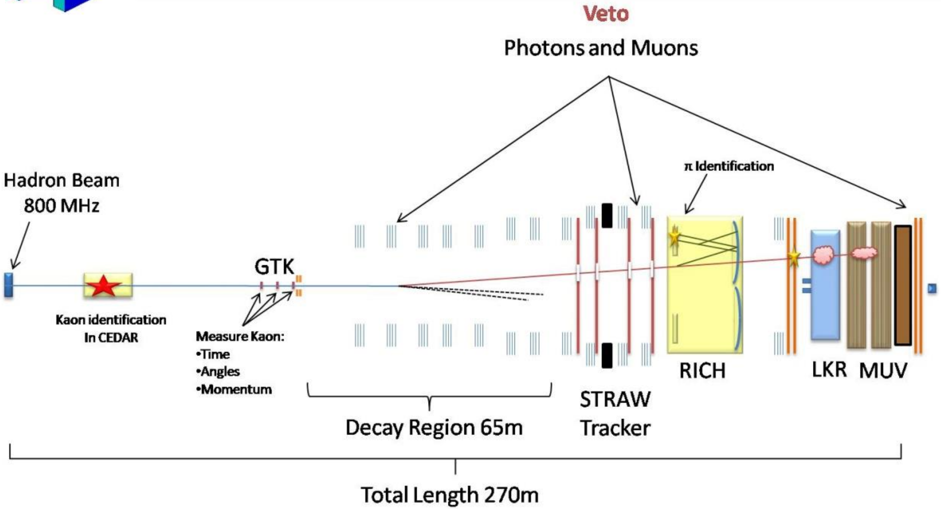
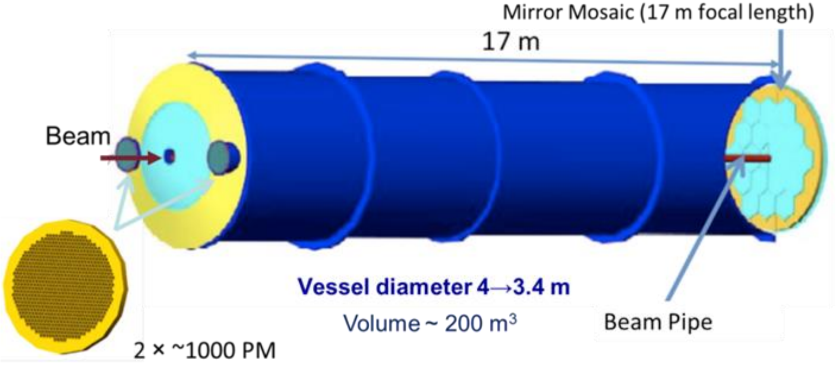
The NaNet Project:
Real-time Distributed Heterogeneous Stream Processing for the NA62 Low Level Trigger

Alessandro Lonardo
(INFN Roma – APE Parallel and Distributed Computing Lab)
on behalf of the NaNet team

23rd International Conference on Computing in High Energy and Nuclear Physics (CHEP 2018)
National Palace of Culture, Sofia, Bulgaria, 9-13 July 2018

The NA62 Experiment at CERN SPS



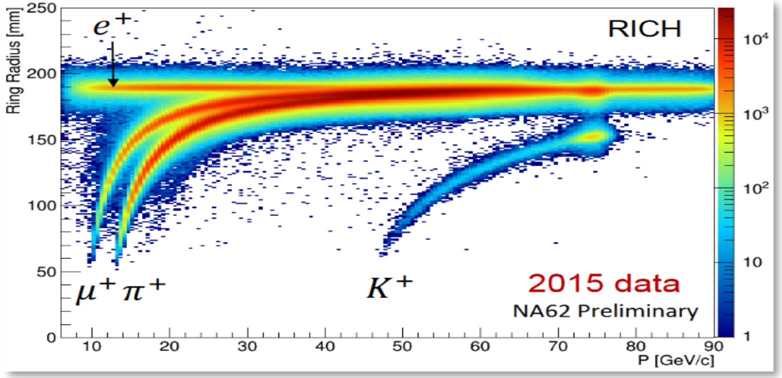


NA62

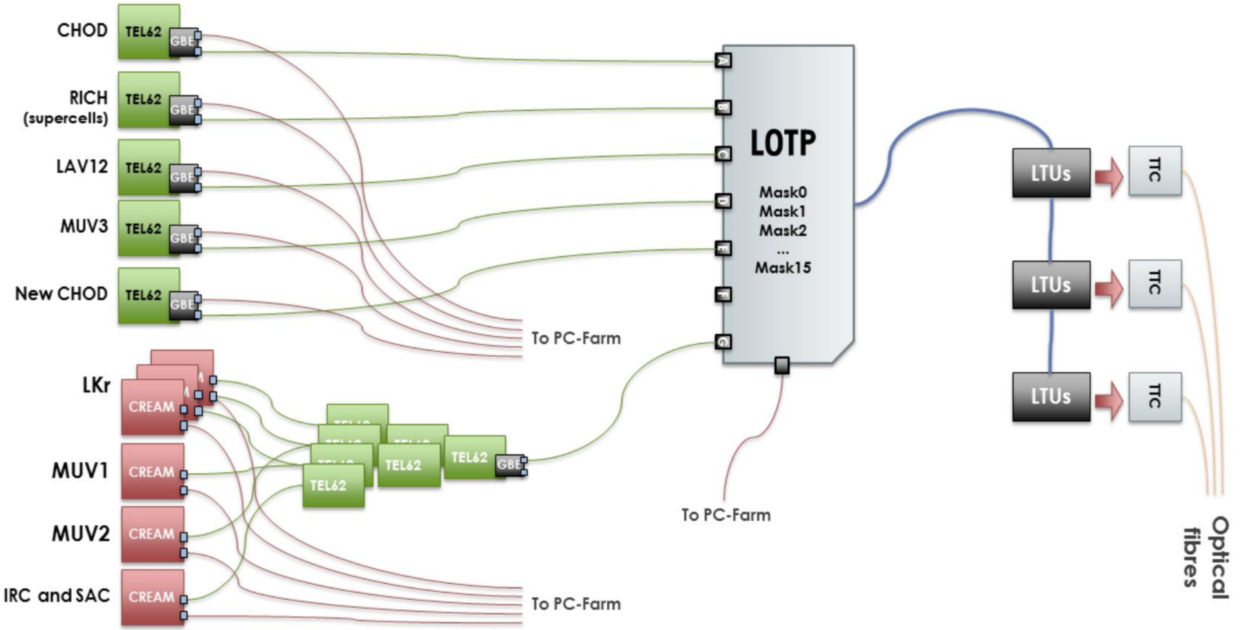
- Measurement of the ultra-rare decay $K^{+} \to \pi^{+}\upsilon\overline{\upsilon}$ (SM BR = 8.4x10^{-11}).
- Fixed Target experiment: 75 GeV secondary beam (6% kaons).

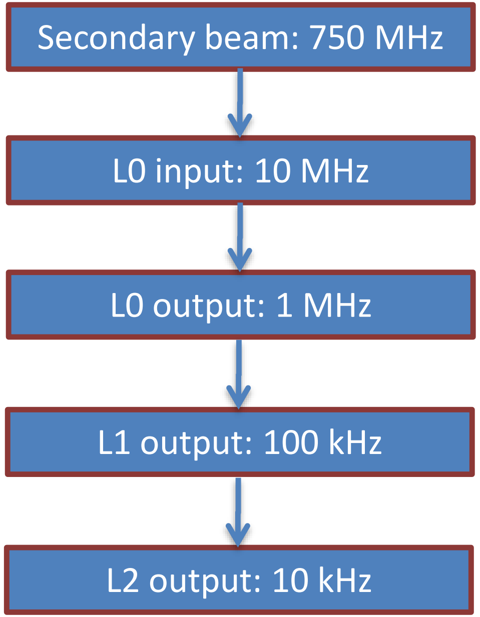
RICH detector

- Cherenkov detector filled with Neon;
- Distinguish between pions and muons from 15 to 35 GeV
- Time resolution of 70 ps (time reference for L0 Trigger)



The NA62 Data Acquisition and Low Level Trigger





□ Fixed target experiment: ~6 seconds long ***bursts***.

□ Some detectors send raw data (***primitives***) to the FPGA-based level-0 trigger processor L0TP.

□ Primitives are generated from TEL62 read out boards.

□ Primitives are sent over 1GbE UDP channels to L0TP.

□ L0 trigger is generated within a maximum latency of **1 ms**.

□ Calorimeters and GTK send data after L1 request.

□ L2 trigger runs over the complete event information.

The NaNet Project

- NaNet is an INFN-funded technology research project aimed to investigate the usage of heterogeneous computing devices in HEP DAQ and low level trigger systems.
- General idea: bring the power and flexibility of modern heterogeneous computing devices, such as GPUs and high-end FPGAs, close to the data source to improve low level trigger performances → more physics.
- Issues: meet the **real-time and throughput requirements** of target systems.
- NA62 was the perfect physics case study:
    - High background rejection requirements.
    - 10 MHz event rate in input to low level trigger.
    - 1 ms time budget for the low level trigger allows to integrate GPUs in the heterogeneous processing pipeline.
    - We started with the RICH detector.

GPU-RICH Motivation

- The knowledge of Cherenkov rings parameters would allow to build more stringent conditions for data selection in the L0 Trigger Processor (L0TP).
- Retrofit the RICH detector with a heterogeneous processing pipeline capable of reconstructing online the rings geometry (center and radius) → GPU-RICH.
- GPU-RICH trigger primitives, from simple hit multiplicity to:
    - total number of rings (0, 1, 2, >2);
    - number of electron rings;
    - number of “spurious” rings (75 GeV K and pi from the beam).
- L0TP can use this refined primitives to tag/veto different decay channels directly **at level-0.**

GPU-RICH Processing

Real-time processing composed by:

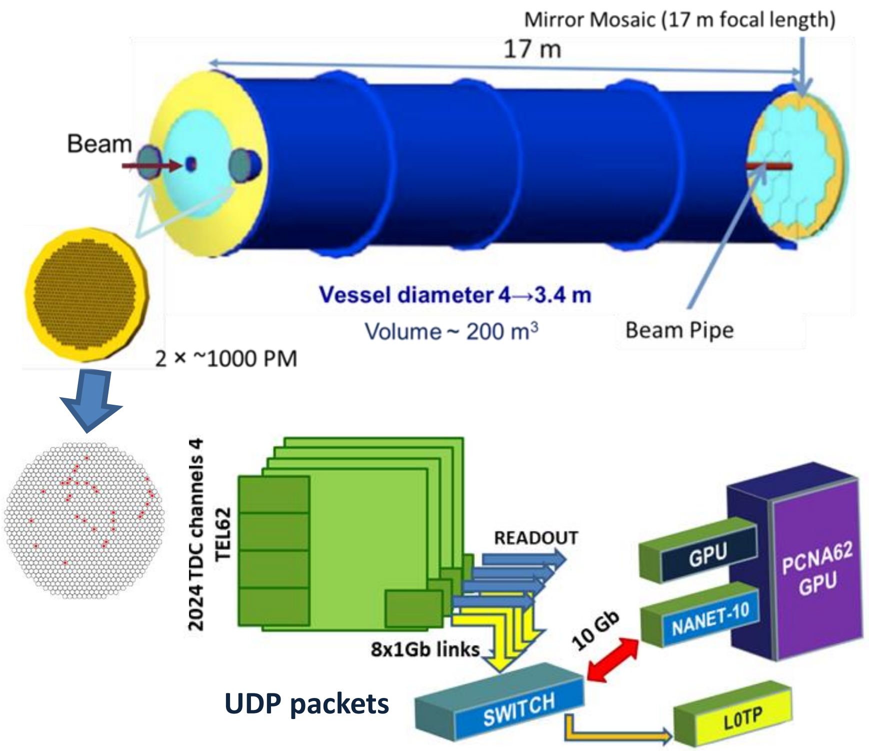
1. **Data transport** tasks (receive data from detector's RO boards directly in the memory of the processing device, send refined primitives to L0TP), *stable latency and synchronous operation*.
2. **Processing** tasks
   a) on data streams (merge events fragments, decompression), *low latency;*
   b) on data buffers (ring reconstruction and primitives generation), *high throughput*.
3. **Control** task (orchestrate other tasks), *low and stable latency*.

GPU-RICH Processing: Allocating Tasks to Devices

Real-time processing composed by:

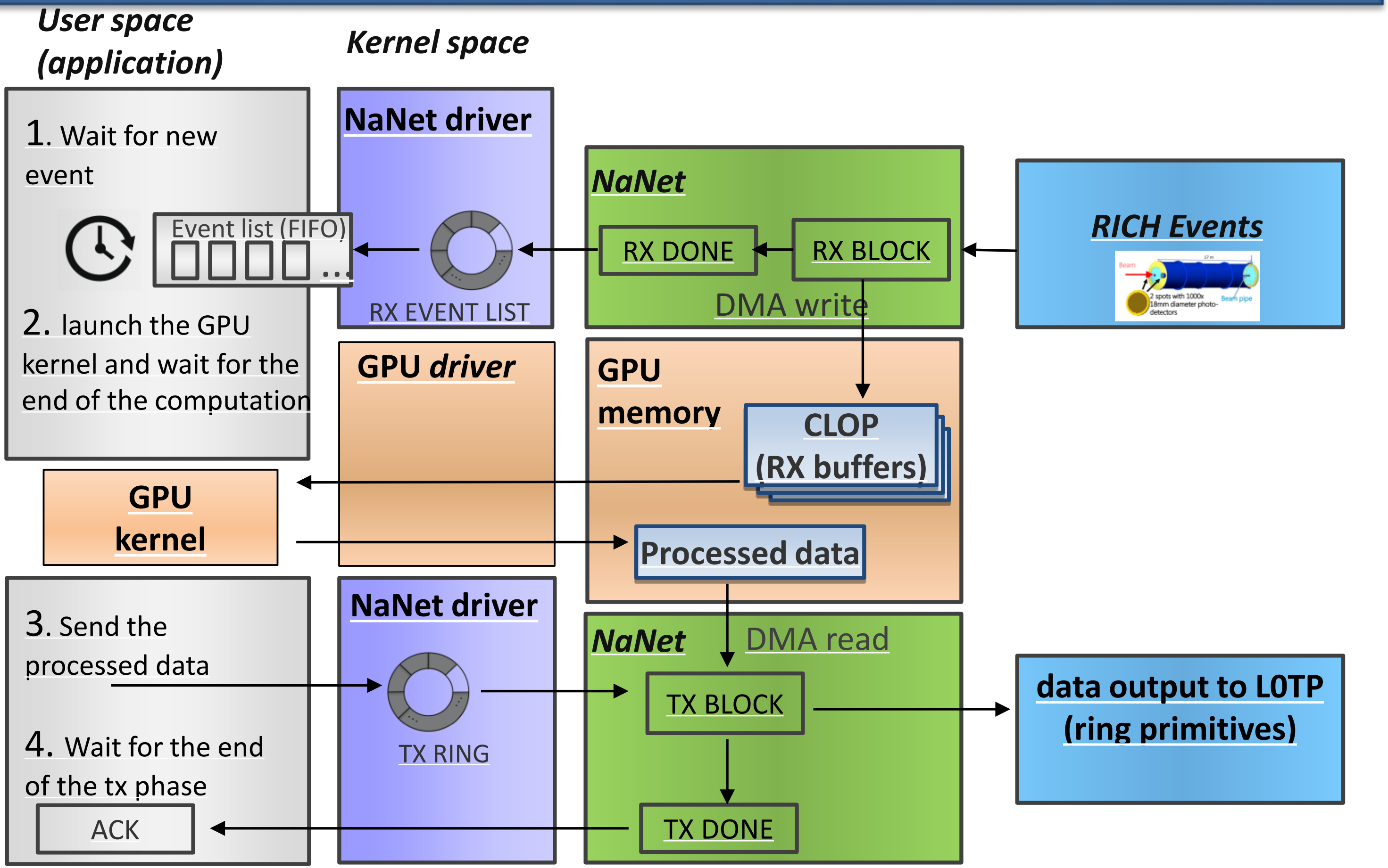
1. **Data transport** tasks (receive data from detector’s RO boards directly in the memory of the processing device, send refined primitives to L0TP), *stable latency and synchronous operation* => **FPGA**
2. **Processing** tasks => **FPGA+GPU**
   a) on data streams (merge events fragments, decompression), *low latency* => **FPGA**
   b) on data buffers (ring reconstruction and primitives generation), *high throughput* => **GPU**
3. **Control** task (orchestrate other tasks), *low and stable latency* => **CPU**

GPU-RICH Heterogeneous Processing Pipeline



- 4 TEL62 RO boards send UDP streams with RICH events to the switch using 8x1GbE links.
- NaNet-10 FPGA-based NIC receives UDP RICH events streams over a 10GbE link (from the switch).
- NaNet-10 processes event streams:
  - decompression;
  - merging of events split on different streams;
  - change data alignment for GPU memory access on the 4kB staging buffer in FPGA.
- NaNet-10 DMA writes over PCIe merged/re-aligned event data to a receive buffer in GPU memory.
- When the receive buffer is full (or the gathering timeout has expired), the CPU gets notified and launches a sequence of CUDA kernels to perform ring reconstruction and ring primitives generation using event data already in GPU memory.
- The CPU issues a send operation for NaNet-10.
- NaNet-10 DMA reads over PCIe the ring primitives from GPU memory and sends them to the L0TP (synchronous operation).

NaNet: Hardware/Software Interplay



User space (application)

Kernel space

1. Wait for new event

Event list (FIFO)

2. launch the GPU kernel and wait for the end of the computation

NaNet driver

RX EVENT LIST

NaNet

RX DONE

RX BLOCK

DMA write

RICH Events

GPU *driver*

GPU memory

CLOP (RX buffers)

GPU kernel

Processed data

3. Send the processed data

4. Wait for the end of the tx phase

ACK

NaNet driver

TX RING

NaNet

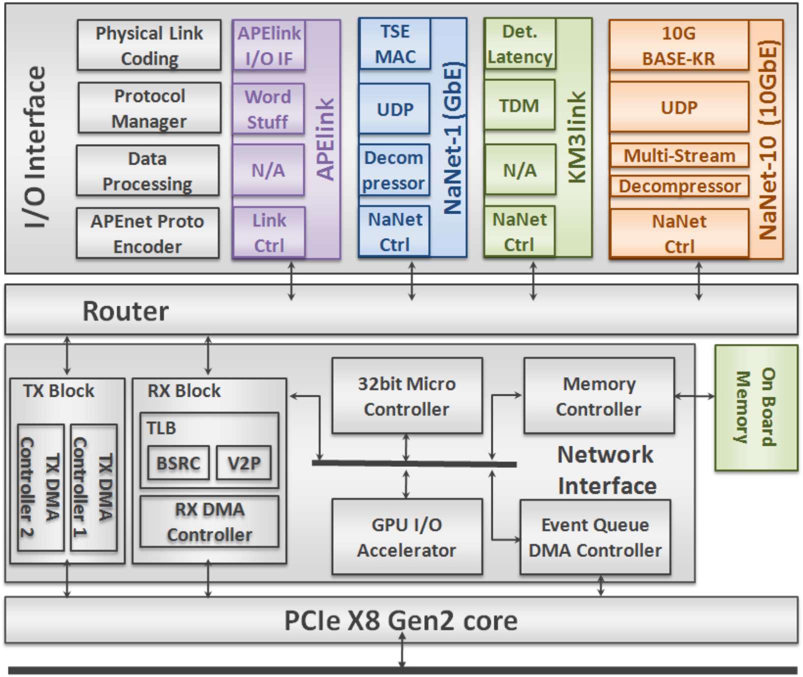
DMA read

TX BLOCK

TX DONE

data output to L0TP (ring primitives)

NaNet: a Design of FPGA-based PCIe Network Interface Cards



I/O Interface

- Support for several physical link technologies and data layers (1GbE Base-T, 10GbE SFP+, 40GbE QSFP+,…).
- Network/transport **protocols offloading** (IP/UDP, …).
- Application-specific processing on data stream.

Router: dynamically interconnects I/O and NI ports.

Network Interface

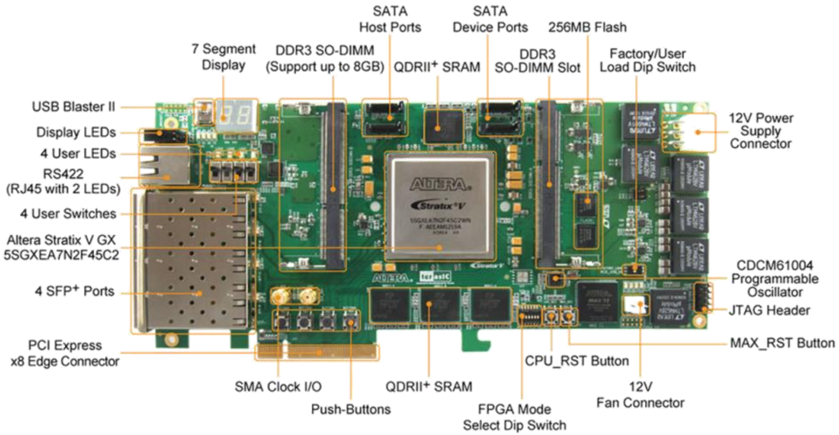
- Zero Copy RDMA in TX and RX.
- **GPUDirect RDMA** (no bounce buffers in CPU).
- TLB for Virtual to Physical memory mapping.
- 32-bit Microcontroller for control and configuration tasks.

PCIe X8 Gen2/3 Core with multiple DMA engines

Timing, Trigger and Control Interface for synchronous operation

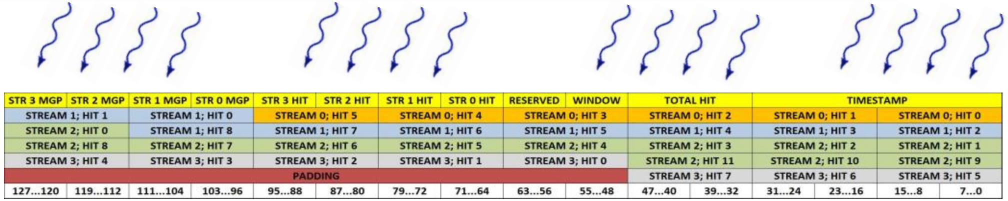
NaNet for GPU-RICH: NaNet-10

- Terasic DE5-NET (Altera Stratix V FPGA)
- PCIe x8 Gen2/3
- 4 SFP+ ports (10GbE)
  - MAC 10GBASE-KR
- nVIDIA GPUDirect RDMA
- UDP offloading
- TTC interface



Rings Reconstruction on GPU

Event finder: the GPU mem buffer is scanned for a specific delimiting pattern to index events

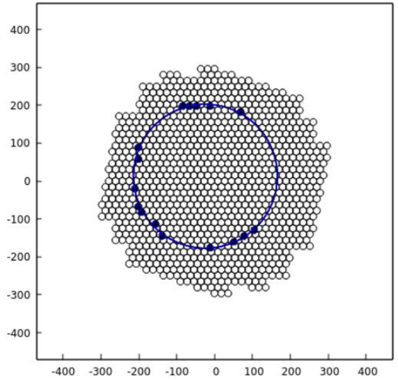


STR 3 MGP	STR 2 MGP	STR 1 MGP	STR 0 MGP	STR 3 HIT	STR 2 HIT	STR 1 HIT	STR 0 HIT	RESERVED	WINDOW	TOTAL HIT		TIMESTAMP			
STREAM 1; HIT 1		STREAM 1; HIT 0		STREAM 0; HIT 5		STREAM 0; HIT 4		STREAM 0; HIT 3		STREAM 0; HIT 2		STREAM 0; HIT 1		STREAM 0; HIT 0	
STREAM 2; HIT 0		STREAM 1; HIT 8		STREAM 1; HIT 7		STREAM 1; HIT 6		STREAM 1; HIT 5		STREAM 1; HIT 4		STREAM 1; HIT 3		STREAM 1; HIT 2	
STREAM 2; HIT 8		STREAM 2; HIT 7		STREAM 2; HIT 6		STREAM 2; HIT 5		STREAM 2; HIT 4		STREAM 2; HIT 3		STREAM 2; HIT 2		STREAM 2; HIT 1	
STREAM 3; HIT 4		STREAM 3; HIT 3		STREAM 3; HIT 2		STREAM 3; HIT 1		STREAM 3; HIT 0		STREAM 2; HIT 11		STREAM 2; HIT 10		STREAM 2; HIT 9	
PADDING										STREAM 3; HIT 7		STREAM 3; HIT 6		STREAM 3; HIT 5	
127…120	119…112	111…104	103…96	95…88	87…80	79…72	71…64	63…56	55…48	47…40	39…32	31…24	23…16	15…8	7…0

buffer slice <---> thread

Ring fitter: fast, no seed, purely geometrical

event <---> block **hit <---> thread**



For every block/event

- Crawford algorithm[*] to estimate radius/center
- χ^2 test btw estimation and hits distances/center of gravity
  - if χ^2<threshold --> 1 ring only
  - If above threshold we'll search for more rings

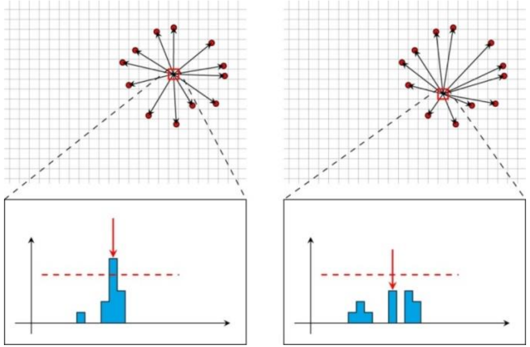
Multi-rings Pattern recognition: histogram-based method

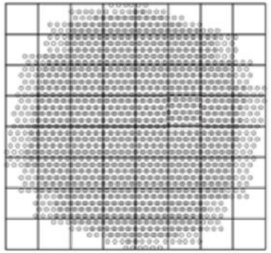
[*] J. Crawford, "A non-iterative method for fitting circular arcs to measured points", Nuclear Instruments and Methods in Physics Research, vol. 211, no. 1, 1983.

Histogram Parallel Ring Reconstruction Algorithm

event <---> block   **grid point <---> thread**

- Each thread computes the distances between its grid point and the hits;
- a histogram of such distances is created;
- grid points having a histogram that has a bin over threshold is a ring center candidate.





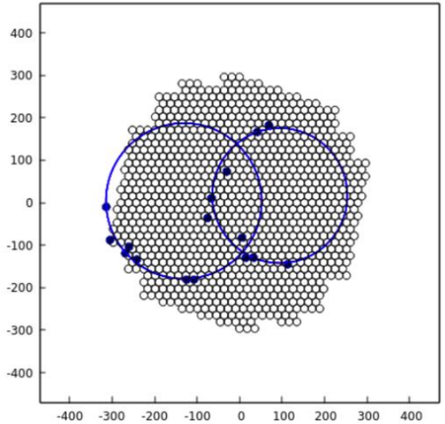
8x8 grid -> 64 threads

hit <---> thread

- Hits in annular region around each candidate center are selected.

ring <---> thread

- Crawford algorithm is applied for every set of selected hits to estimate ring parameters



Results from the 2018 NA62 RUN

Setup

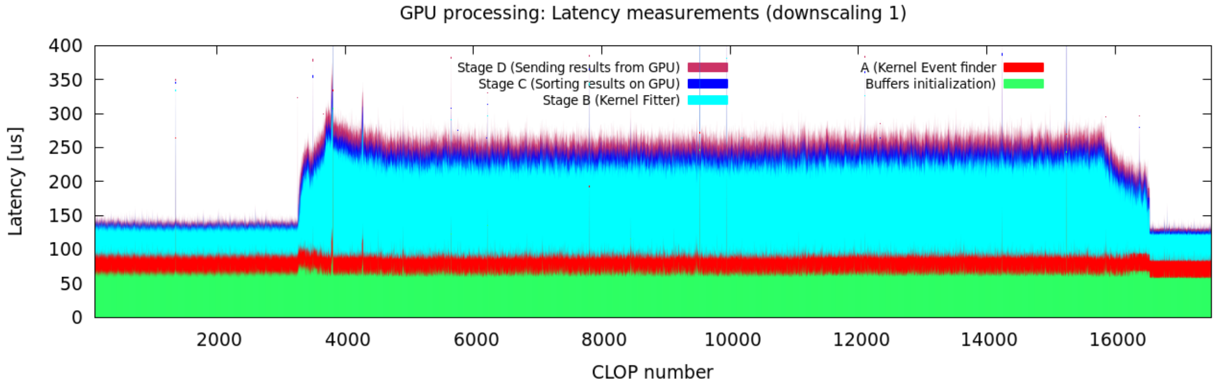
- Supermicro X9DRG-QF Intel C602 Patsburg
- Intel Xeon E5-2602 2.0 GHz (3.10.0-514.16.1.el7.x86_64)
- **NaNet-10 (Terasic DE5-Net)**
- **nVIDIA P100** (CUDA 8.0)

Configuration

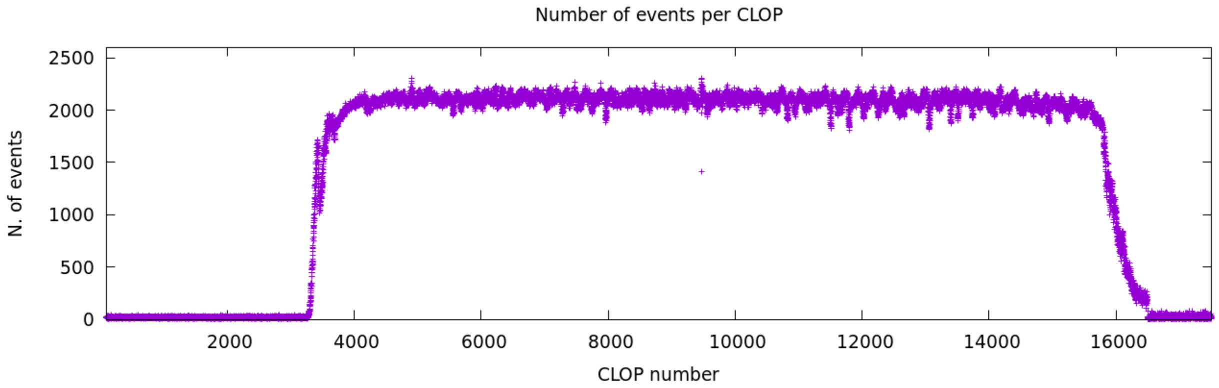
- Gathering timeout: 350us
- ~ 60% target beam intensity (~ 21*10^{11} Pps)
- Histogram algorithm
- System running in parasitic mode

Results

- **Processing latency: 260 us (99th percentile)**
- **Processing time per event: ~ 0.13 us**
- **Compatible with the 1 ms time budget for the L0 trigger**



GPU processing: Latency measurements (downscaling 1)



Number of events per CLOP

Summary

- Results collected during NA62 Runs let us demonstrate the effectiveness of our design, showing how the GPU-RICH system is able to sustain the events rate of ~6 MHz, computing the Cherenkov ring parameters and sending the corresponding primitives to the low level trigger within a latency of ~300 us.
- NaNet FPGA-based NIC is the key component enabling the integration of this heterogeneous processing pipeline, allowing the full orchestration of tasks allocated to different devices from a user-space application running on CPU.
- Working to assess the quality of online reconstruction (with respect to the offline reconstruction).
- Our target is the integration in the operational L0 trigger chain before LS2.