Studies on many-core devices in the HLT trigger

S.Amerio*°, G.Collazuol*°, M.Corvo^°,
S.Gallorini*°, A.Gianelle°, D.Lucchesi*°

*University of Padova, ^University of Ferrara, °INFN

Meeting LHCb Italia
Frascati 13-14 October 2015

Introduction

High Level Triggers in LHC experiment are based on *farms of CPUs*

- easier maintenance and upgrade
- the HLT code can be easily simulated offline
- HLT code as much similar to the offline as possible
- additional computing power for offline processing when the experiment is not taking data

New challenges in the future:

- LHC upgrade → Increased event rates and pile-up → more computing power needed
- We are facing a technological change: many-cores, parallel computing.

New data transfer technologies allow the application to low latency environments.

In this talk:

- *Which solutions are being explored in the other experiments?*
- *LHCb strategy for Run3*
- *Ongoing work on many-core devices in LHCb*

Alice

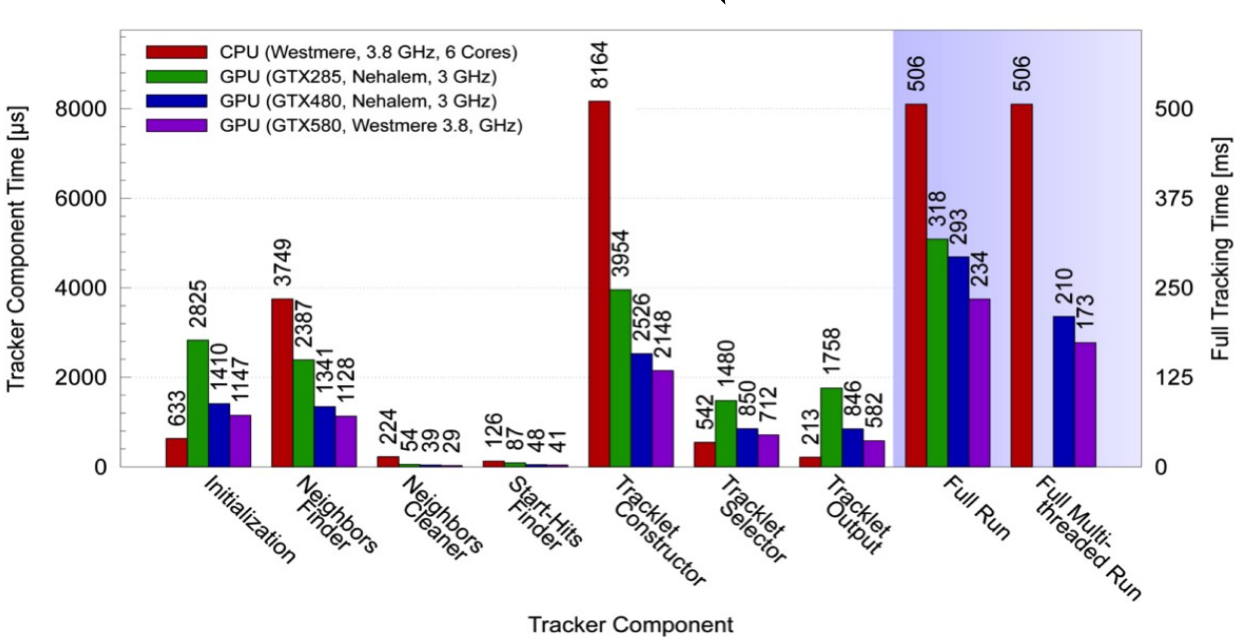
Run 3 expectations: event rate x 100, data volume x 10 wrt current rates.

Data reduction to manageable levels thanks to a dedicated computing facility which incorporates DAQ/HLT and Offline functionalities:

- Data volume reduction
- Heterogeneous hardware (CPU/GPU/FPGA), well tested approach in Run2
- New framework (ALFA) to incorporate all tasks

TPC tracking most time consuming step in the HLT

- Neighbor Finding, Tracklet Construction and Tracklet Selection on GPU;
- Inizialization, Tracklet Output on CPU.
- *Overall total processing time from 500 ms to 170 ms.*

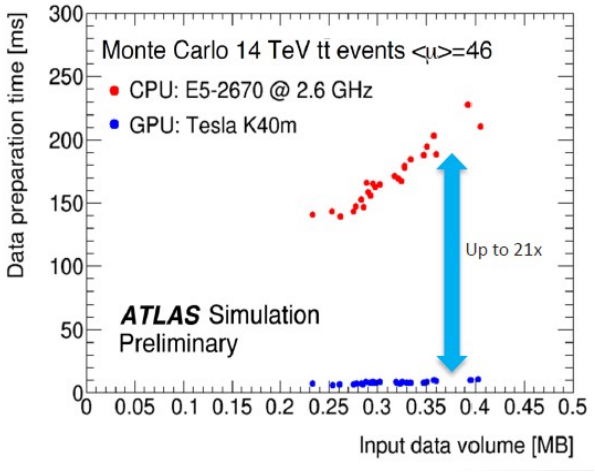


Atlas

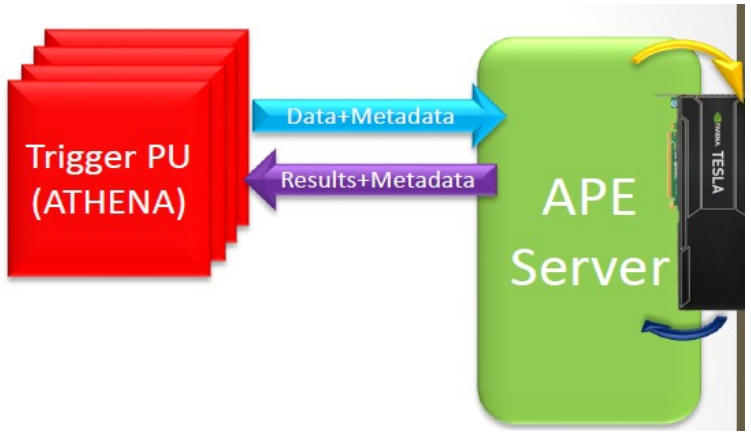
Studying the inclusion of GPUs in the HLT. Issues under study:

- Integration of GPUs in Atlas framework
- Development of high-parallelizable algorithms

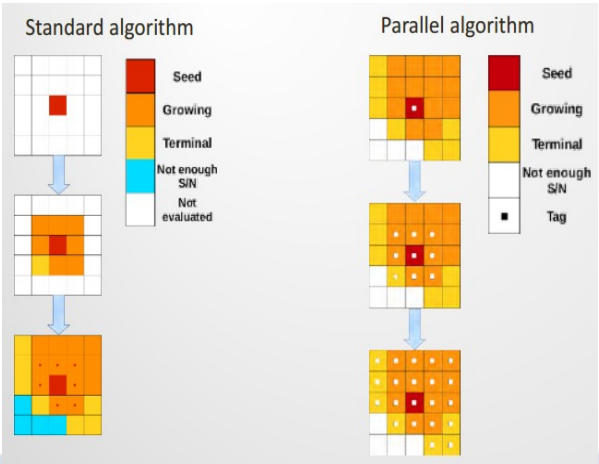
Inner detector data preparation on GPU.



Client/server approach.
Accelerator Process Extension (APE)
Server manages offload requests and executes kernels on GPU(s)

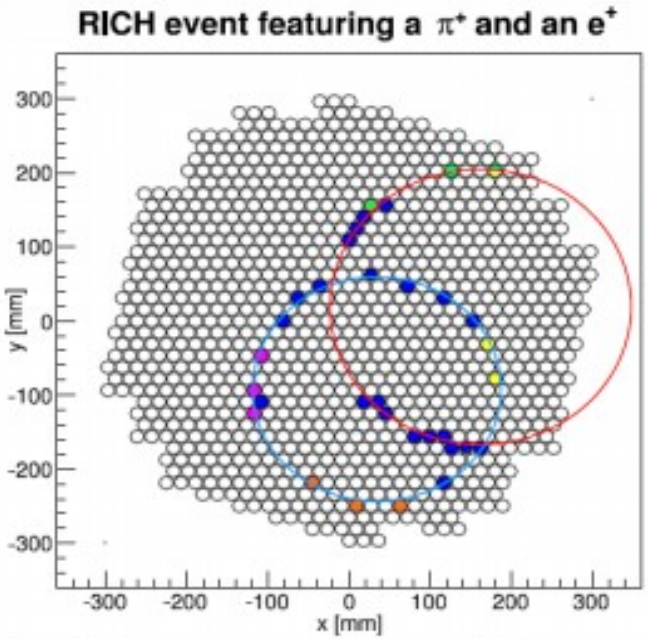


Calorimeter clustering



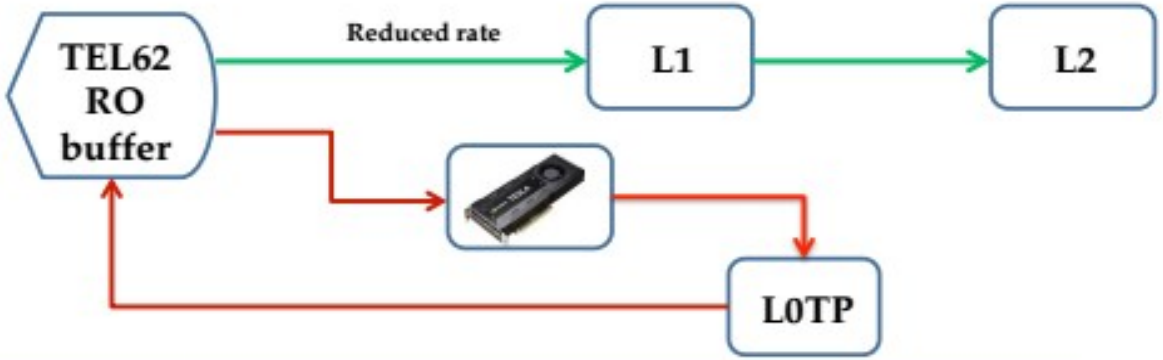
Same solution adopted at LHCb, more details later.

NA62 RICH: L0 GPU-trigger



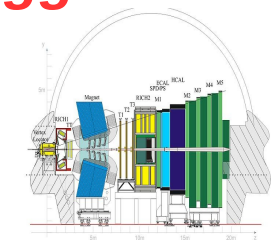
- RICH: π/μ separation at 15-35 GeV/c
  - 10 MHz events rate (~20 hits/track)
- 1 MHz L0 trigger rate (1 ms maximum latency, synchronous system)
  - Baseline: PMT multiplicities from TEL62
  - Possible upgrade: trackless multi-ring fitting using GPUs (fast and non-iterative)
- NaNet: PCIe interface with GPU Direct P2P/RDMA capability

- Demonstrator with NVIDIA Tesla K20, NaNet 1Gbps and TTC
  - Currently taking data in NA62 in parasitic mode
  - Nanet 10Gbps upgrade soon



LHCb strategy for Run 3

Triggerless readout system* followed by a *full software trigger



40 MHz

DAQ
(custom electronics)

40 MHz

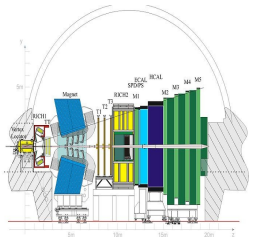
Software LLT on EB CPU
Event building on CPU

1-40 MHz

HLT
(CPU only)

20 kHz

Storage



40 MHz

DAQ
(custom electronics)

40 MHz

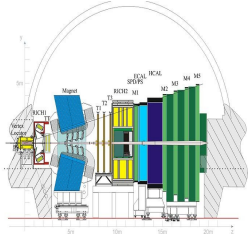
Pre-processing with GPUs
in EB nodes.
Event building on CPU

1-40 MHz

HLT
(CPU only)

20 kHz

Storage



40 MHz

DAQ
(custom electronics)

40 MHz

Pre-processing with GPUs
in EB nodes.
Event building on CPU

1-40 MHz

HLT with CPU/GPU, e.g. to
speed up particle identification
algorithms

20 kHz

Storage

Trigger and DAQ TDR
https://cds.cern.ch/record/1701361?ln=en

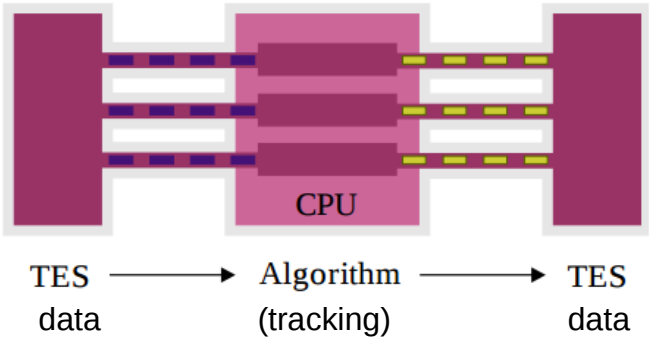
Ongoing activities on GPUs @ LHCb

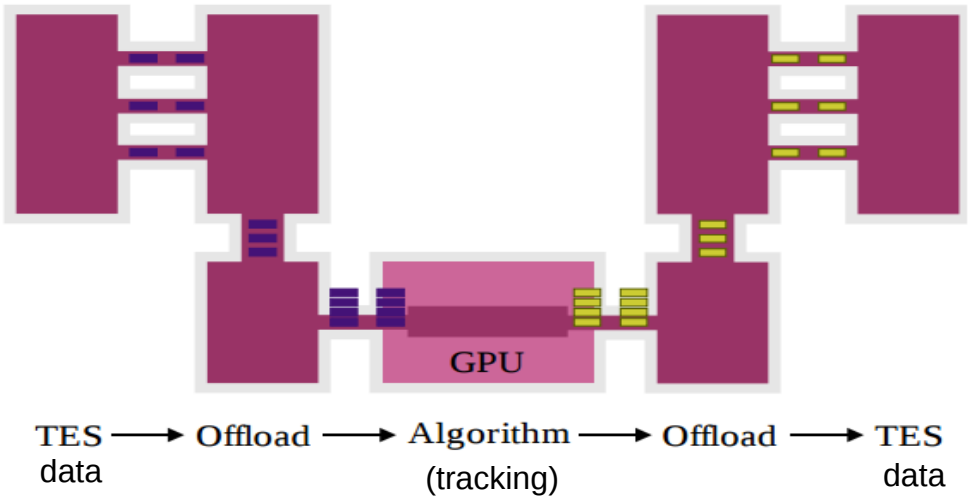
Infrastructure

How do we send/retrieve data to/from the GPU?

- Development and test of a co-processor manager to use many-core devices within LHCb framework *(Cern)*
- Setup of a testbed in the online system --> test during Run2 in real data taking conditions

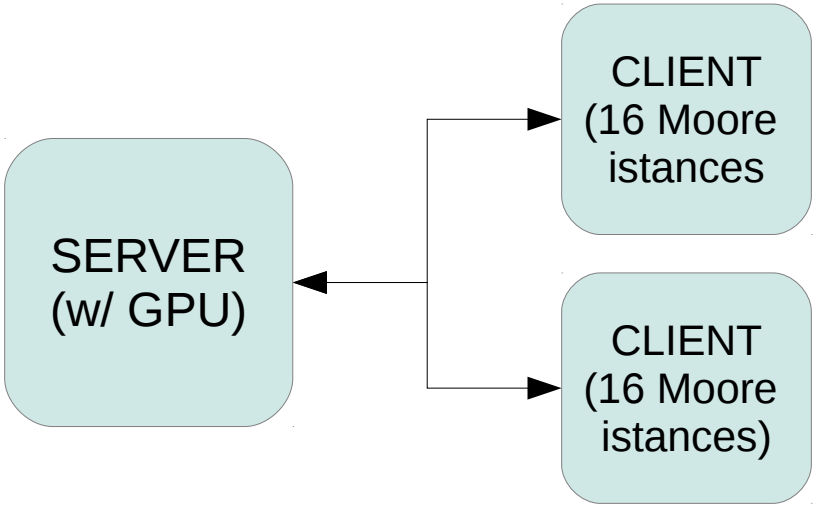
Coprocessor manager





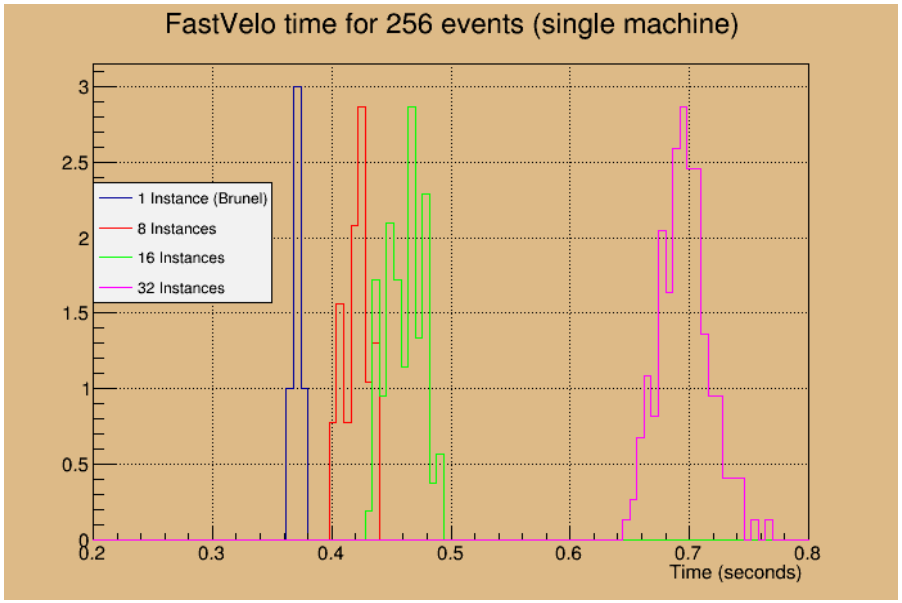
- Server/Client tool necessary to submit algorithms to GPU from the LHCb framework to offload data.
- Recently added network support, TCP-IP, to send data between different machines across a network.
- Other communication methods, such as Infiniband, could be added
- It can be used to submit to any accelerator

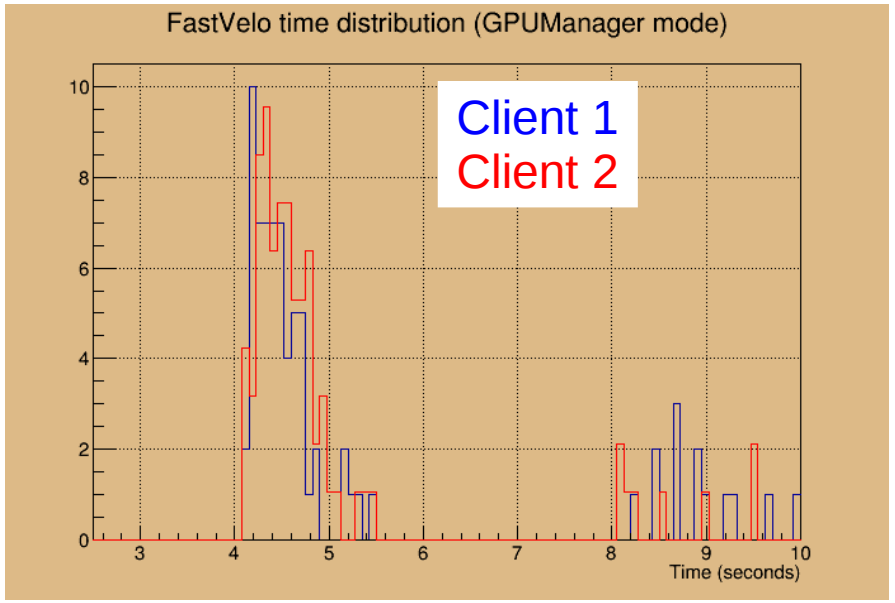
Coprocessor manager - first tests



Overhead negligible with client/server on the same machine.

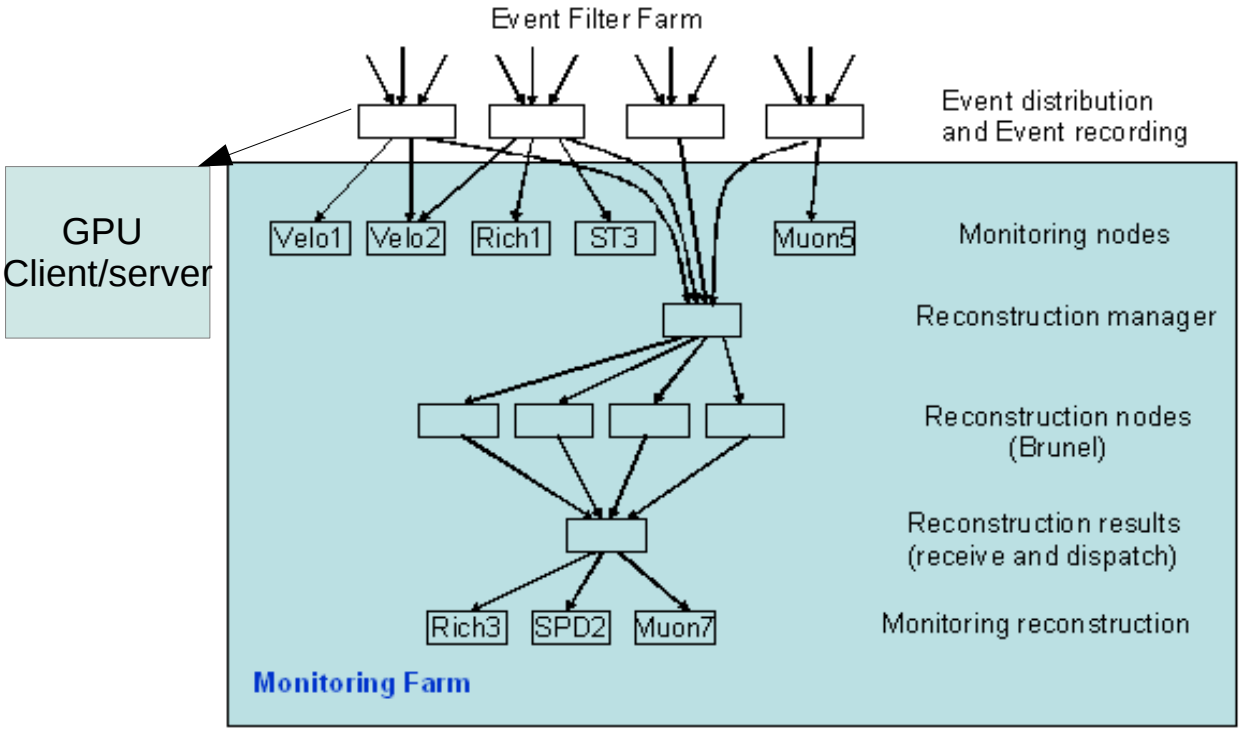
Significant overhead (~ 3 s) when client and server are on separate machines.





GPU testbed

- Working to test *coprocessor manager* + *tracking algorithms* in real data taking conditions.
- First tests in the monitoring farm: PC equipped with a NVidia Titan GPU installed as additional monitoring node, but running as HLT node.
- Goals:
  - Include new device in the online system
  - Real time comparison of CPU/GPU based algorithms



Ongoing activities on GPUs @ LHCb

Infrastructure

How do we send/retrieve data to/from the GPU?

- Development and test of a Co-processor manager *(Cern)*
- Setup of a testbed in the online system --> test during Run2 in real data taking conditions

Algorithms

- Current detector:
  - Porting to GPU the current HLT track reconstructions algorithms (FastVelo + PatForward)
  - Development of algorithm (Cellular automata) for T stations
- Upgrade
  - Development of algorithms for the upgraded Velo, Patpixel *(Cern)*

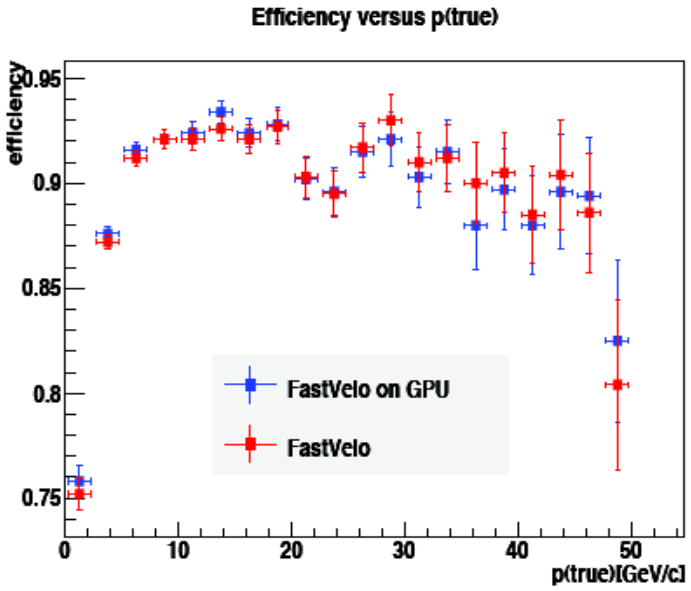
FastVelo

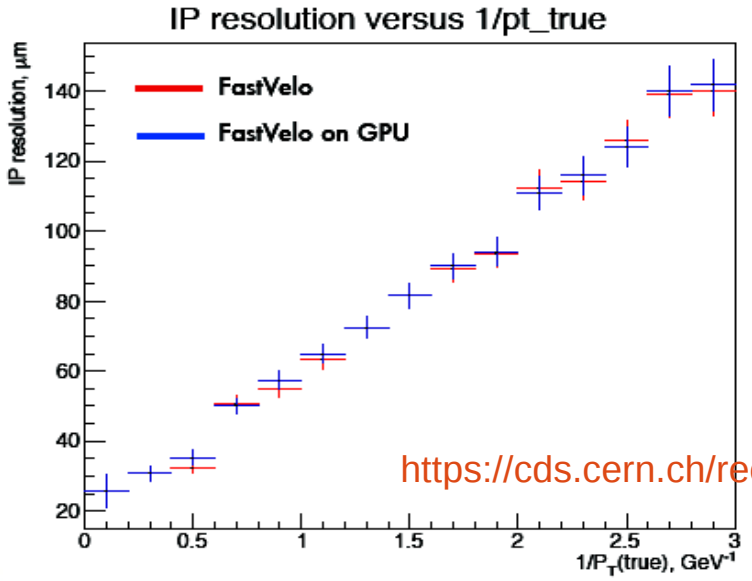
- FastVelo CPU algorithm ported to GPU (simple parallelization over *for* loops; no optimization of the code yet).
- Efficiencies and resolution as in the CPU algorithm
- x 3, x 2 speedup wrt a single core.

B inclusive decays				
Run condition 2015 25ns 2.6nu	FastVelo on GPU		FastVelo	
HLT1 Only	Efficiency	Clones	Efficiency	Clones
VELO, all long	87.1%	0.3%	89.3%	0.5%
VELO, all long, p>5 GeV	90.0%	0.2%	92.3%	0.4%
VELO, all long B daughters	87.3%	0.2%	89.0%	0.8%
VELO,all long B daughters, p>5 GeV	89.5%	0.2%	90.9%	0.8%
VELO, ghosts	8.6%		7.9%	

CPU: Intel(R) Core(TM) i7-3770 3.40 GHz

GPU:NVidia GTX Titan





https://cds.cern.ch/record/1698101?ln=en

Forward tracking

- Forward tracking algorithm (PatForward) is being ported to GPU (as for Velo, simple parallelization whenever possible)
- Efficiencies lower than CPU algorithm due to memory limits (need to rethink event structure)
- Timing and resolution under evaluation.

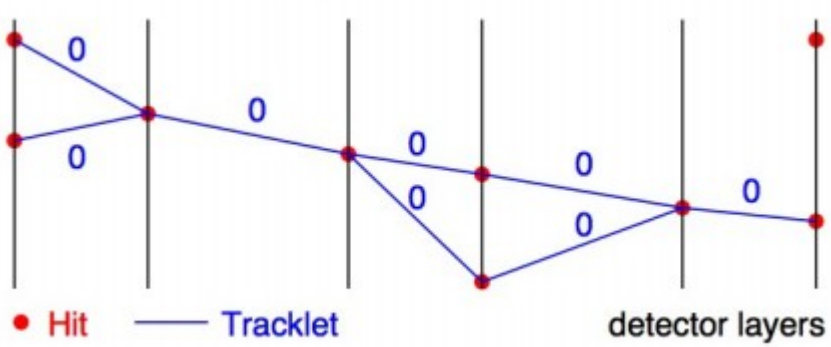
Very preliminary

Track category	Forward on CPU (OT only)		Forward on GPU (OT only)	
	Efficiency	Clones	Efficiency	Clones
Forward, all long	60.0 %	0.2 %	50.9 %	0.1 %
Forward long, p> 5 GeV	60.7 %	0.1 %	56.5 %	0.1 %
Forward, all long B daughters	66.5 %	0.1 %	60.4 %	0.1 %
Forward, all long B daughters, p> 5 GeV	68.1 %	0.1 %	63.5 %	0.1 %
Forward, ghosts	41.6 %		38.0 %	

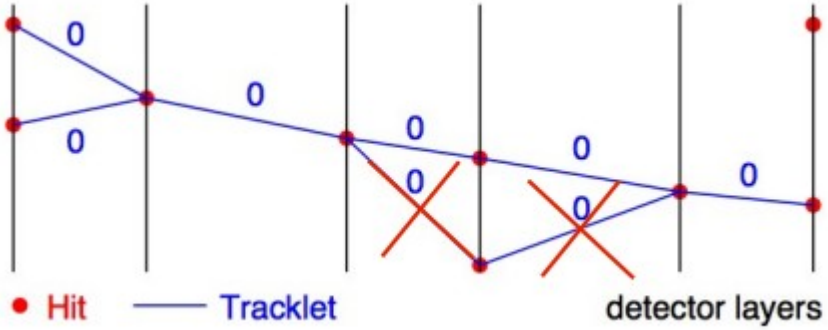
1000 MC events $B_s \rightarrow \varphi\varphi$ (2015, ν 1.6, 25 ns)

Cellular automata

Tracklets (segments between two adjacent layers, cells) generation; counter set to 0.

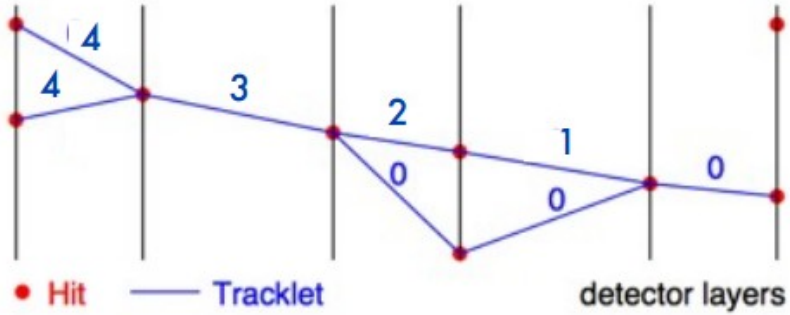


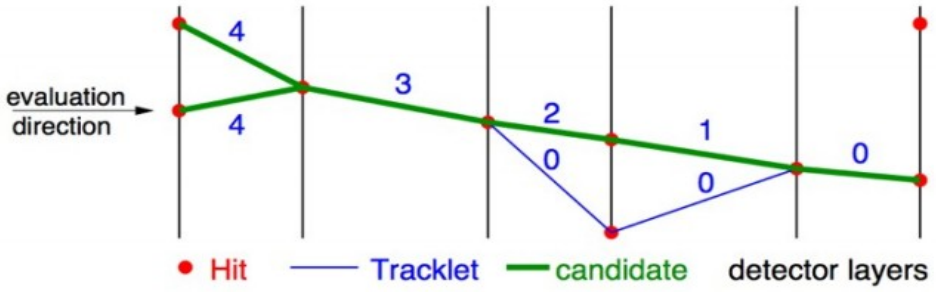
Neighbour finding: connect adjacent tracklets according to the track model



Evolution and track formation:

- all cells evolve in parallel in time steps according to an evolution rule
- Rule: each cell counter is incremented if one of its neighbours has a counter which is at least as high as its own counter.





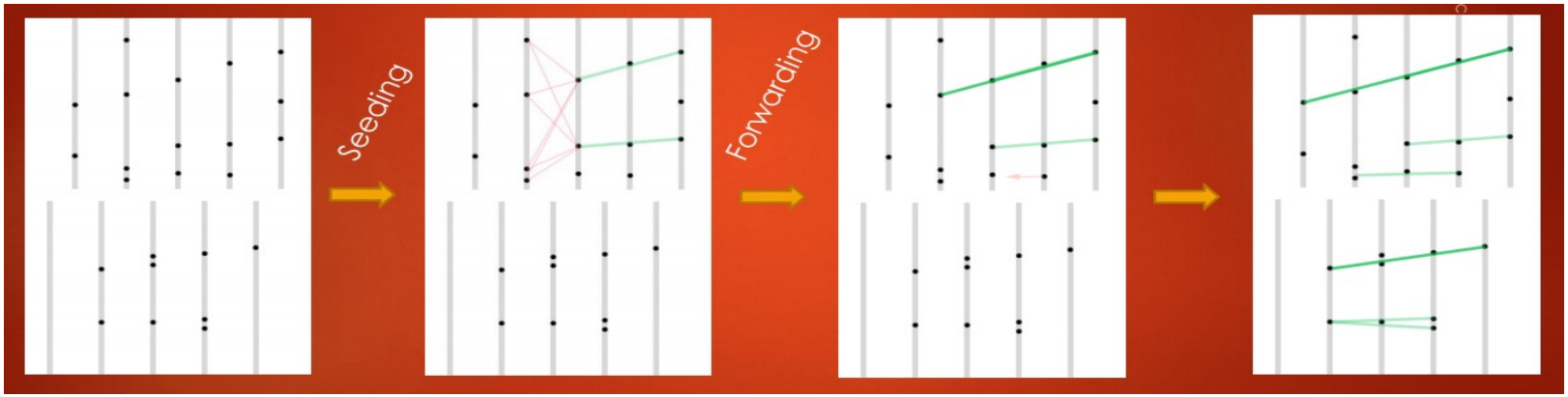
Cellular automata

Algorithm developed on CPU so far. OT only.
Now being ported to GPU.

Track category	PatSeeding		Automata	
	Efficiency	Clones	Efficiency	Clones
T-Tracks, all long	74.2 %	0.0 %	68.9 %	0.0 %
T-Tracks long, p> 5 GeV	66.7 %	0.0 %	61.9 %	0.0 %
T-Tracks, all long B daughters	77.6 %	0.0 %	72.1 %	0.0 %
T-Tracks, all long B daughters, p> 5 GeV	74.6 %	0.0 %	69.1 %	0.0 %
T-Tracks, ghosts	20.3 %		30.3 %	

500 MC events $B^0 \rightarrow K^*\mu\mu$, 25 ns, ν 1.6

Velo pixel tracking on GPU

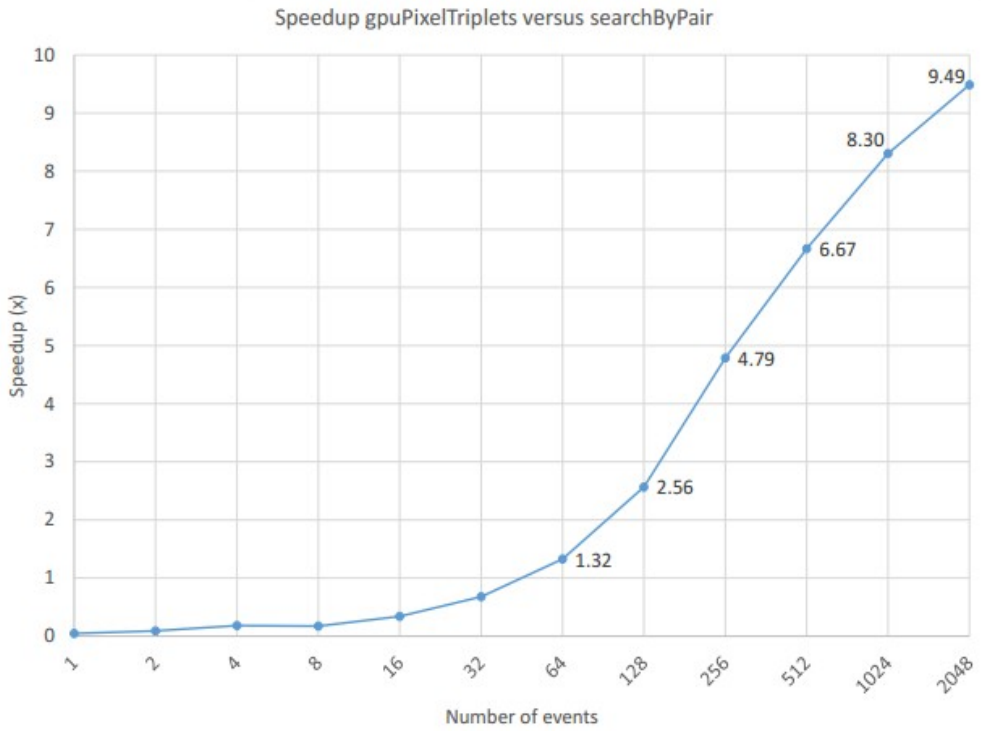


Triplets of clusters in neighbouring sensors are selected as seeding, forwarding is done in parallel.

Same efficiencies and ghost rate as CPU algorithms.

x 9 speedup compared to single CPU.

Need to process data in large batches to obtain maximum gain from parallel devices.



Conclusions

The application of many-core devices is already a reality in some experiments, while being explored by others.

Main challenges
- adapt analysis/online frameworks to work with heterogeneous system;
- data movement and event structure.

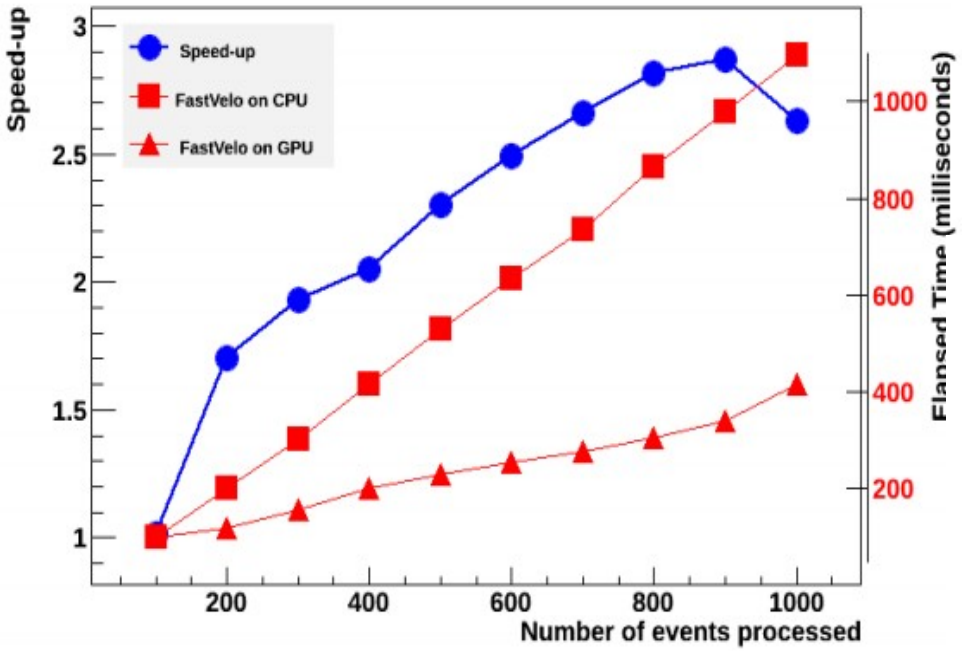
At LHCb working to demonstrate the feasibility of this application, understand strenghts and weak points in view of Run 3:
- current tracking algorithms ported to GPU
- developed tool to interface LHCb framework with new devices
- test bed for tests in online environment in preparation.

In parallel, development of parallel algorithm for the upgraded VELO.

BACKUP

FastVelo timing

Figure 6 Tracking execution time and speedup versus number of events using a 2012 MC sample of $B_s \to \phi\phi$ decays ($\nu = 2.5$). The GPU is compared to a single CPU core (Intel(R) Core(TM) i7-3770 3.40 GHz).



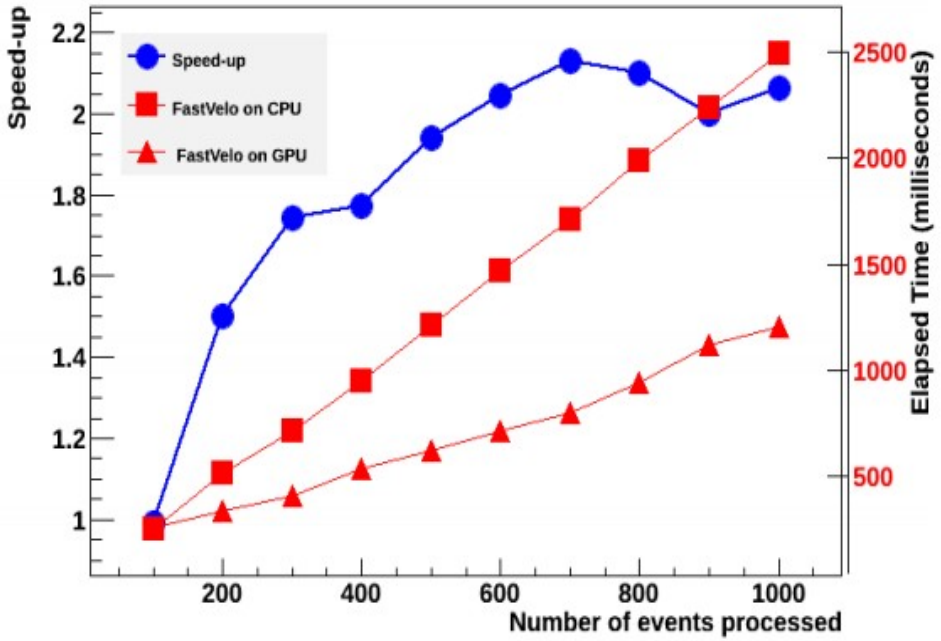


Figure 8 Tracking execution time and speedup versus number of events using a 2015 MC sample of b-inclusive decays generated with $\nu = 4.8$. The GPU is compared to a single CPU core (Intel(R) Core(TM) i7-3770 3.40 GHz).