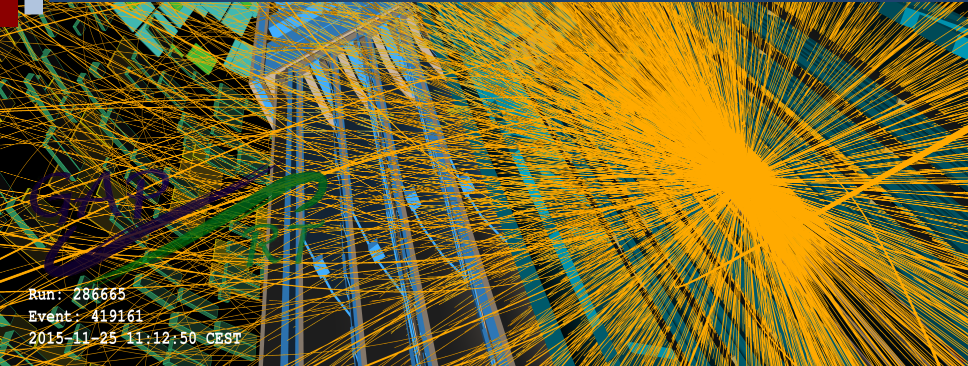
SAPIENZA
UNIVERSITÀ DI ROMA

Matteo Bauce

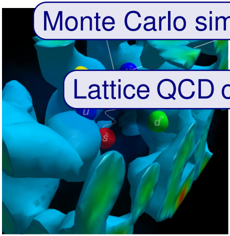
GPU integration in High Energy Physics experiment online event selection systems

Perspective of GPU computing in Science, 26-28/09/16, Roma



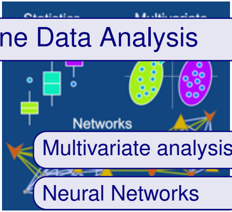
GPU in High Energy Physics

Many applications of GPU in High Energy Physics



Monte Carlo simulation

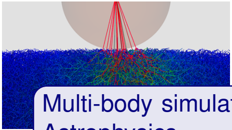
Lattice QCD calculations



Offline Data Analysis

Multivariate analysis

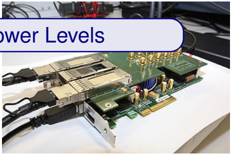
Neural Networks



Multi-body simulation in Astrophysics

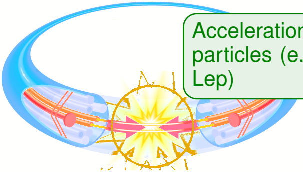
Higher Levels

Realtime event selection

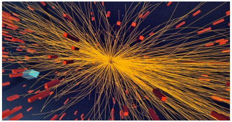


Lower Levels

HEP experiment at colliders

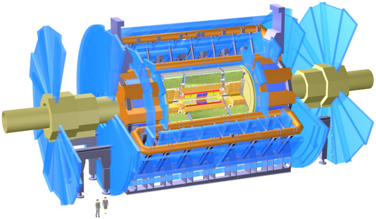


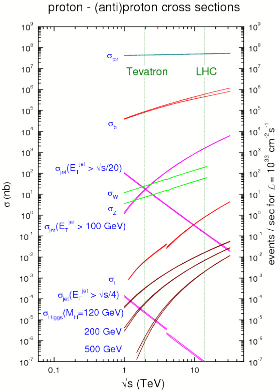
Acceleration and collisions of particles (e.g. LHC, Tevatron, Lep)



Center-of-mass energy up to 13 TeV *pp* collision, rates up to 40 MHz

Multipurpose detector to reconstruct most of the collision information

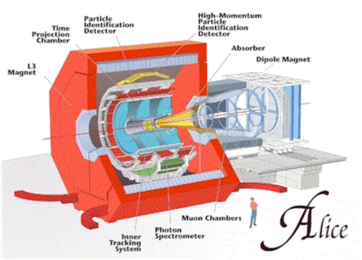




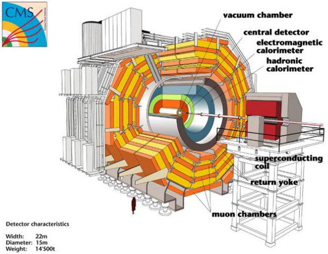
Interesting events are only $1/10^{7-12}$: need to reject most of the others.

Realtime selection plays a fundamental role

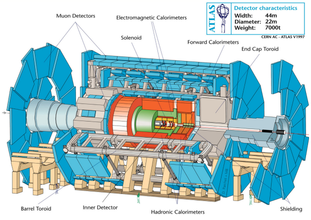
Data-Aquisition systems in LHC experiment



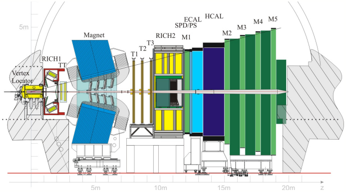
Event size: 100 MB, 300 Hz: 30 GB/s



Event size: 1-2 MB, 40 MHz: 40 TB/s



Event size: 1.5 MB, 40 MHz: 30 TB/s



Event size: 100 kB, 40 MHz: 4 TB/s

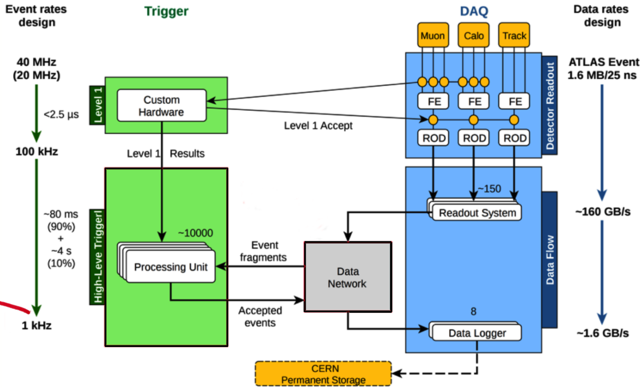
Future upgrades

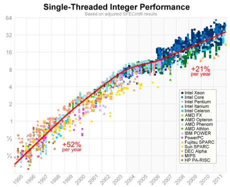
	Run 1	Run 2	Run 3	
Energy ($\sqrt{s}$)	7/8 TeV	13 TeV	14 TeV	
Peak Luminosity ($cm^{-2}s^{-1}$)	10^{34}	$1.5{\cdot}10^{34}$	$2\text{-}3{\cdot}10^{34}$	
Interactions/bunch crossing	30	23	55-80	◄ pileup
Bunch crossing rate	20 MHz	40 MHz	40 MHz	
Offline Storage rate	600 Hz	1000 Hz	1000 Hz	
Bunch spacing	50 ns	25 ns	25 ns	

Data-taking conditions will be more and more demanding in the upcoming years

- Higher collision rates
- Higer number of multiple overlapping events (pileup)
- Detector upgrades might increase event size

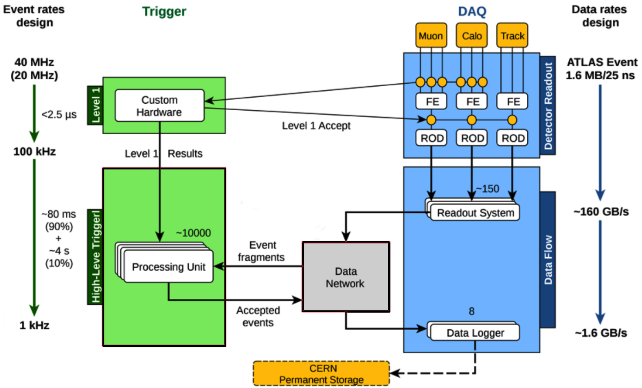
► Processing latencies should remain almost the same $\mathcal{O}(100\ ms)$



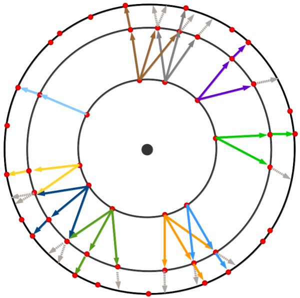


- Multi-stage system based on hardware (LLT) and software (HLT)
- CPU computing power reaching saturation: change of paradigma toward parallel computing

Realtime event selection systems: ATLAS example

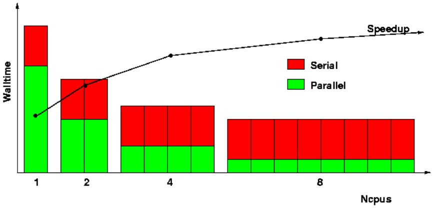


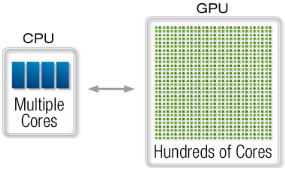
- Multi-stage system based on hardware (LLT) and software (HLT)
- CPU computing power reaching saturation: change of paradigma toward parallel computing
- Try to include GPUs in trigger systems

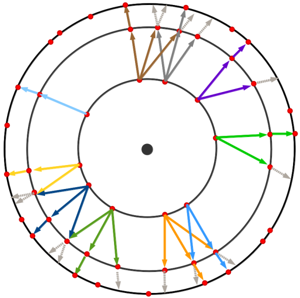


- *Pattern-recognition* algorithms suitable for parallelization (SIMD)

e.g. Different color → Different core

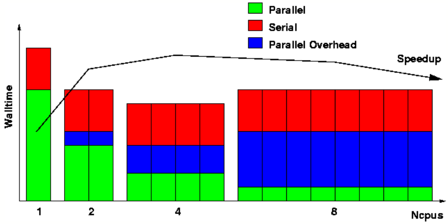


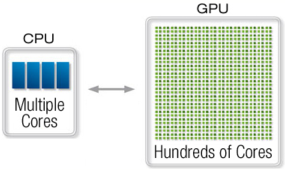


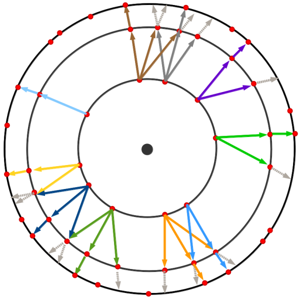


e.g. Different color → Different core

- *Pattern-recognition* algorithms suitable for parallelization (SIMD)
- memory usage is a limitation: small amount available, overhead for data cross-reading algorithm.

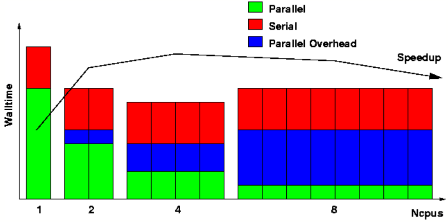


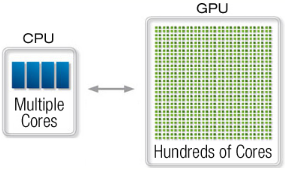




e.g. Different color → Different core

- *Pattern-recognition* algorithms suitable for parallelization (SIMD)
- memory usage is a limitation: small amount available, overhead for data cross-reading algorithm.
- Multi-event parallelization is a BONUS!



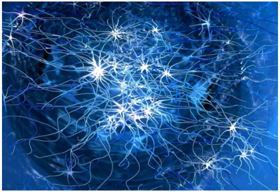


- Main questions that need an answer:

1. How to integrate a GPU in a pre-existing data-taking software?
   - Need to redesign software from scratch?

2. How fast can a GPU be within time constraints from the DAQ system?
   - i.e. how low can you go in the trigger levels?

3. What algorithms get the best from parallelization on GPU?
   - Existing ones are suitable for parallelization?
   - How innovative ones compare in terms of efficiency?

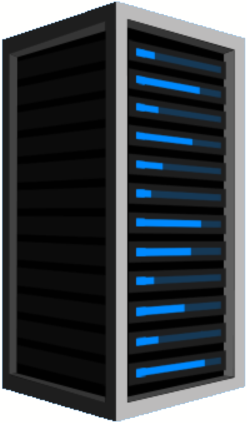


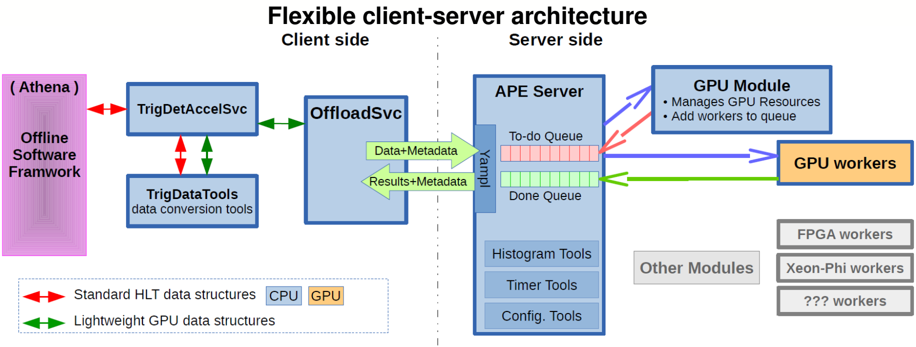
Prototype for a GPU-based ATLAS trigger

- Aim at the evaluation of benefit and disadvantages
  - Need to suppress increase in CPU time due to pileup
  - Limit on HLT farm size from cooling and power
  - Evaluate processing time/event per unit cost
- Investigation on trigger algorithm for Inner Detector (tracking), Calorimeter clustering, Muon segment reconstruction

Server with NVidia Tesla K80

- 2 chips in each card
- 2 GB RAM
- 13 multi-processor
- 192 cores per multiprocessor
- 2496 CUDA cores
- 824 MHz GPU, 2505 MHz memory clock





Client:

- One HLT processing unit per core
- Offline & Online framework (Athena)
  - manage data
  - execute chains of algorithms
  - monitors data-processing

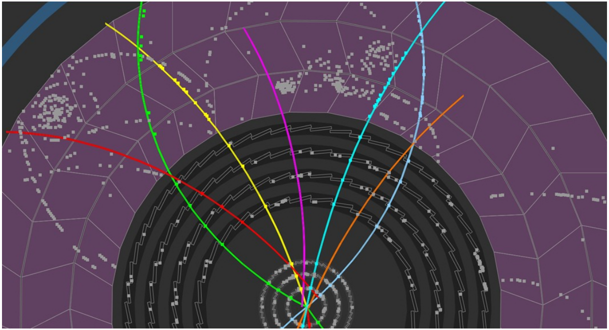
Server:

- Independent from Client framework
- Flexible hardware resources management (multi-devices)
- Preallocate memory for data and store constants

Inner Detector

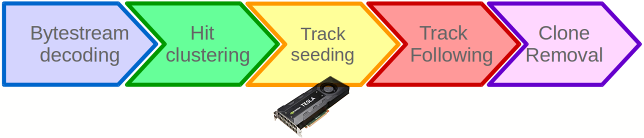
Tracking is the most time consuming algorithm

- Sequential steps: silicon hit clustering, seeds creation, track following

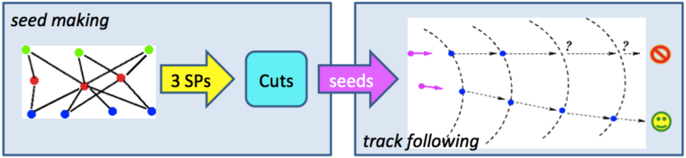


Inner Detector

Tracking is the most time consuming algorithm



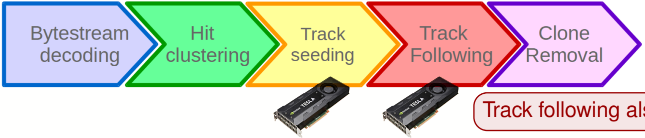
- Sequential steps: silicon hit clustering, seeds creation, track following
- Parallelization on GPU of **track-seeding**
- Huge data multiplicity for a full-detector scan tracking: a GPU makes it feasible



Pair formation: 2D thread array checking for pairing conditions

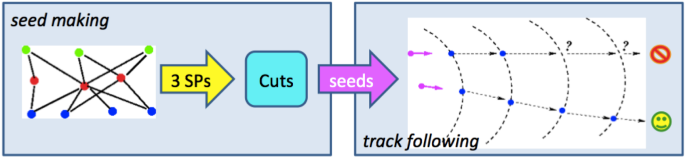
Triplet formation through 2D thread block

Tracking is the most time consuming algorithm



Track following also may benefit

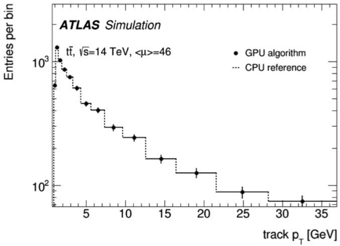
- Sequential steps: silicon hit clustering, seeds creation, track following
- Parallelization on GPU of **track-seeding**
- Huge data multiplicity for a full-detector scan tracking: a GPU makes it feasible

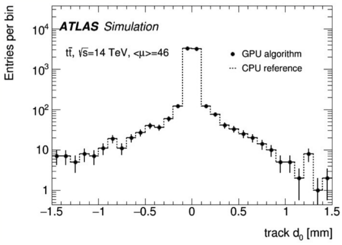


Pair formation: 2D thread array checking for pairing conditions

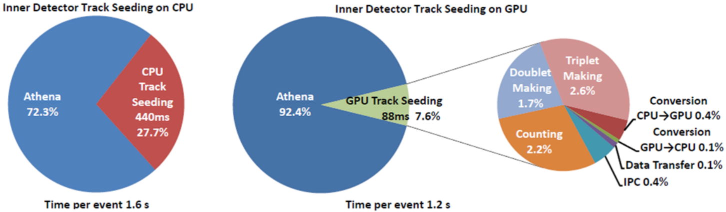
Triplet formation through 2D thread block

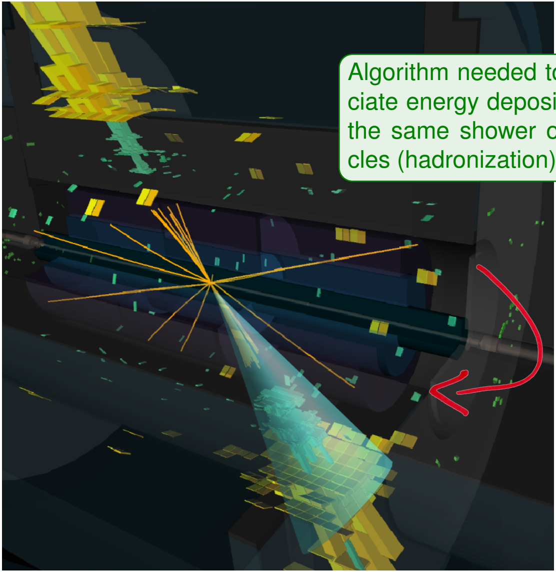
GPU algorithm has same efficiency and resolution as CPU one





- Algorithm execution time reduced by a factor $\sim$5
- Small data transfer overhead: $\sim$0.6%

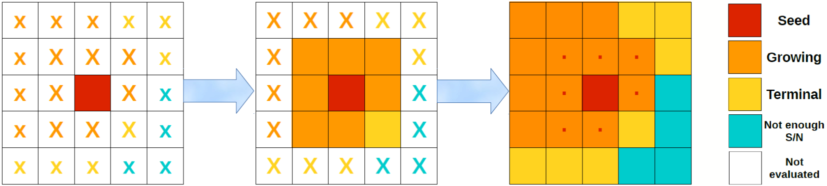




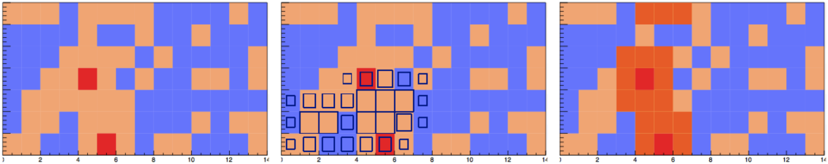
Algorithm needed to associate energy deposits from the same shower of particles (hadronization)

Calorimeter

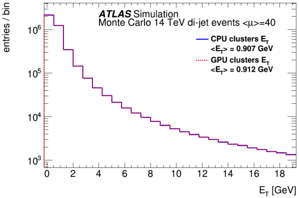
- Topological Calorimeter Cell Clusters reconstruction on CPU: $\sim$ 8% of total time
  - Cells are grouped according to their signal-to-noise ratio



- Topo-Automaton Clustering on GPU to maximize parallelism
  - Propagation of a flag through a grid of elements (cell pairs)
  - Cells get the largest flag and continue until no flag changes

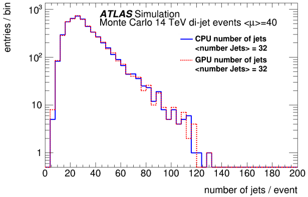


Calorimeter

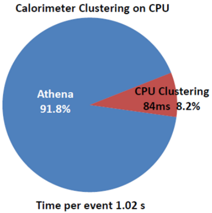


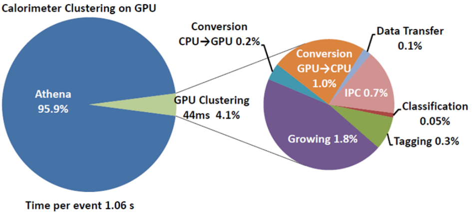
◀ Energy difference <5% in most clusters

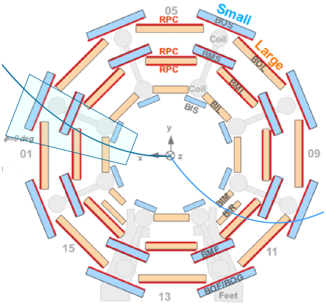
▶ no significant effect on jet reconstruction

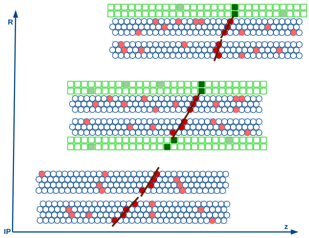


- 30% reduction for di-jet events with 40 interactions/bunch-crossing(μ), $\times$3 reduction for $t\bar{t}$ with μ=138
- Data-format conversion reduce the benefits
- Potential larger gain from parallelization of following clusterization steps

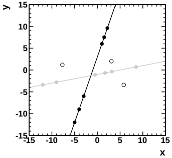


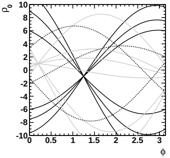


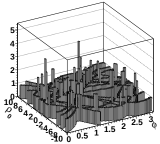




- ▶ Muon segment reconstruction through Hough Transform
  - algorithm translates track finding to maxima finding
  - Filter hits and fill Hough parameter space
  - Select maxima above a given threshold and reconstruct track parameters

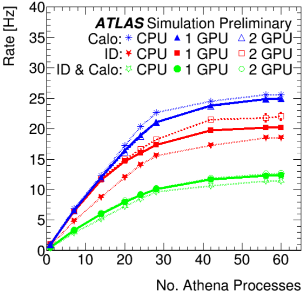


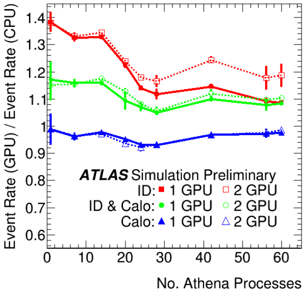


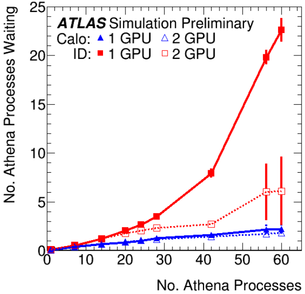


Development ongoing - public results expected soon

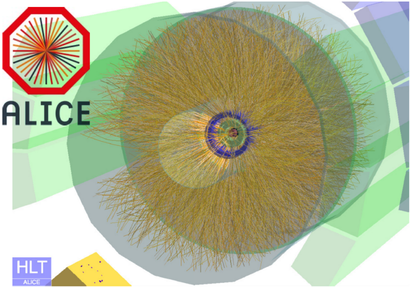
- Testing E5-2695 v3 14-core vs. 1(/2) NVidia K80 GPU
  - 20-40% gain in throughput, depending on the number of processes running
  - 1 GPU saturation when serving 14 clients (no performance loss)
  - Slight benefit from the additional GPU

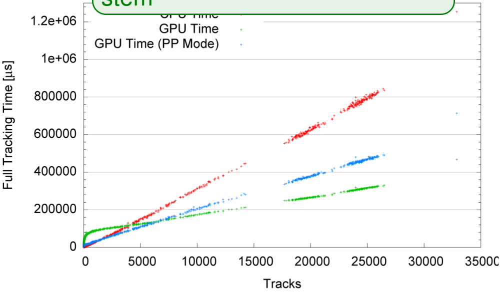






1s $\rightarrow$ 300 ms reduction by deploying GPU in TPC tracking system

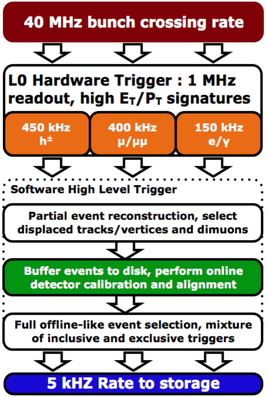




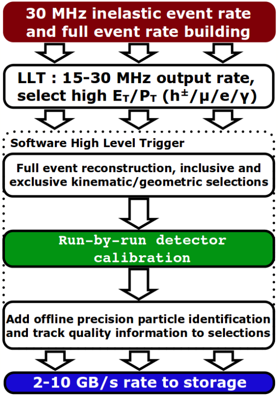
Considering improvements and modification to the trigger scheme for the experiment upgrades.

more info in talk from D. Rohr

LHCb - toward a triggerless approach

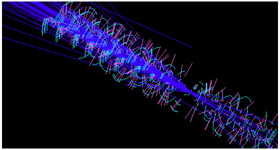


- For HL-LHC aim at a triggerless scheme (no hardware)
- GPU deployment can boost software trigger level
- Evaluation in progress to minimize communication latencies and throughput

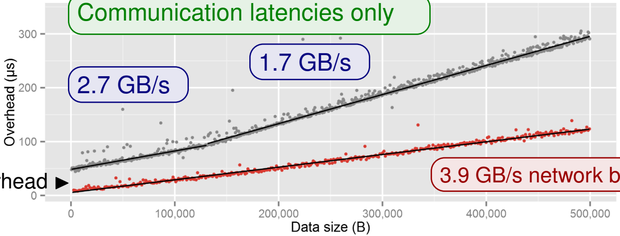


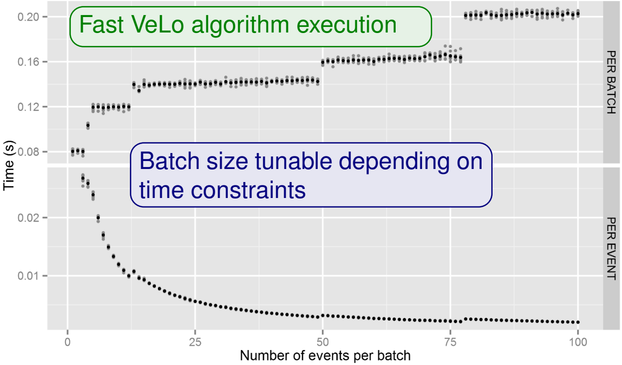
Focusing on vertex reconstruction and tracking algorithm ▶

- VErtex LOcator detector fundamental for displaced vertices detection
- Tracking in muon detectors

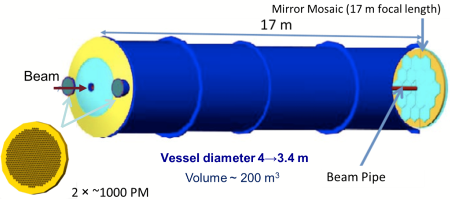


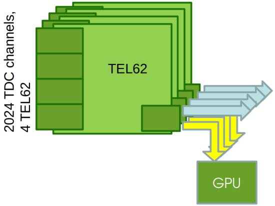
LHCb performance tests

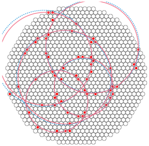


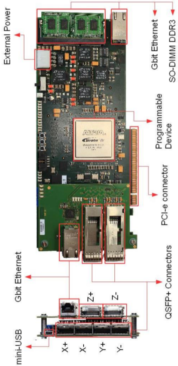


NA62 - a low-level trigger application

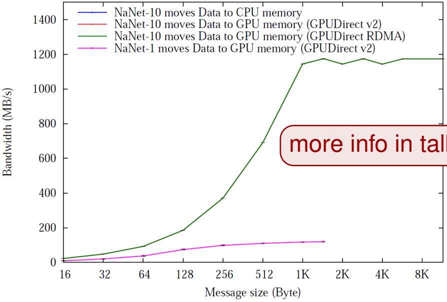








Improved RDMA scheme allowed to increase throughput and deploy GPU in **low-level trigger** (smaller event size)



more info in talk from A. Biagioni

Conclusions

- **Parallelism in realtime selection system is a must: GPUs deployment is crucial**

- GPU integration can be achieved in a transparent way
  - Client-Server architecture: the most flexible solution for DAQ existing frameworks
  - New experiment can deploy different scheme, no constraints
  - Careful design of EDM needed

- Communication overhead latencies define the feasibility domains
  - High-level trigger applications accessible for typical HEP experiment sizes (latencies $\mathcal{O}(100\ \mu s - 100ms)$)
  - From the detector to the GPU in Low-level trigger application, achieved thanks to dedicated interface cards

- Algorithm optimization can add gain in parallelization
  - Several developed for pattern recognition algorithms (Hough Transform, Cellular Automaton, ...)
  - Neural Networks (and MVA) might come into the game in the future