GPUs for Parallel Trigger Implementation For Muon Detection

Bachir Bouhadef, Mauro Morganti, Antonio Marinelli

VLVnT11 - Very Large Volume Neutrino Telescope Workshop 2011.

INFN Pisa & Physics Department of Pisa

Outlines

The aim of this presentation is :

- **Why GPUs.**
- ***Showing a possibility of using a system CPU-GPU (NVIDIA) for an online muon-track selection.***
- ***Proposing a method for paralyzing the online trigger software NEMO-II Tower.***
- ***Test the GPU-online trigger in the NEMO-II DAQ structure.***
- ***Proposion KM3NeT Tower Trigger data handling test.***

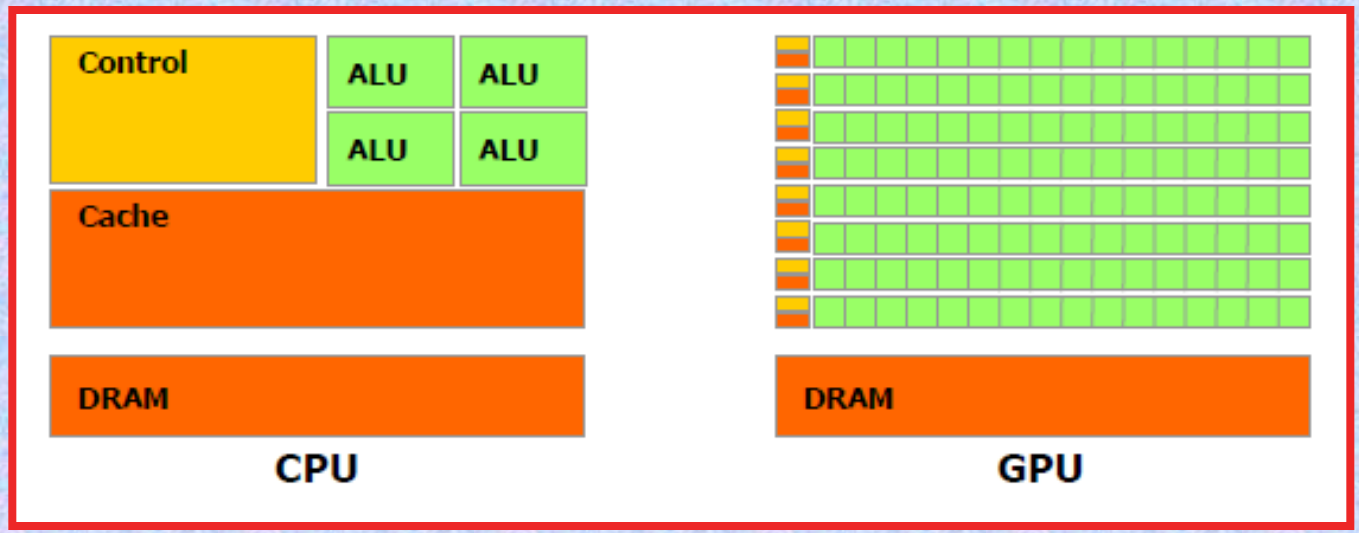
Why GPU ?

448 core CUDA

Data Transfert using PCI Express Gen 2.0. (~5GB/s up, 4.5GB/s down)

We gain the hardware space.

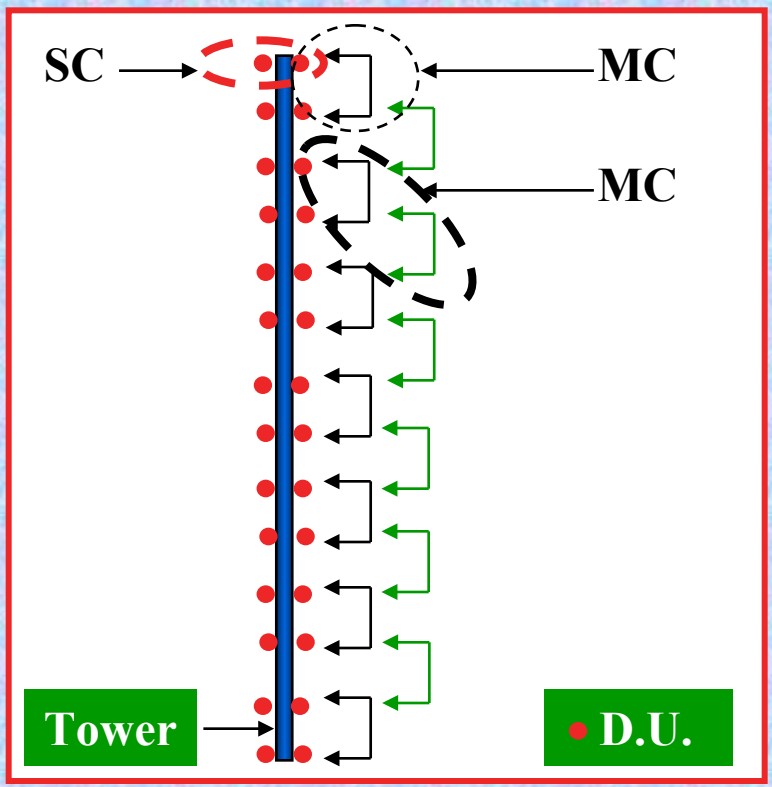
GPU versus CPU

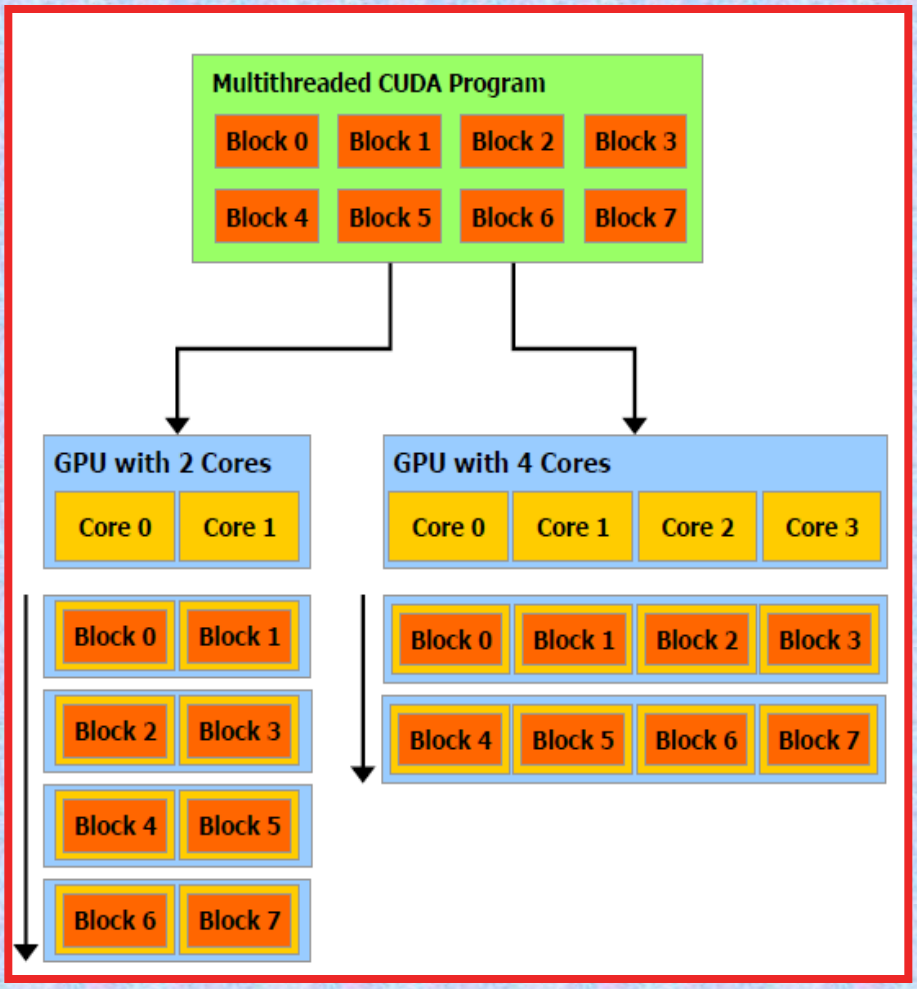


GPU devotes more transistors to data processing,
GPUs are especially well-suited to address problems that can be expressed as data-parallel computations.

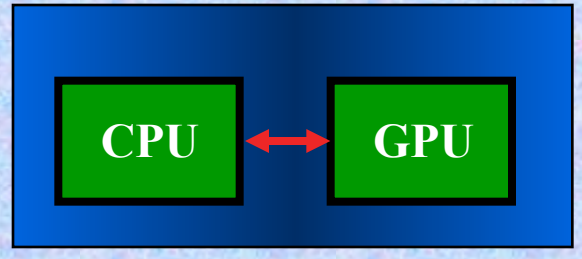
Scalable Programming Model

A GPUs use blocks and threads for parallel programming

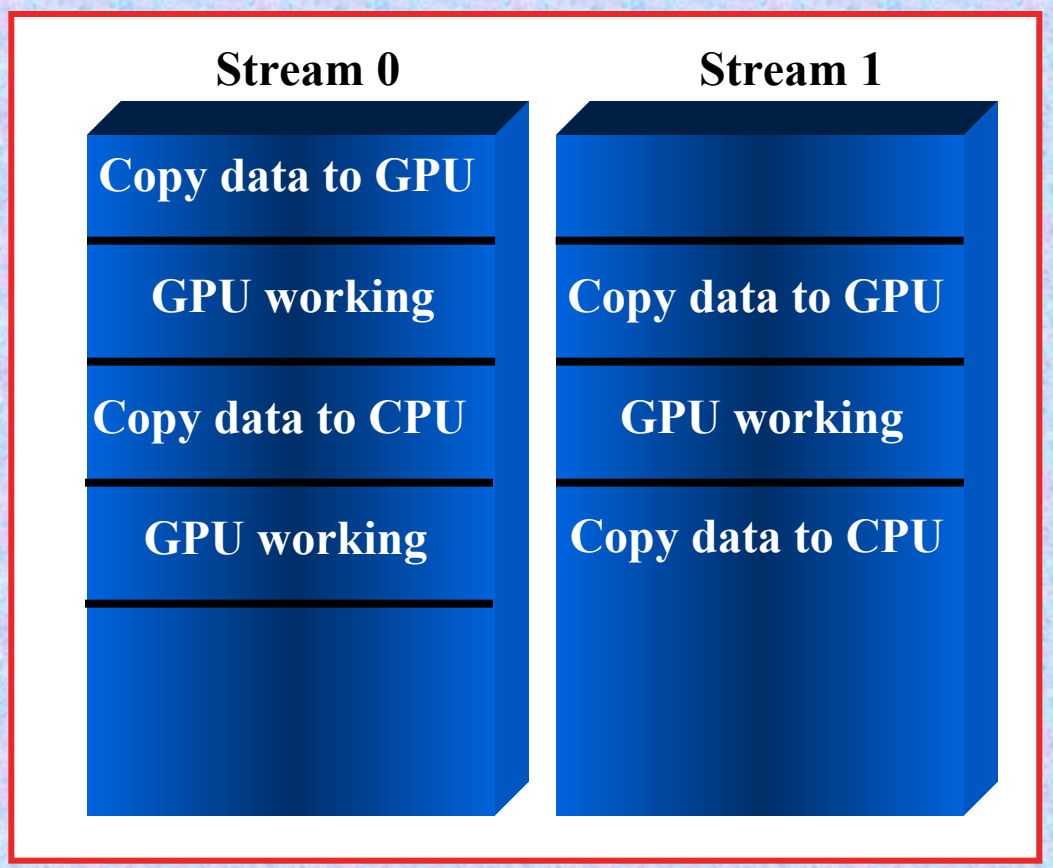




Streams in GPU



Functions in GPU are called kernels

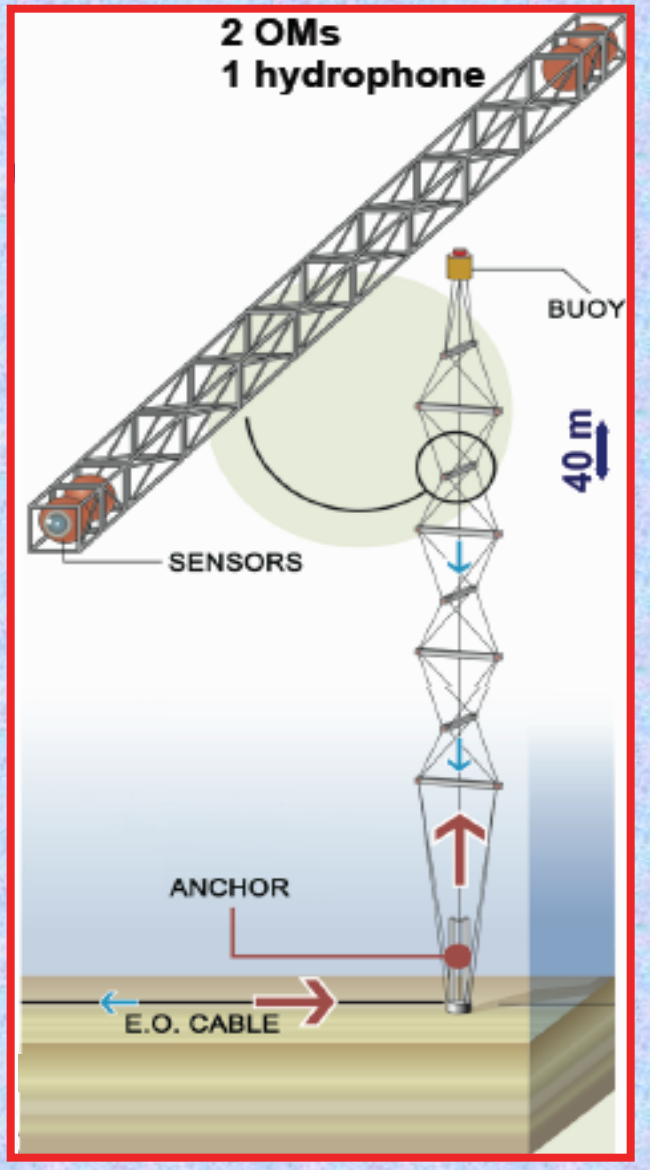


In present simulation we benefit from the scalability and the streaming option of the GPU.

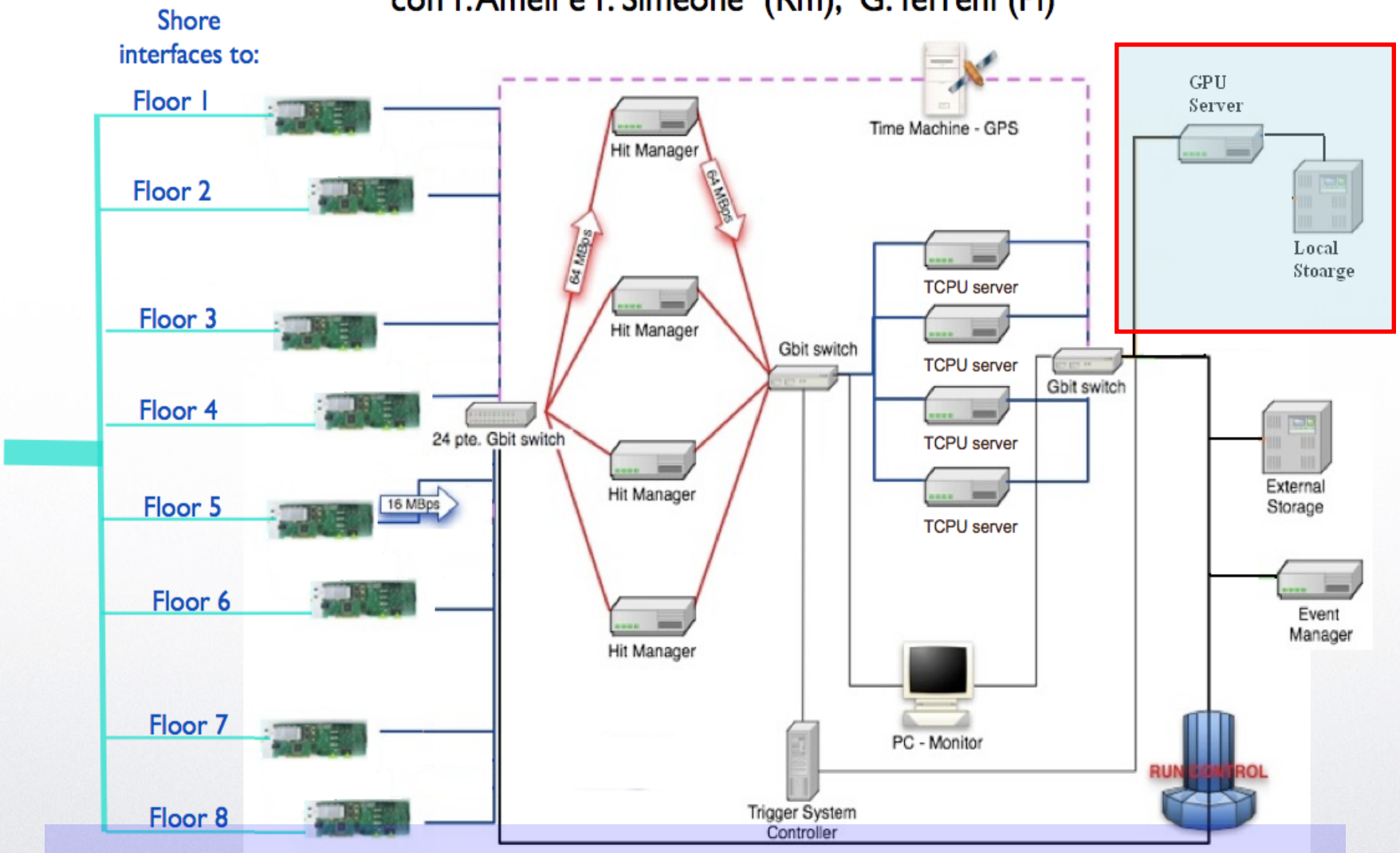
NEMO Tower Phase II.

9 floors
4 PMT/Floor
40m interFloor distance
6m Floor Arme.

The Simuated Tower
16 floors.



con F.Ameli e F. Simeone (Rm), G.Terreni (Pi)



GPU in TRIDAS scheme

Neutrino generation and propagation:

A Nemo Phase-II tower:
16 floors with 4 PMTs

50 kHz of background

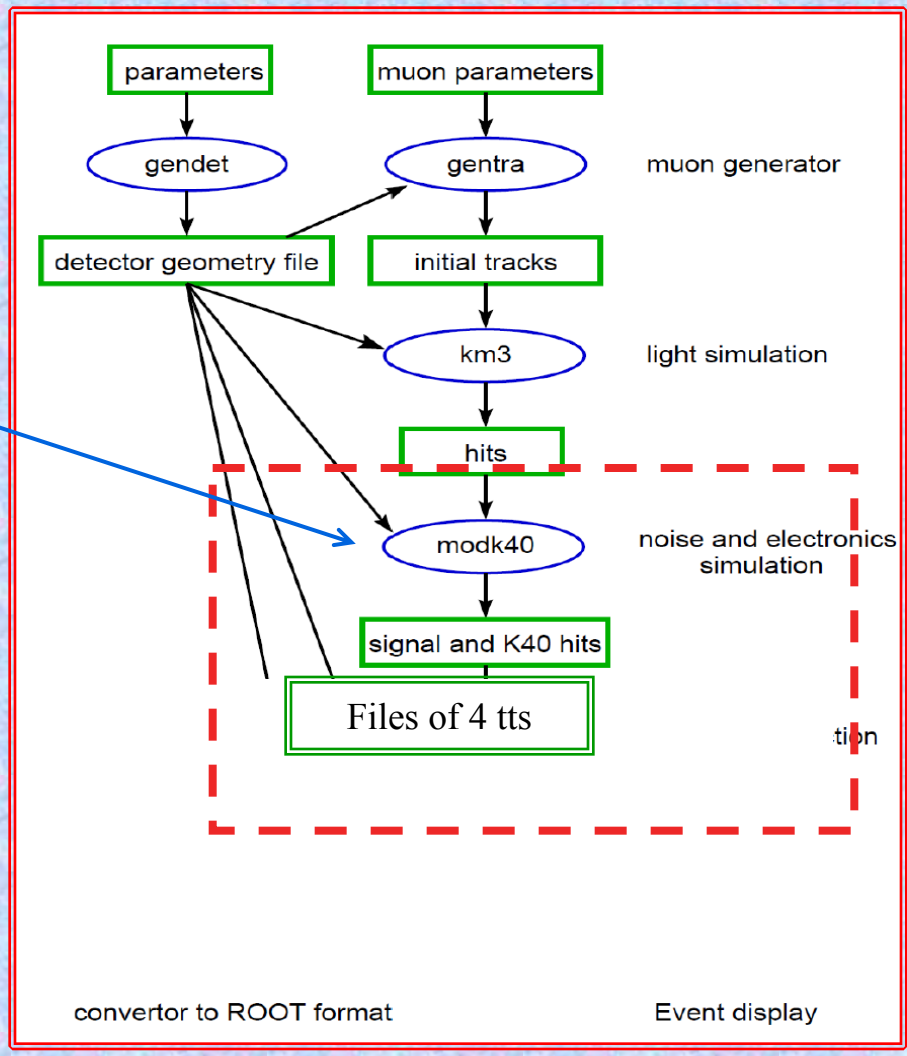
Muon track: >3 hits.

Inter-Events time : 0.1 ms fixed.

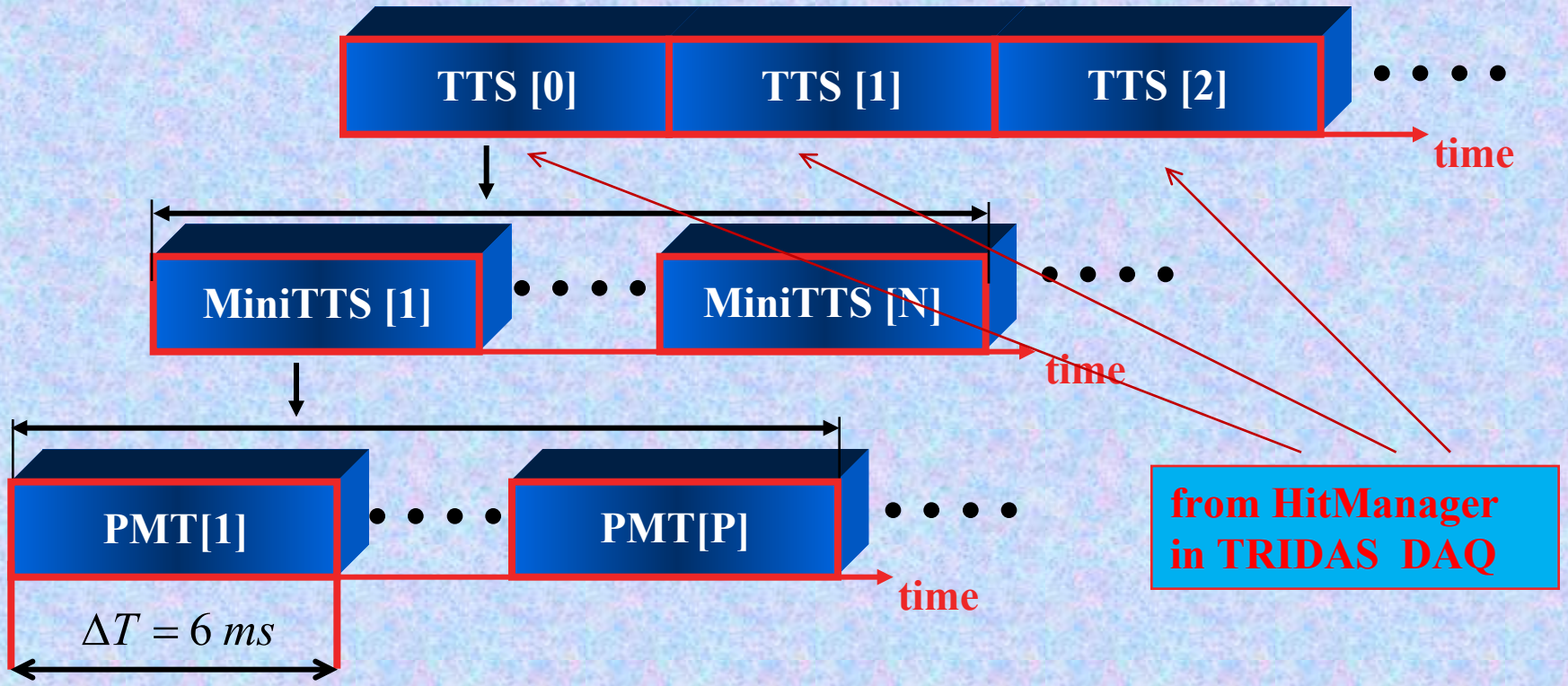
1920 muons track every TTS(192 ms).

4 TTS are grouped in a File =768 ms.

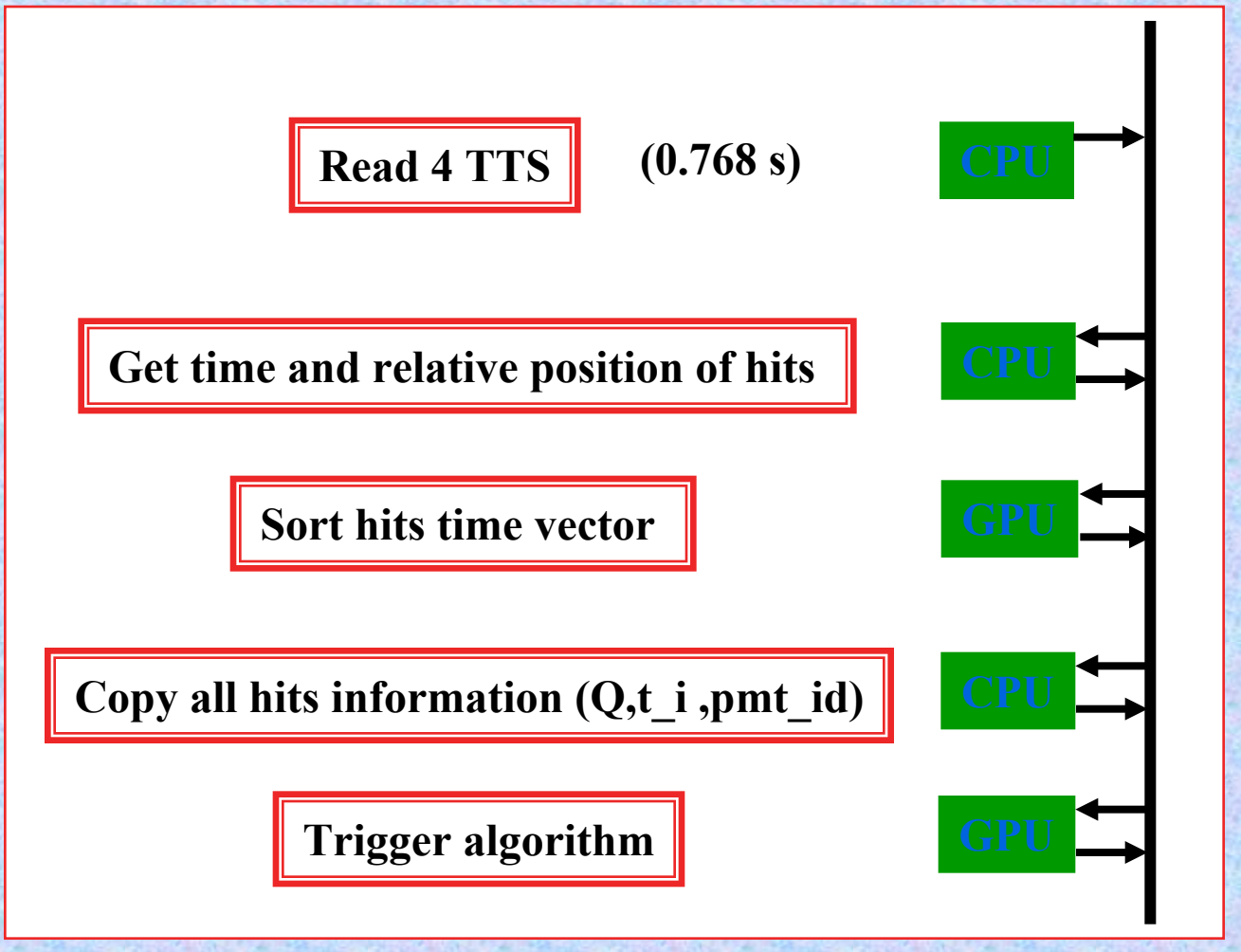
TTS has 32 miniTTS 6ms.



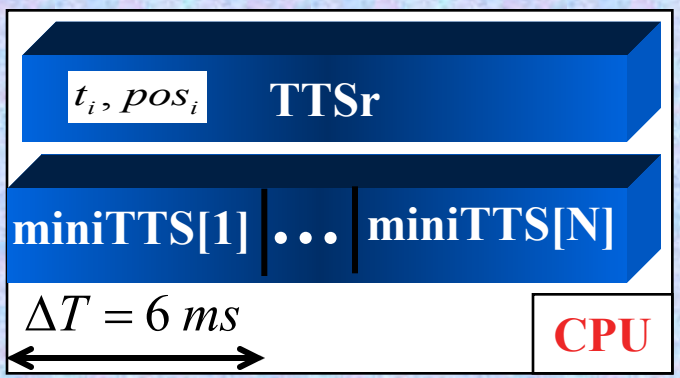
TTS Structure



Global view of the software Implementation



Time Slice Ordering with GPU

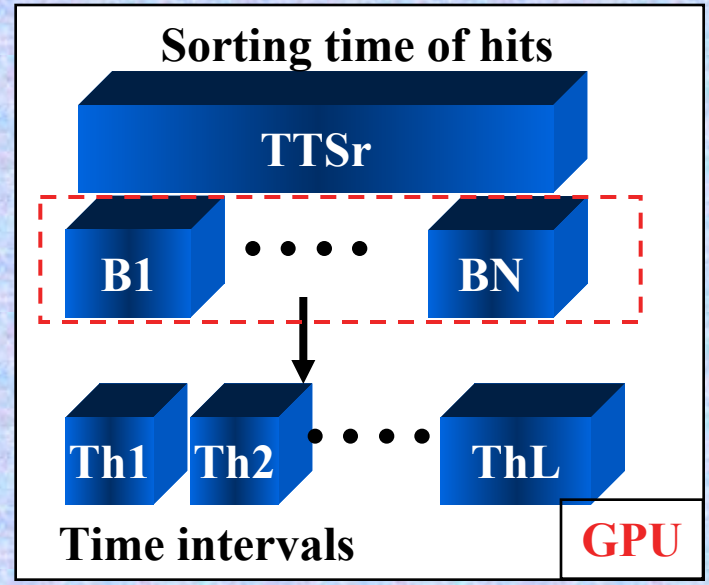


Preparing data in CPU:

- *From each TTS we will read hit time t_i and its relative position (TTSr)*
- *A TTSr is split in N-MiniTTS*
- *The N-GPU block works on the N-MiniTTS*
- *The M-thread of the N-block has its L=MxN working time interval, [*T,T+ΔT*].*

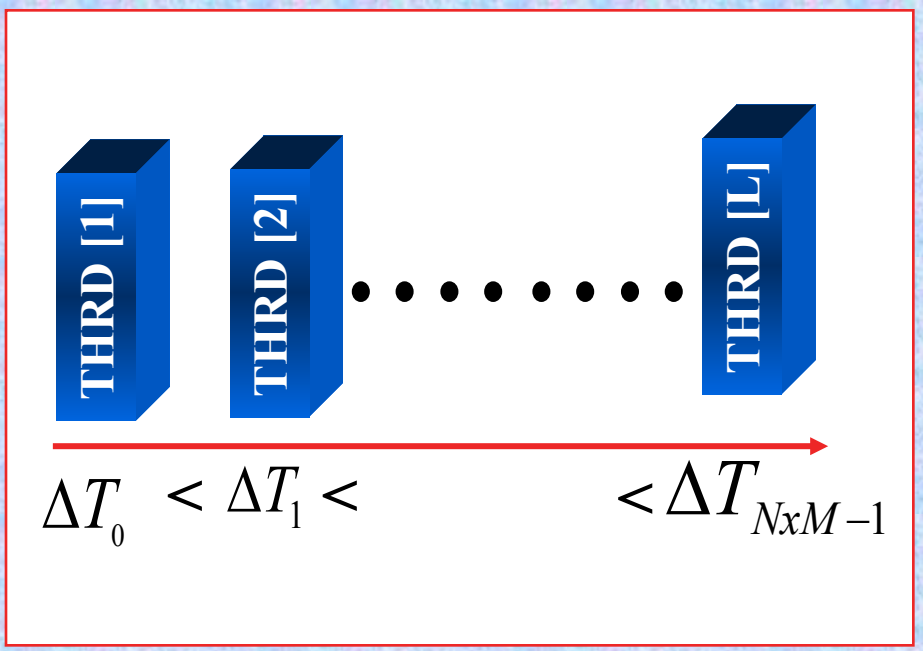
- *All thread in the same block pick up all hits. that belong to their time inteval.*
- *Each thread orders all hits in the ΔT.*

The treads have on average the same number of hits.



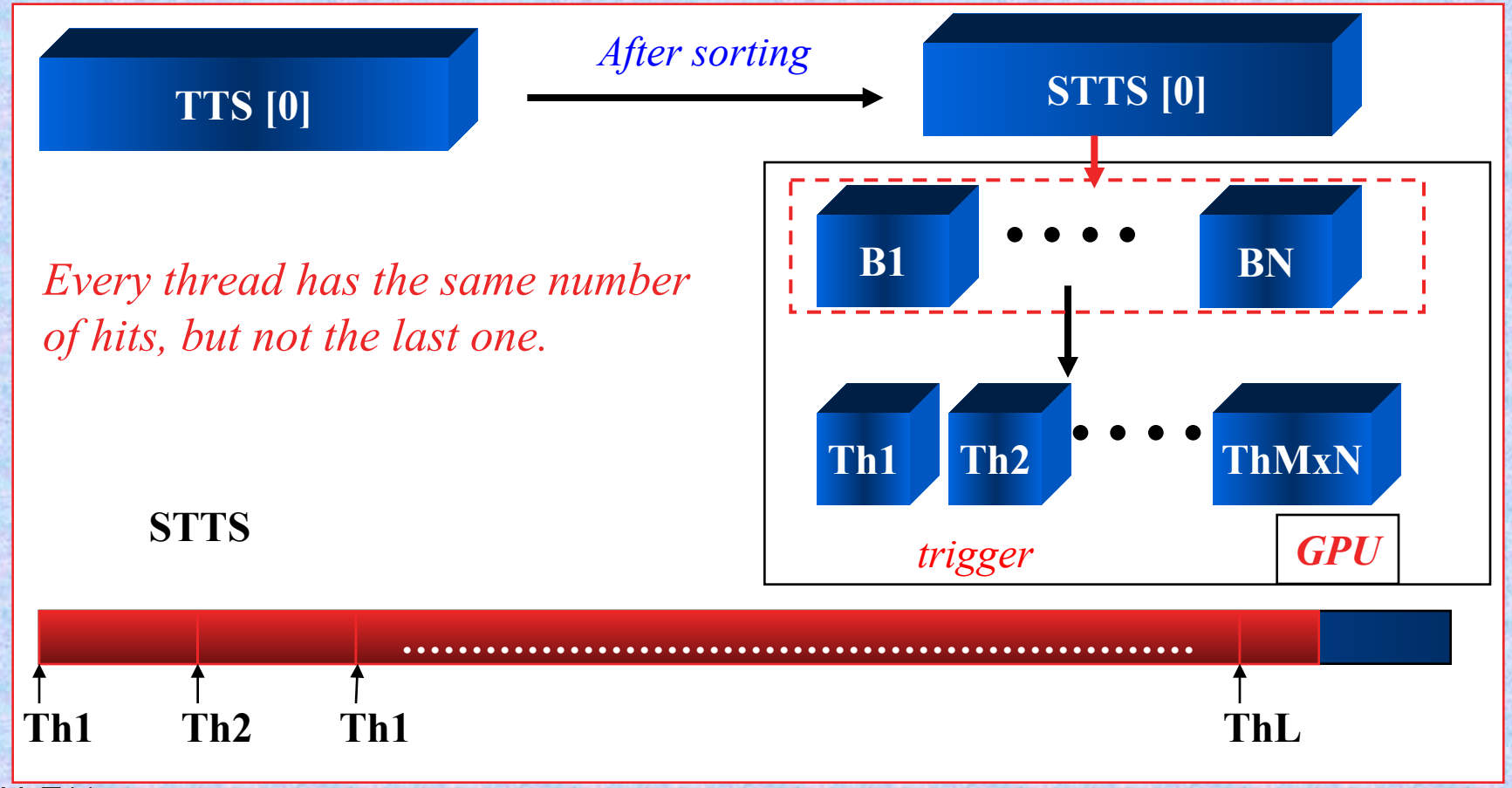
Sorting Alorithms

Few ordering algorithm were tested (shellsort, bubble sort, quickSort). Quicksort is the best.
We used quickSort algorithm, based on divide-and-conquer strategy which has in average O (n log n) operations.



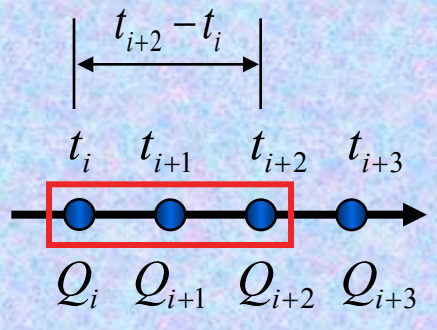
Trigger with GPU

All hits in threads are assembled to form a new TTS called STTS used for the trigger selection.

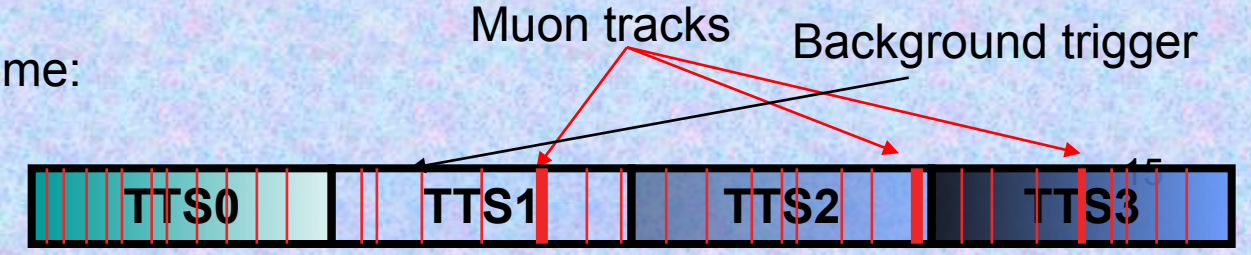


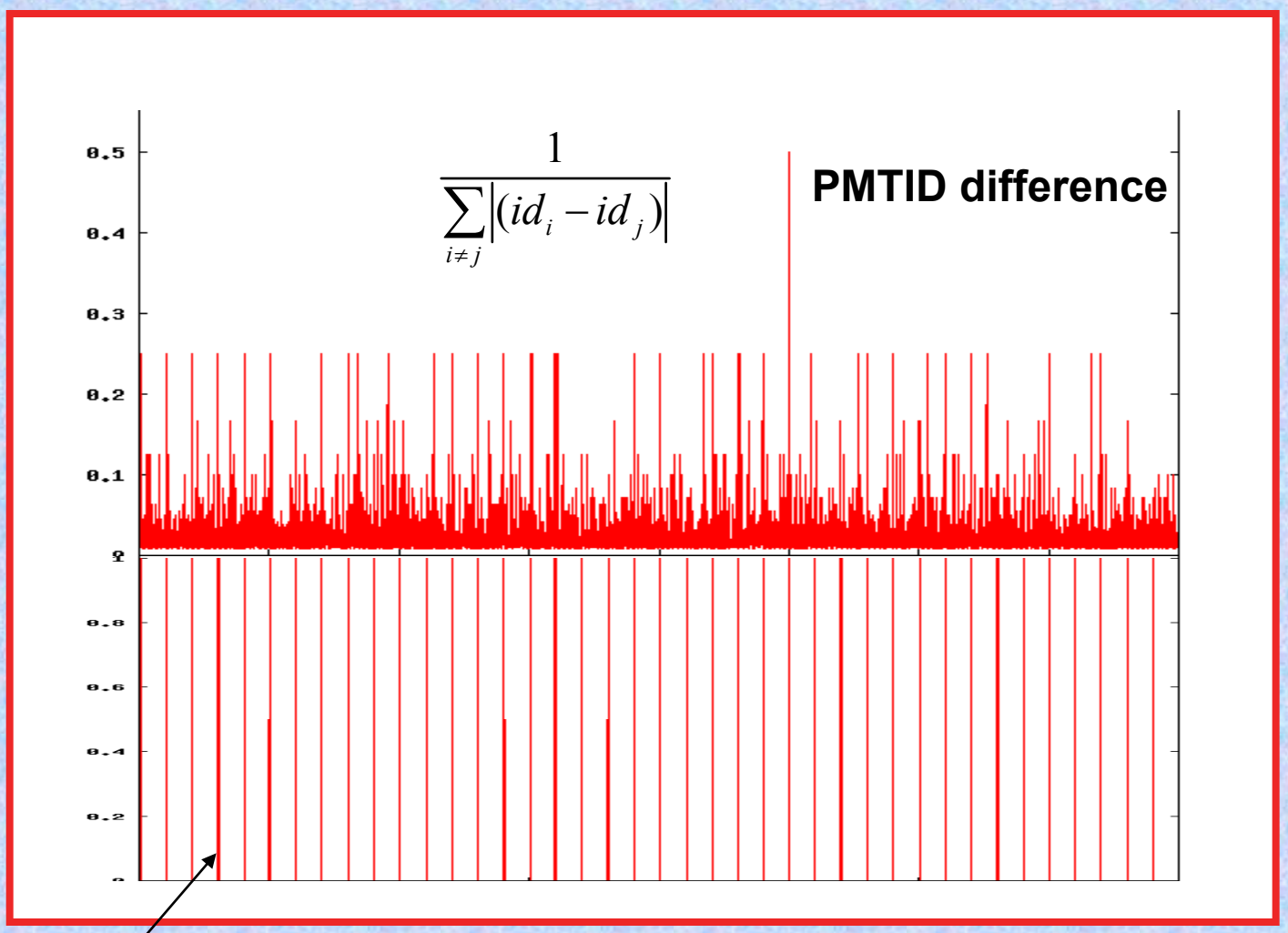
Trigger algorithm (on each thread)

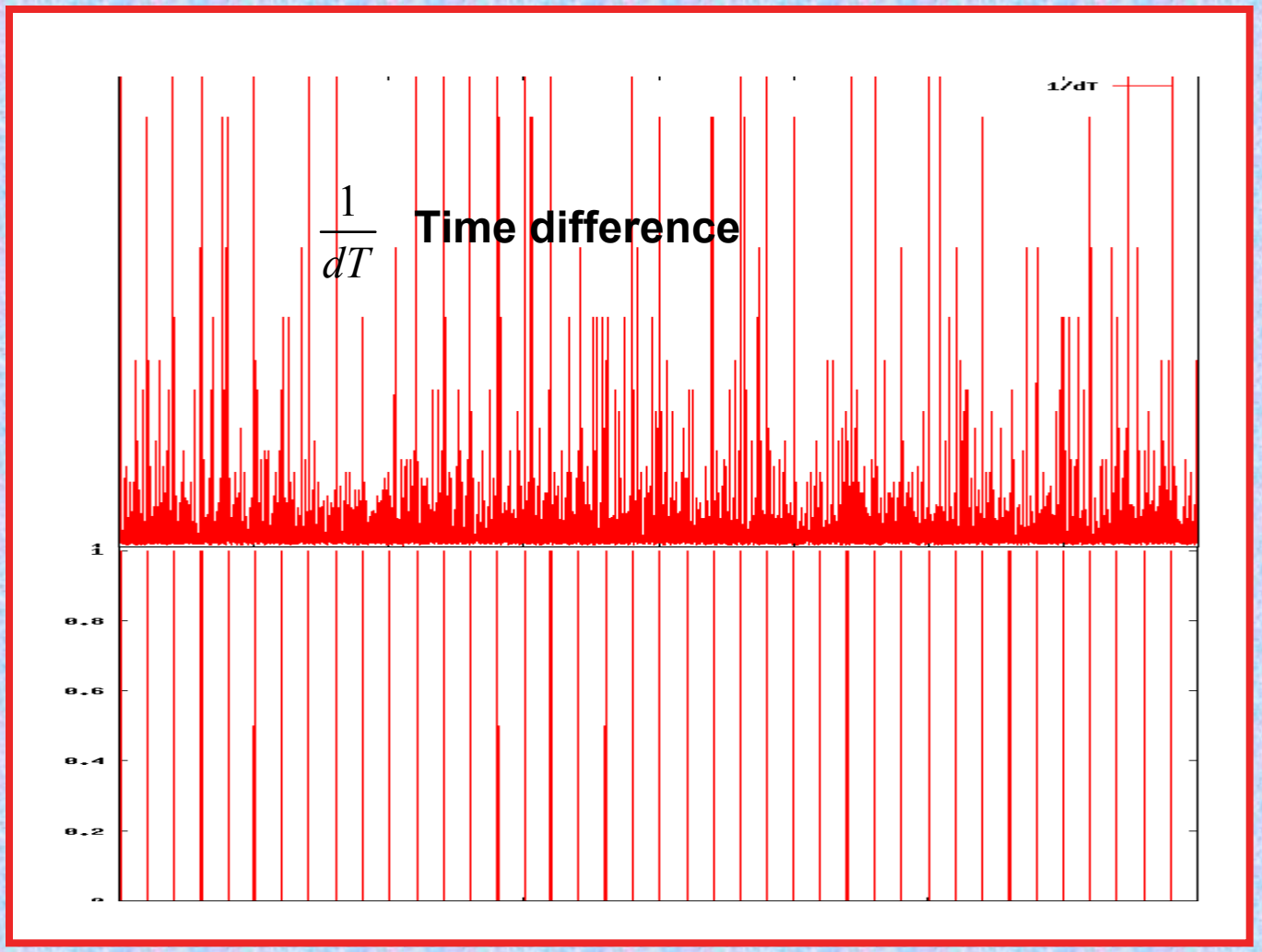
- Each thread will work on portion of data, where three tasks to do:
- For hit $h[i]$,and the two next, we calculate the sum of its charges $Q_m = Q_i + Q_{i+1} + Q_{i+2}$.
- For each hit $h[i]$, we calculate the time difference δT between the current hit $t_{h[i]}$, and $t_{h[i+2]}$.
- For each hit $h[i]$, we calculate sum of pmt_{id} differences $\equiv id_m$ e.g. $| id_{h[i]} - id_{h[i+1]} |$, and $| id_{h[i]} - id_{h[i+2]} |$.
- The quantity $\frac{1}{\delta T \times id_m}$ can be used as trigger.
- To include the charge effect, we will use $\frac{Q_m}{\delta T \times id_m}$ as first level of trigger .

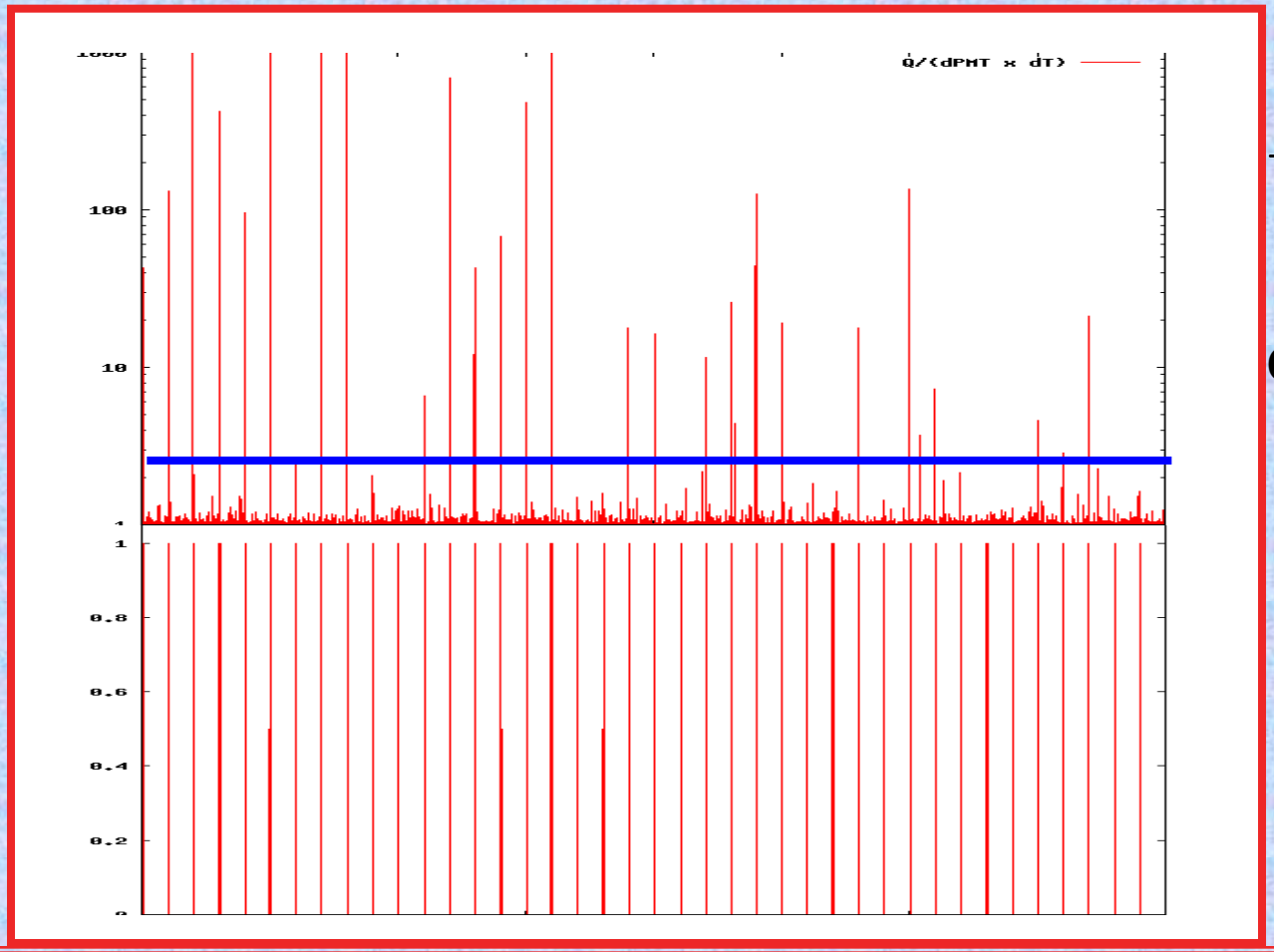


Trigger selection scheme:




$\frac{1}{\sum_{i \neq j} |(id_i - id_j)|}$
PMTID difference
0.5
0.4
0.3
0.2
0.1
0.8
0.6
0.4
0.2
True time of events

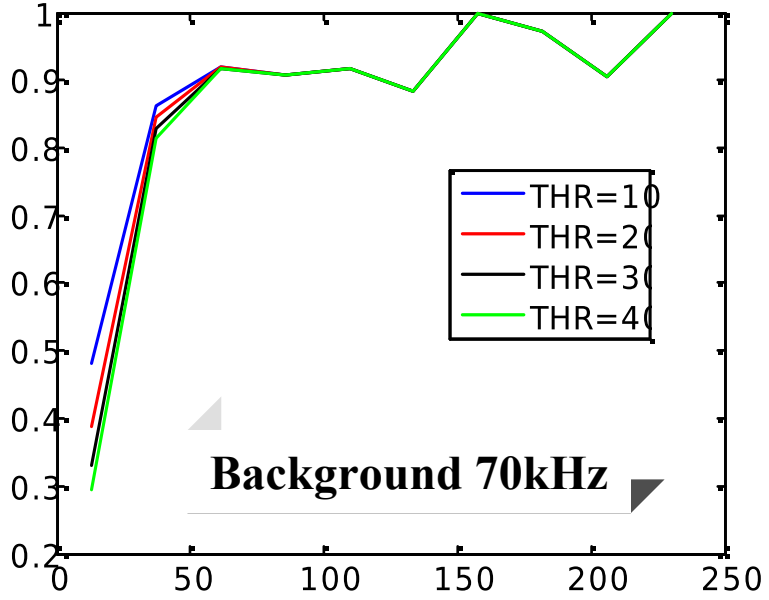

$\frac{1}{dT}$ Time difference
1/dT
1
0.8
0.6
0.4
0.2

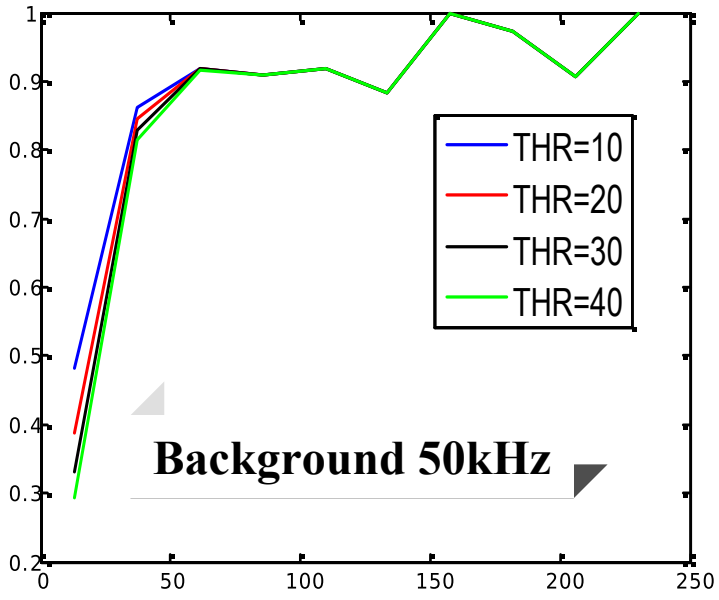


$$\frac{30 \times Q_i}{(dT_i \sum_{i \neq j} \left| (id_i - id_j) \right| + 1)}$$

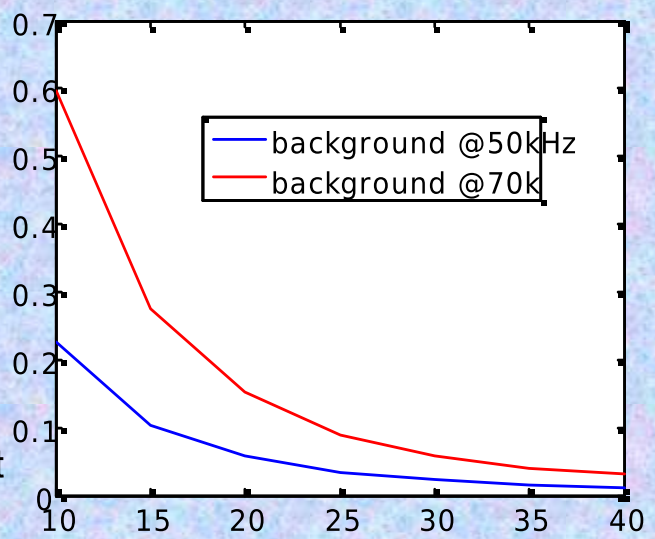
Charge +PMTID + time difference

Trigger Efficiency hits>=4 $\frac{N_{\text{detected}}}{N_{Total}} = 70\%$ **Thr=10**



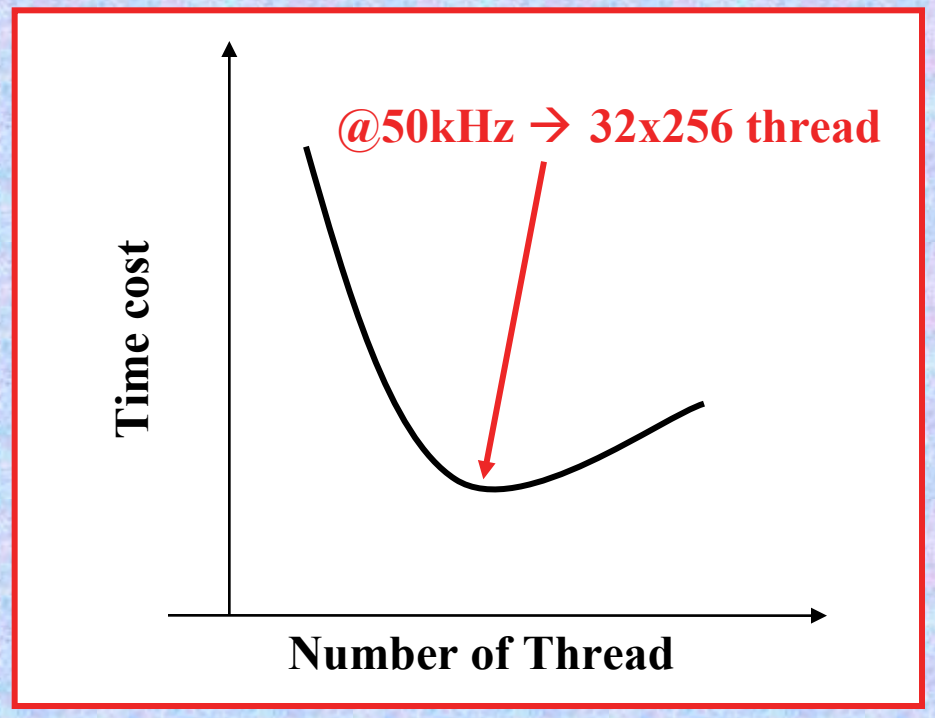


4 x TTS(192ms) was done in:
250ms @50kHz,
300ms @70kHz.



How many threads can be excecued in GPU ?

Maximum threads per GPU 65536x65536 But :



onclusion and Future Work

-Working with GPU in Neutrino Telescope is feasible.
-Time as well hardware space can be saved.
-A test in NEMO Phase II, and a test on km3Net Tower will be done
-Studying more efficient parallel algorithm for triggers and data manipulation to save time as well power.
- On-line visualisation can also be done wit GPUs.

Thank your for attention
End