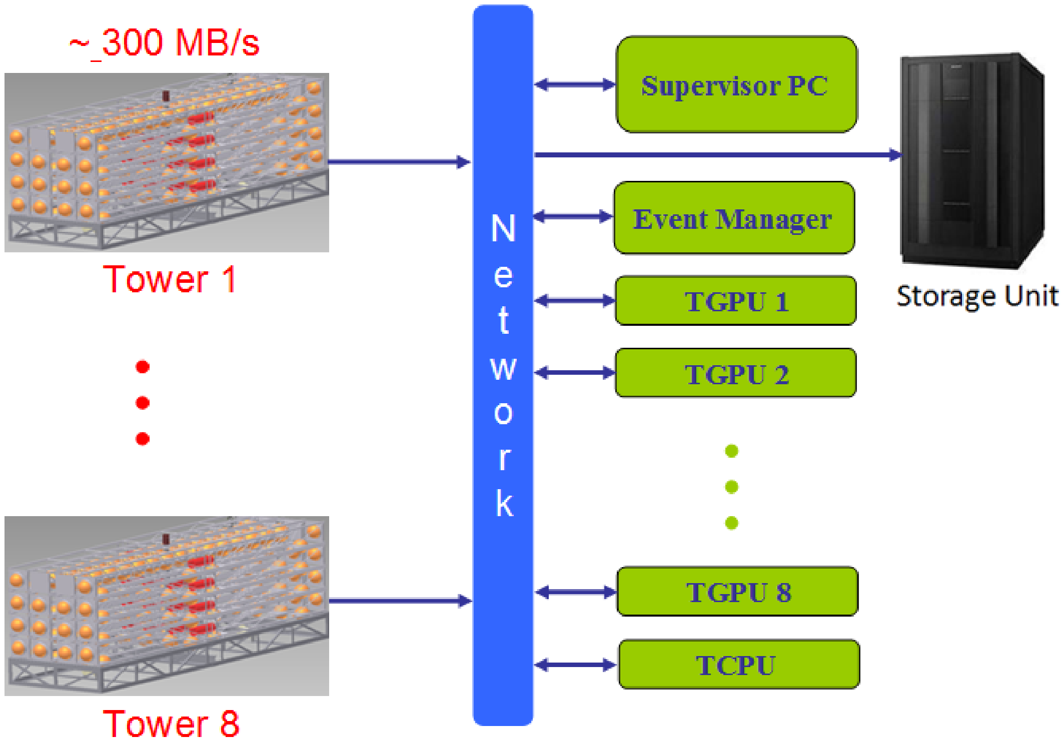

Figure 4: Proposed DAQ based on CPU-GPU structure.