

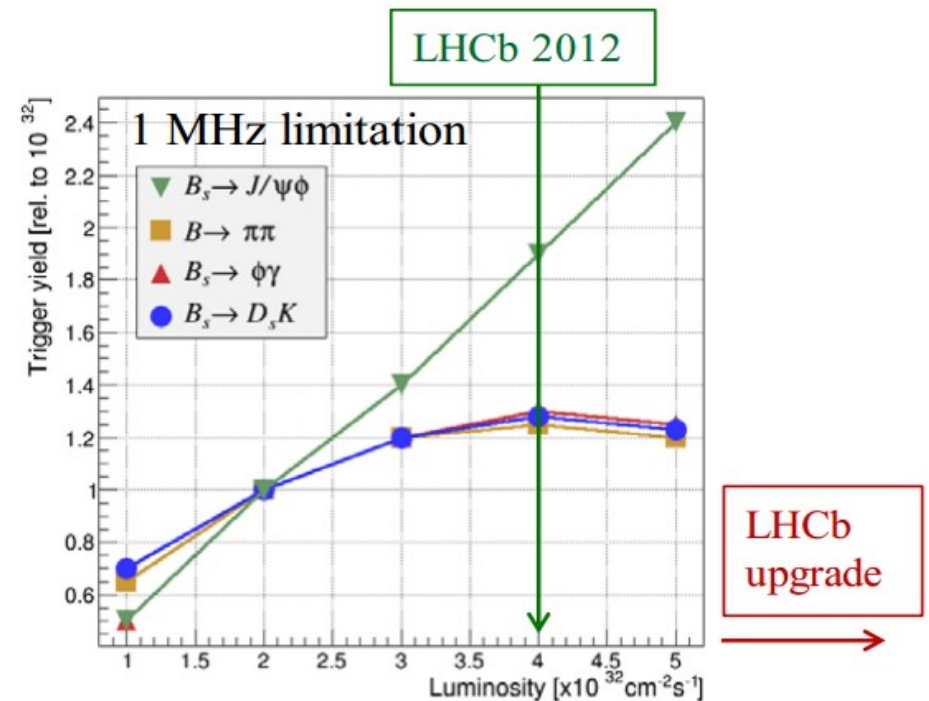
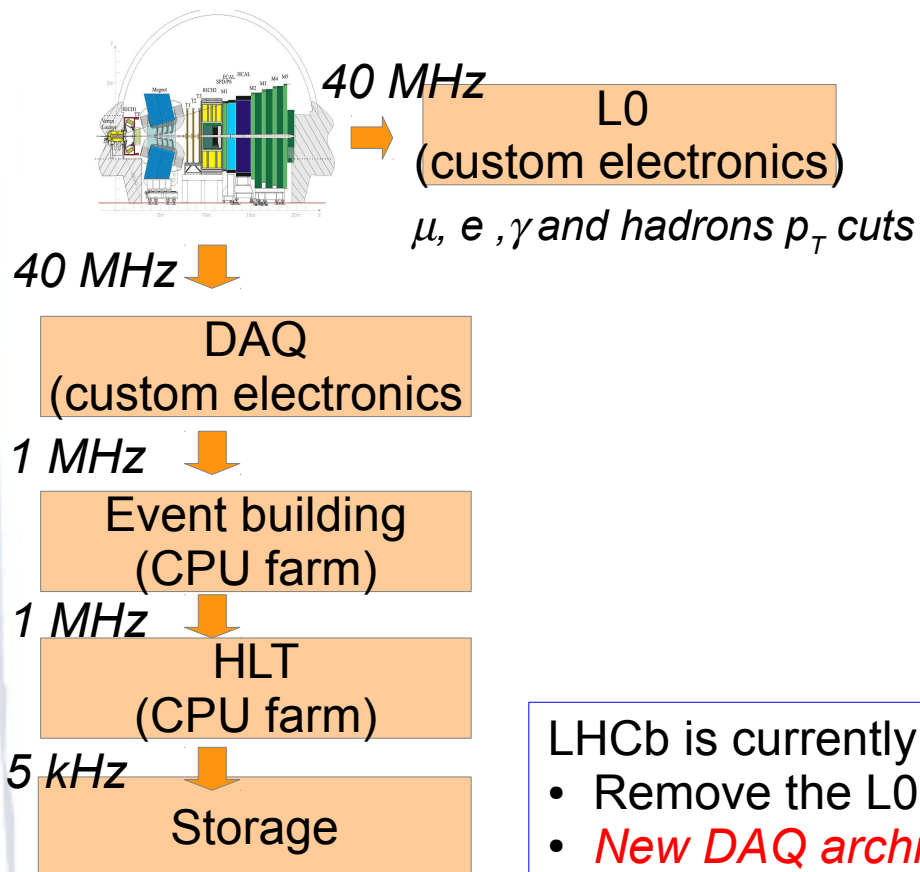
# LHCb trigger upgrade and data preservation activities in Padova

Silvia Amerio

for the LHCb group

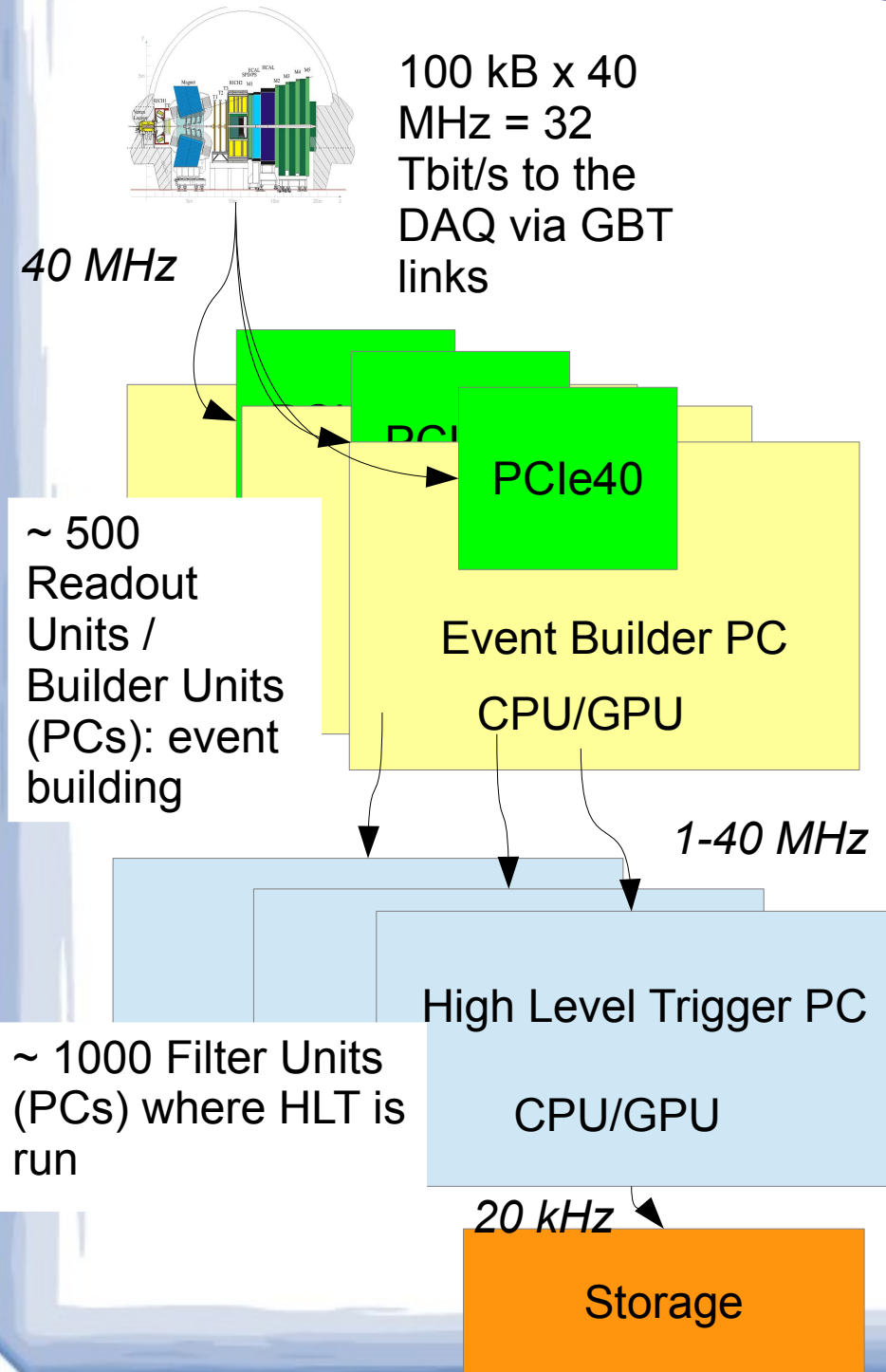
# LHCb upgrade: motivations

No deviations from the SM seen yet, we need to **probe NP at scales  $\gg 1$  TeV**  $\rightarrow$  **increase the precision** of the measurements  $\rightarrow$  **increase the size** of the collected signal samples  $\rightarrow$  **LHCb upgrade to collect up to  $50$  fb $^{-1}$**



- LHCb is currently limited by trigger and DAQ. Upgrade plan:
- Remove the L0 bottleneck  $\rightarrow$  read out full detector at 40 MHz
  - **New DAQ architecture (PD)**
  - **Software Low Level Trigger (LLT)** for rate control out of DAQ
  - **New EB/HLT (PD)**

# LHCb trigger upgrade

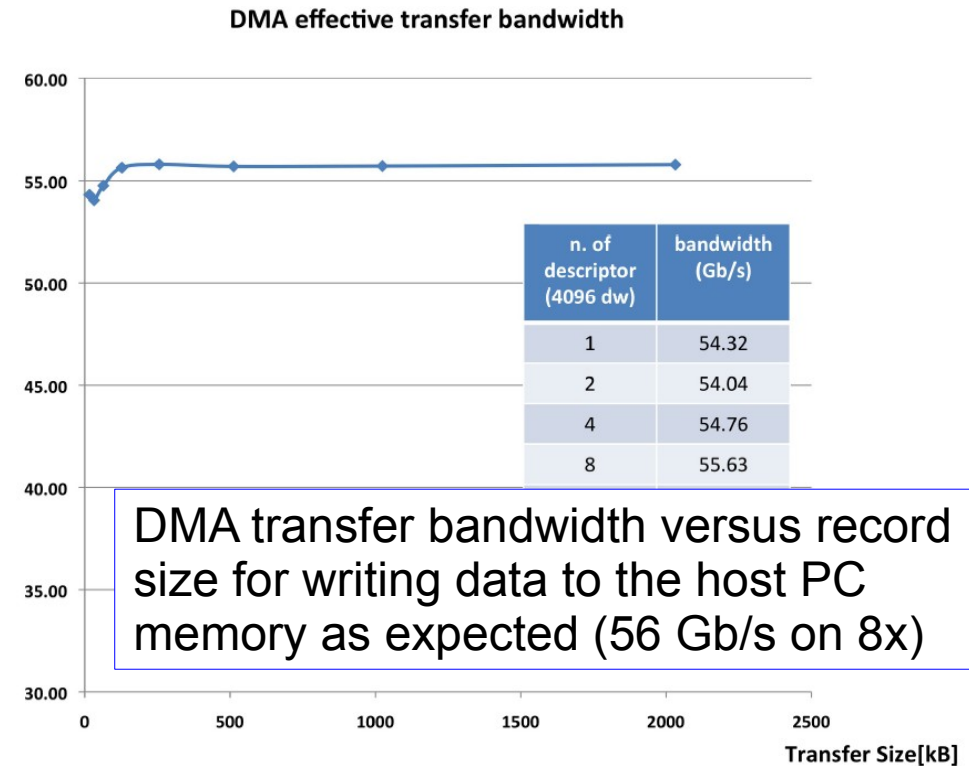
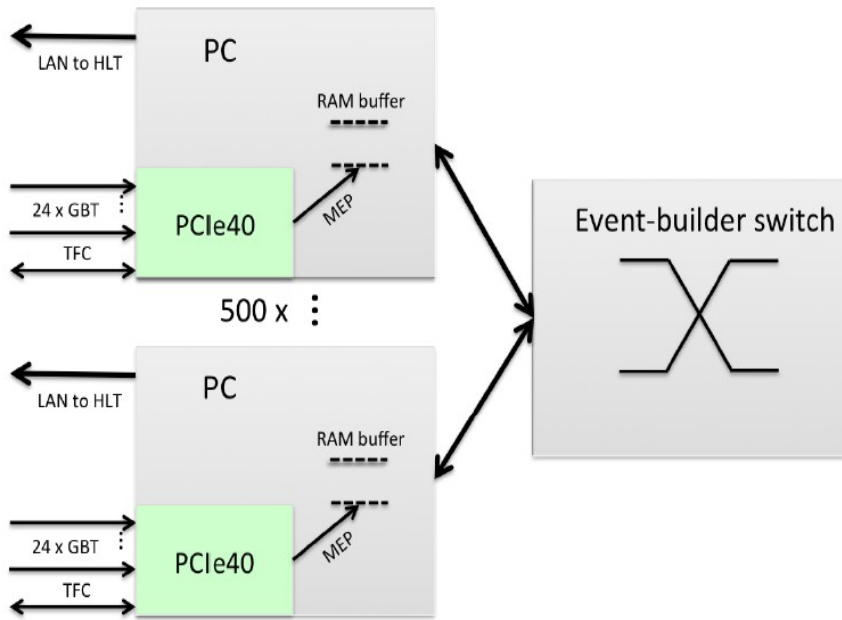


- PCIe40 = PCIe card to push data into the main memory of the hosting PC
- Low Level Trigger on EB CPU (*eventually can be removed*)
- Very flexible system: new technologies (e.g. GPUs) can be added both in EB and HLT units
- New and optimized HLT algorithms

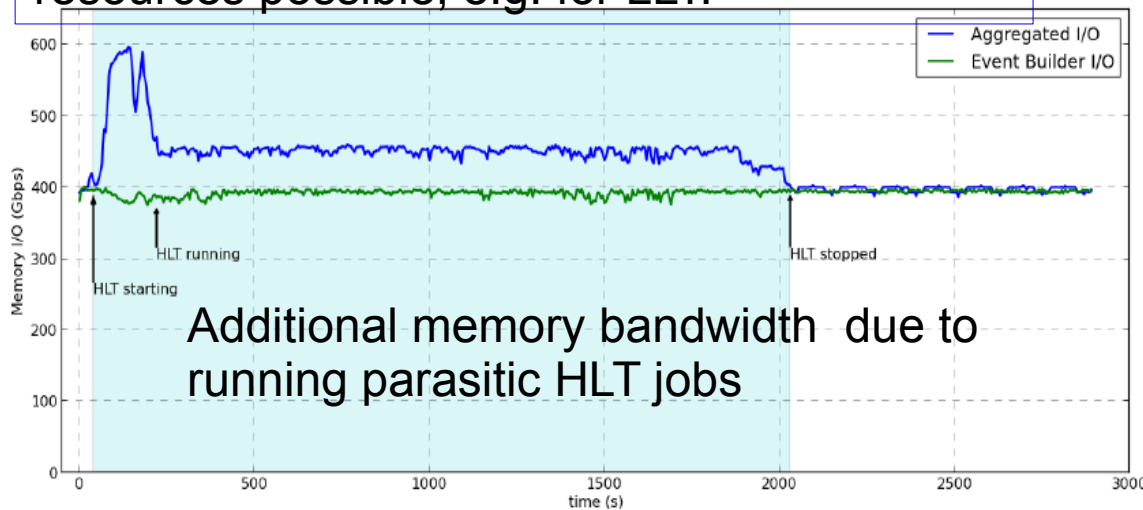
**40 MHz triggerless readout!**

- ✓ TDR published on 21st May 2014  
<https://cds.cern.ch/record/1701361?ln=en>
- ✓ This project has been approved by INFN CTS

# DAQ upgrade: PCIe based RO and EB



EB performs stable at 100 Gb/s on one node (total throughput 400 Gb/s on a single server)  
Parasitic usage of the event builder units resources possible, e.g. for LLT.



## Advantages of PCIe:

- It is the interconnect choice for many computing and network technologies (GPU cards, IB or 40Gb Ethernet, solid state disks,..)
- 4<sup>th</sup> generation under development - backward compatible.

# HLT upgrade

## Requirements:

- Increase efficiency on hadronic modes

mode	D $\rightarrow$ hhh	B $\rightarrow$ hh
$\epsilon(\text{L0})$ [%]	27	62
$\epsilon(\text{HLT/L0})$ [%]	42	85
$\epsilon(\text{tot})$ [%]	11	52

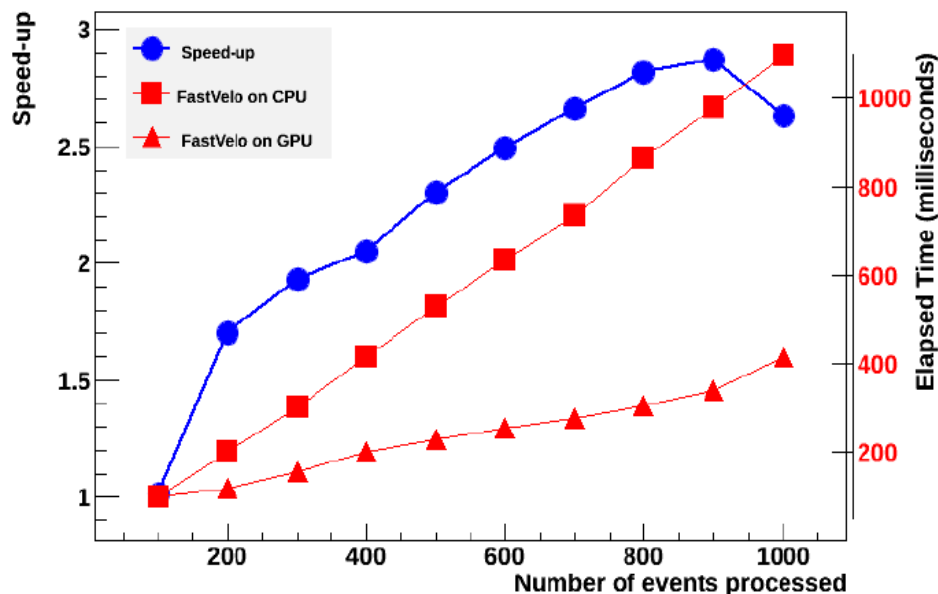
- After the upgrade background rejection is not the main problem anymore, 25% event will contain a b or c quark --> *we need to better categorize signal while reducing the background* --> We need the maximum available information to distinguish between signals
- Ideal solution: No “trigger”, just offline selections in the online farm
- BUT assuming a CPU farm 10x the current one the time budget for HLT  $\sim$  15 ms/event --> we need fast reconstruction algorithms
- *R&D on many-core computing ongoing*
- *New and optimized HLT algorithms*

# Track reconstruction on GPUs

Current VELO (Vertex Locator detector) tracking algorithm has been ported on CUDA and run on NVIDIA GPUs:

Track category	FastVelo on GPU		FastVelo	
	Efficiency	Clones	Efficiency	Clones
VELO, all long	86.6%	0.2%	88.8%	0.5%
VELO, long, $p > 5$ GeV	89.5%	0.1%	91.5%	0.4%
VELO, all long B daughters	87.2%	0.1%	89.4%	0.7%
VELO, long B daughters, $p > 5$ GeV	89.3%	0.1%	91.8%	0.6%
VELO, ghosts	7.8%		7.3%	

B- $\rightarrow\phi\phi$  events



Efficiencies, ghost and clones rates and parameter resolutions (backup) comparable to current algorithm.

Factor 3 speed-up compared to a single core (Nvidia Titan vs Intel i7, 3.40 GHz)

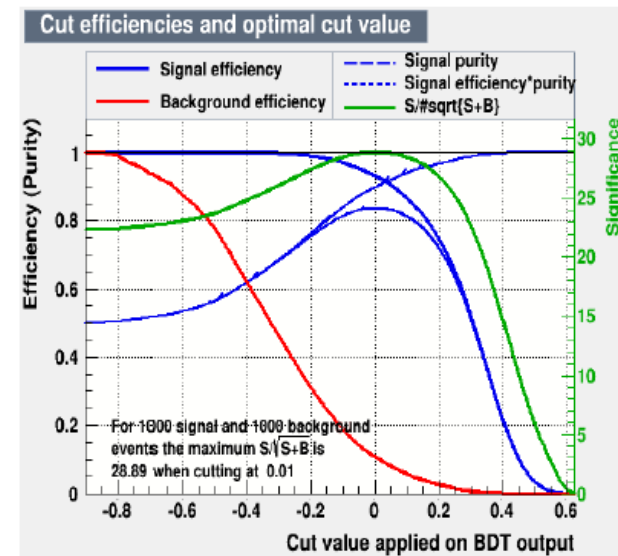
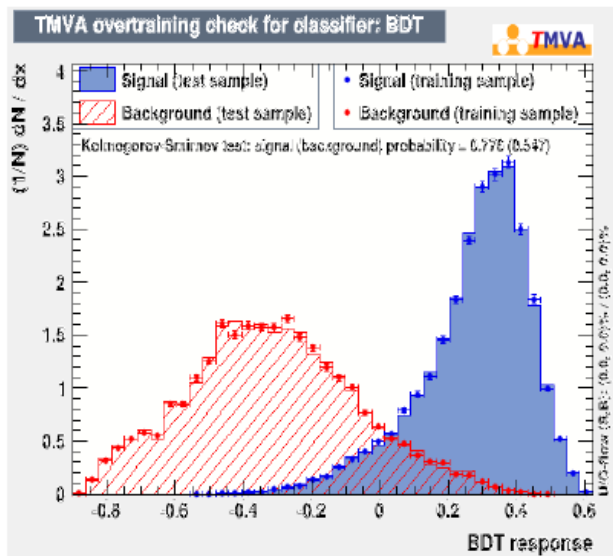
*Ongoing: add the forward tracking*  
*Short term goal: full tracking on GPUs in parasitic tests during 2015*

# Improving HLT algorithms

All HLT selection algorithms are being re-tuned for the upgrade.  
New algorithms under development.  
Use of multivariate techniques where possible to increase classification purity.

Inclusive charm trigger based on Boosted Decision Tree under development in Padova:

- Goal: selection of events with the production of a  $D^*$
- Main decay channel:  $D^*(2010)^+ \rightarrow D^0 \pi_{\text{sl}}$
- Several  $D^0$  decays considered.



# Trigger: summary of Padova activities

## DAQ

**People:** Gianmaria Collazuol, Marco Bellato, Fabio Montecassiano

### **Activities:**

- R&D for the **implementation of the PCIe-gen3 data transfer** protocol in the Readout Board FPGA
- development of the Linux **drivers for high-speed data transfer** into the RO/EB PC
- evaluate the possibility of implementing an **early HLT stage** in the RO/EB PC (possibly using GPUs)
- functional **test protocol definition** for card validation at manufacturer and at pit

## HLT

**People:** Silvia Amerio, Stefano Gallorini, Alessio Gianelle, Donatella Lucchesi, Anna Lupato, Emanuele Michielin, Andrea Rugliancich, Lorenzo Sestini

### **Activities:**

- **Coordination** of the R&D on many-core technology group (S.Gallorini, Cern simifellow)
- Porting of current **Velo track reconstruction algorithm** to GPU
- Porting of **full track reconstruction algorithm** to GPU
- **Tests in parasitic configuration**
- New **HLT selection algorithms**



# LHCb Long Term Future Data Preservation

# LHCb LTDP project

- Dedicated task force for long term future data preservation and open access
- Data preservation activities developed within DPHEP and in collaboration with other LHC experiments whenever possible
- Long term data preservation project organized in 5 work packages:

Data preservation

Software preservation

Validation Framework

Goal: Access and analysis of old (Run 1) data by LHCb collaboration

Analysis preservation

Goal: preserve all the information related to an analysis, from user code to final ntuples to internal and public documentation. The information preserved should allow a LHCb member to reproduce the analysis.

Open access  
& outreach

Goal: provide general public access to LHCb data (in simplified format and/or analysis level format)

# Activities in Padova

## Analysis preservation framework



Home / Deposit

### LHCb Data Analysis Mock-Up [DO NOT SUBMIT YET]

**NOTE:** Access to all submitted data will be restricted to the LHCb collaboration only.

Basic Info

Event Samples - Data

Event Samples - MC

User Code

Final N-Tuples

Internal Documentation

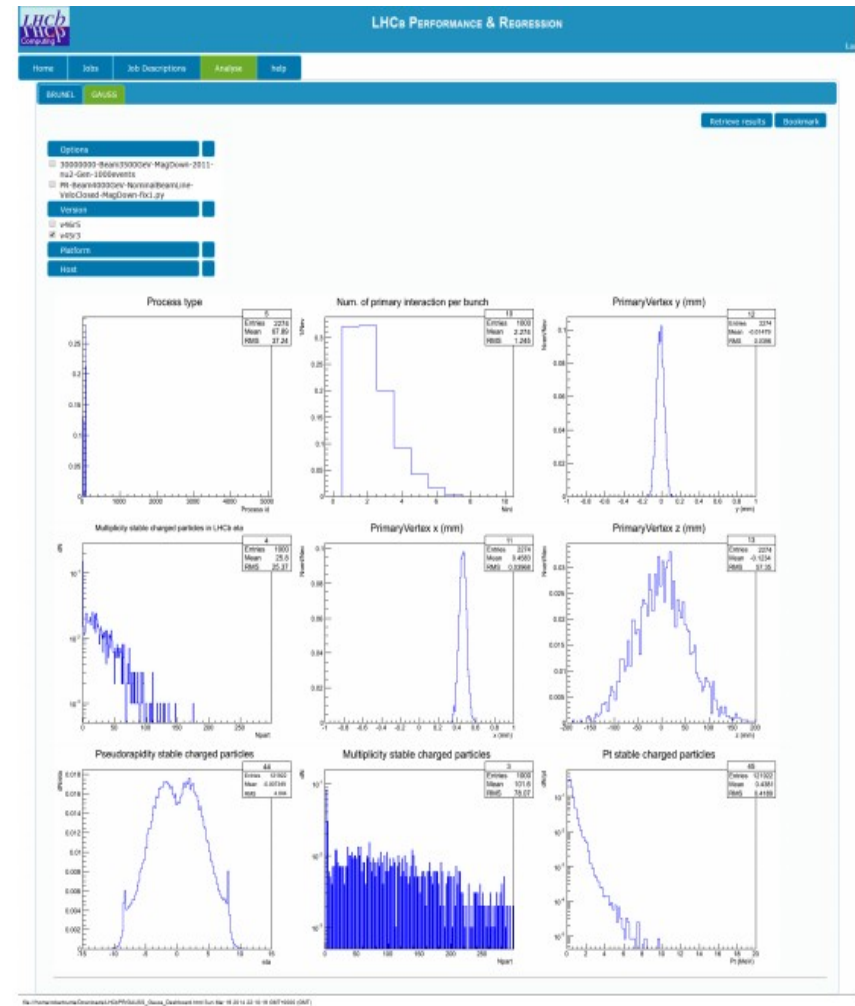
Internal Discussion

Presented already?

Published already?

Submit

## Validation system



# DP: summary of Padova activities

LTDP

**People:** Silvia Amerio, Alessio Gianelle, Mauro Morandin

**Activities:**

- **Coordination** of the LHCb DP task force (S.Amerio)
- Definition of **data and software legacy release**
- Development of long term future **validation system**
- Development of **analysis preservation framework**

**BACKUP**

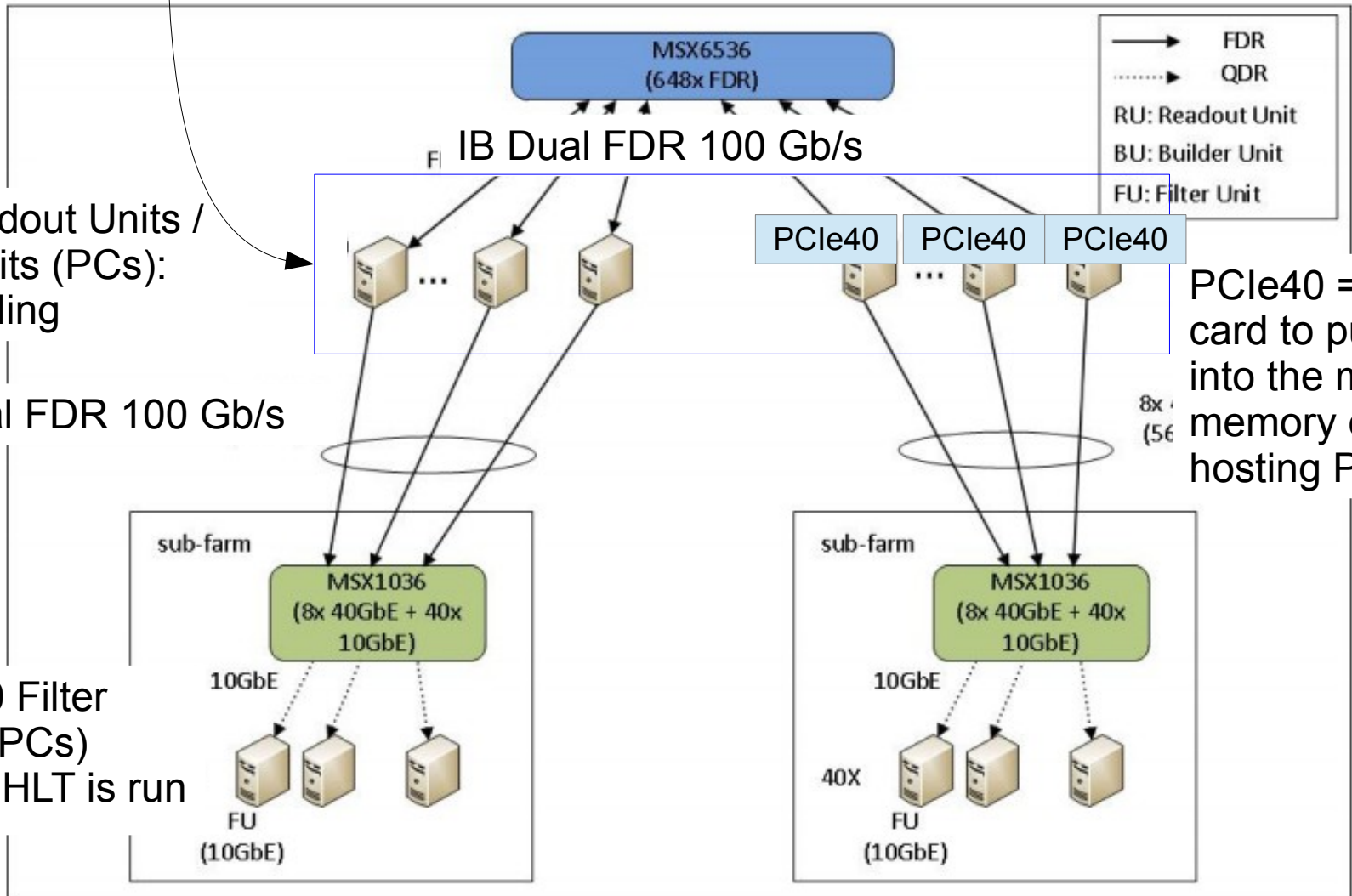
# New DAQ architecture

100 kB x 40 MHz = 32 Tbit/s  
to the DAQ via GBT links

~ 500 Readout Units /  
Builder Units (PCs):  
event building

IB Dual FDR 100 Gb/s

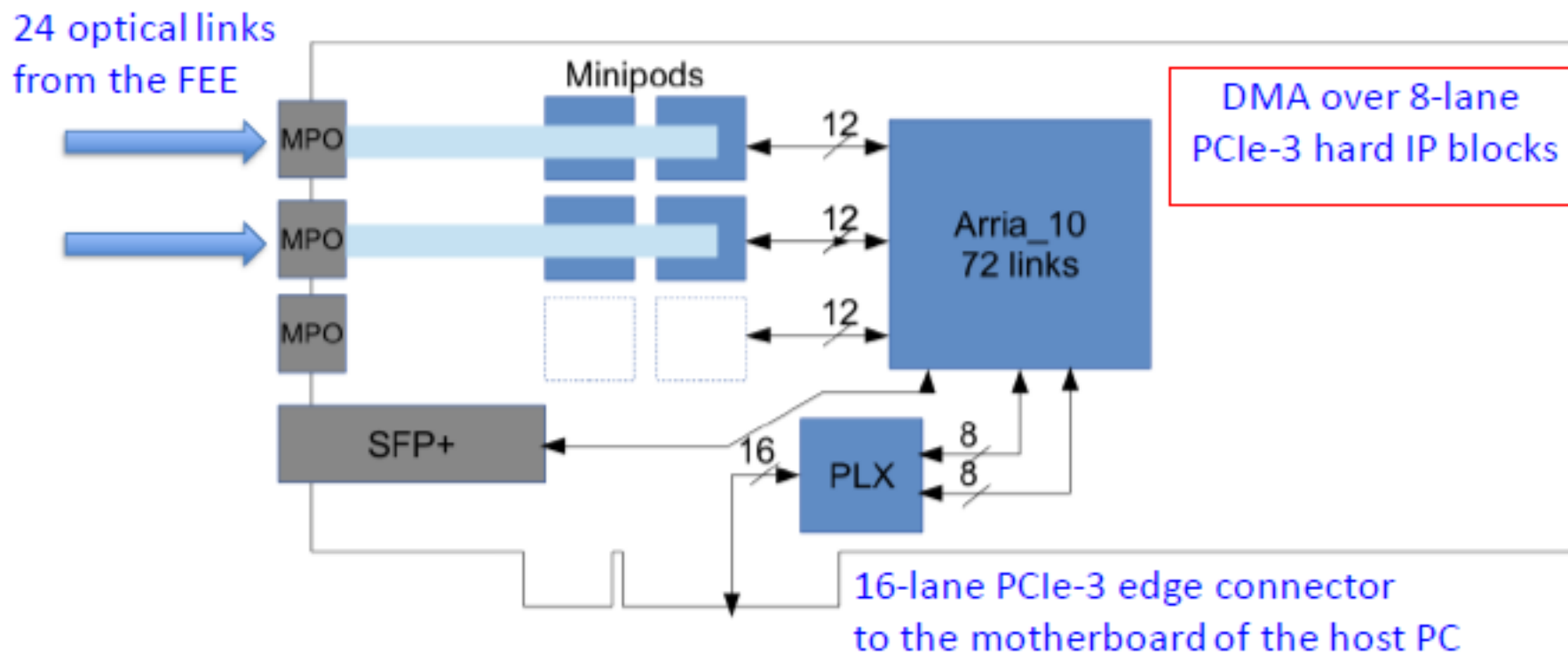
~ 1000 Filter  
Units (PCs)  
where HLT is run



PCIe40 = PCIe  
card to push data  
into the main  
memory of the  
hosting PC

# PCIe gen3 based Readout

- A main FPGA manages the input streams and transmits data to the event-builder PC by using DMA over PCIe Gen3.
- The readout version of the board uses two de-serializers.
- The same board can be used to clock and control distribution.

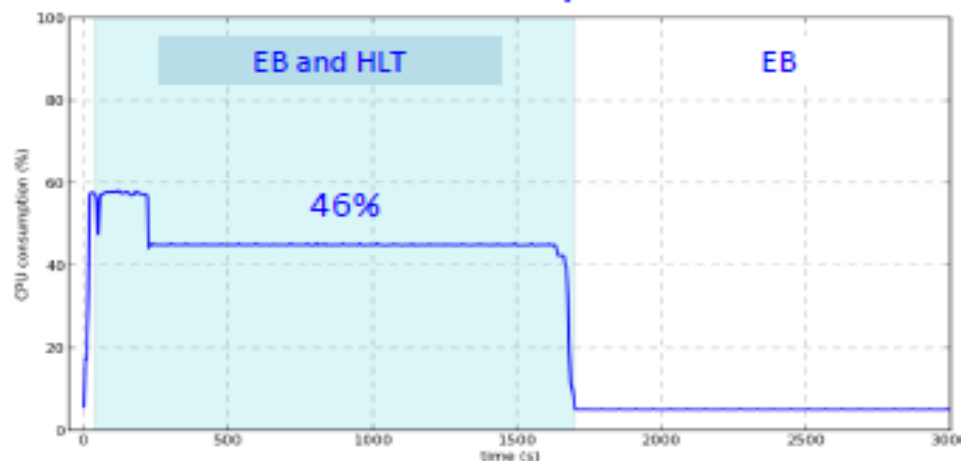


DMA Tests performed → OK  
on Altera Stratix V dev. board

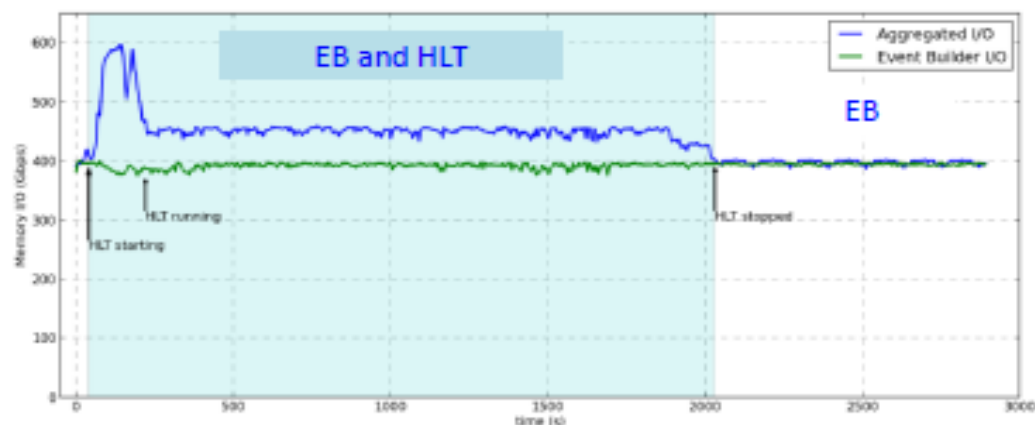
# EVB performance

At about 400 Gb/s more than 80% of the CPU resources are free

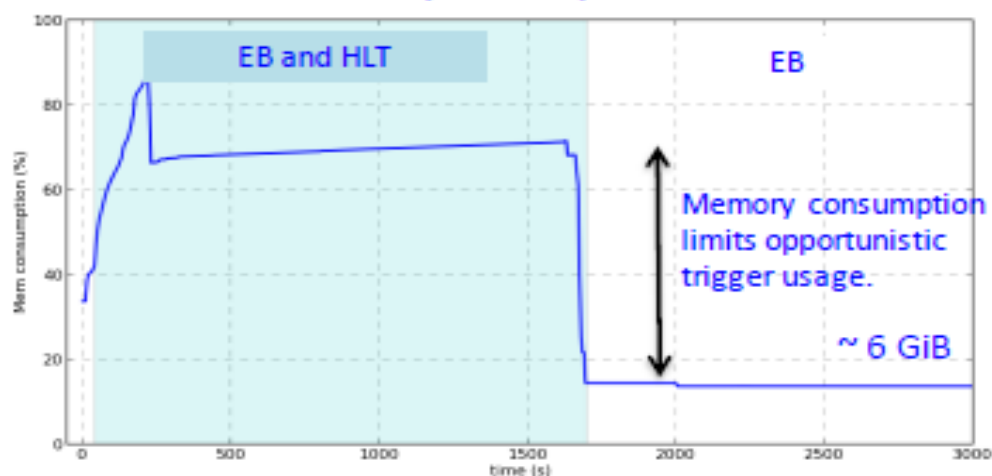
CPU consumption



Memory I/O bandwidth



Memory consumption

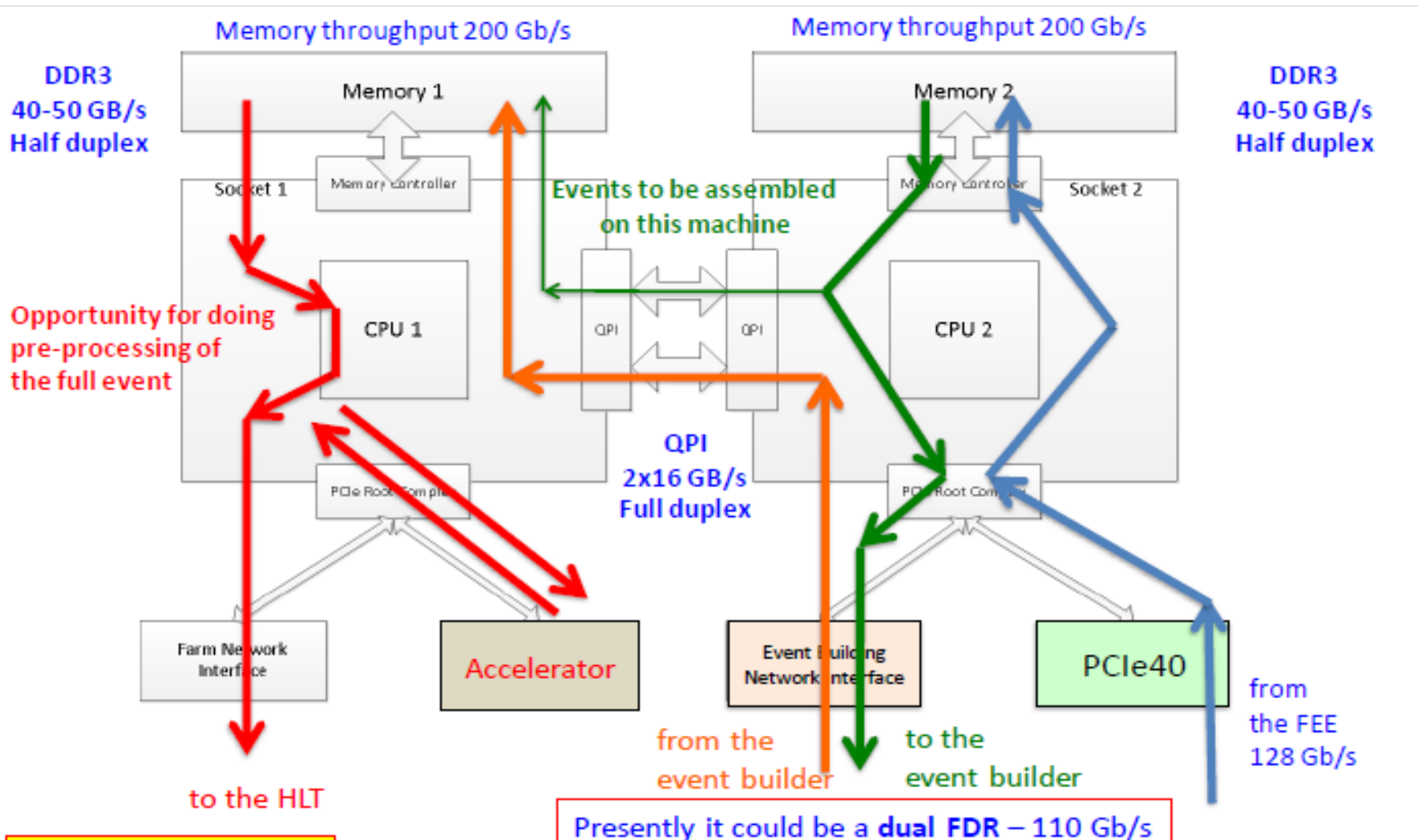


- PC sustains the **event building at 100 Gb/s today.**
- The Event Builder performs stably at **400 Gb/s**
- Aggregated CPU utilization of EB application and trigger 46%
- We currently observe **50% free resources for opportunistic triggering on EB nodes**: event builder execution requires about 6 logical core. Additional 18 instances of the HLT software running simultaneously.

The CPUs used in the test are Intel E5-2670 v2 with a C610 chipset. The servers are equipped with 1866 MHz DDR3 memory in optimal configuration. Hyper-threading has been enabled.



# Event Builder data fluxes: 400Gb/s



... it works!

→ PCIe40 based event builder is now baseline → TDR

# FastVelo on GPU (1)

- The goal of this work is to evaluate the performances of FastVelo on GPU wrt the original code (**optimized** for CPU):
  - ▶ Timing and tracking efficiencies (e.g. clone and ghost rates, efficiency for long tracks)
- Focus on the current VELO tracking running in HLT1

## Definitions:

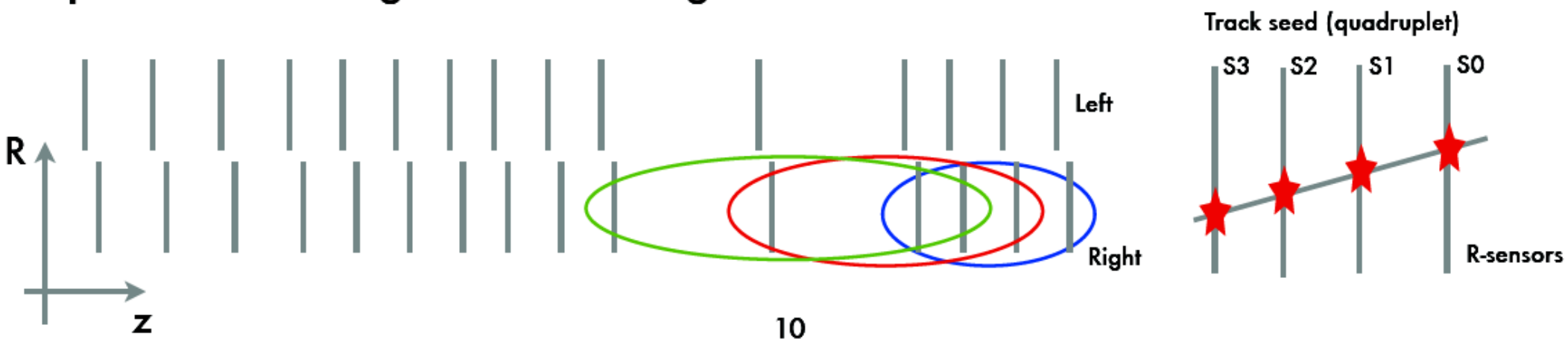
- $\text{efficiency} = \frac{N_{\text{reconstructed \& reconstructible particles \& no electrons}}}{N_{\text{reconstructible particles \& no electron}}}$
- ghost track = reconstructed track not matched to any true particle
- clone tracks = tracks associated to the same true particle
- long track = track reconstructed in VELO and in tracking stations ("T-stations")

# FastVelo on GPU (2)

- **Strategy:**
- Parallelize on the events (obvious...)
- Parallelize the algorithm:
  - ▶ Process each RZ track concurrently:
    - ▶ In the original algorithm hits already used in a track are marked and not further considered in the following iterations ("hit tagging")
    - ▶ ... but to avoid race-conditions, **hit tagging** must be **removed** in the GPU algorithm (clones and ghosts tracks diverge!)
- For the rest... try to keep the GPU version as closest as possible to the original one (code written in CUDA language)

# FastVelo on GPU (3)

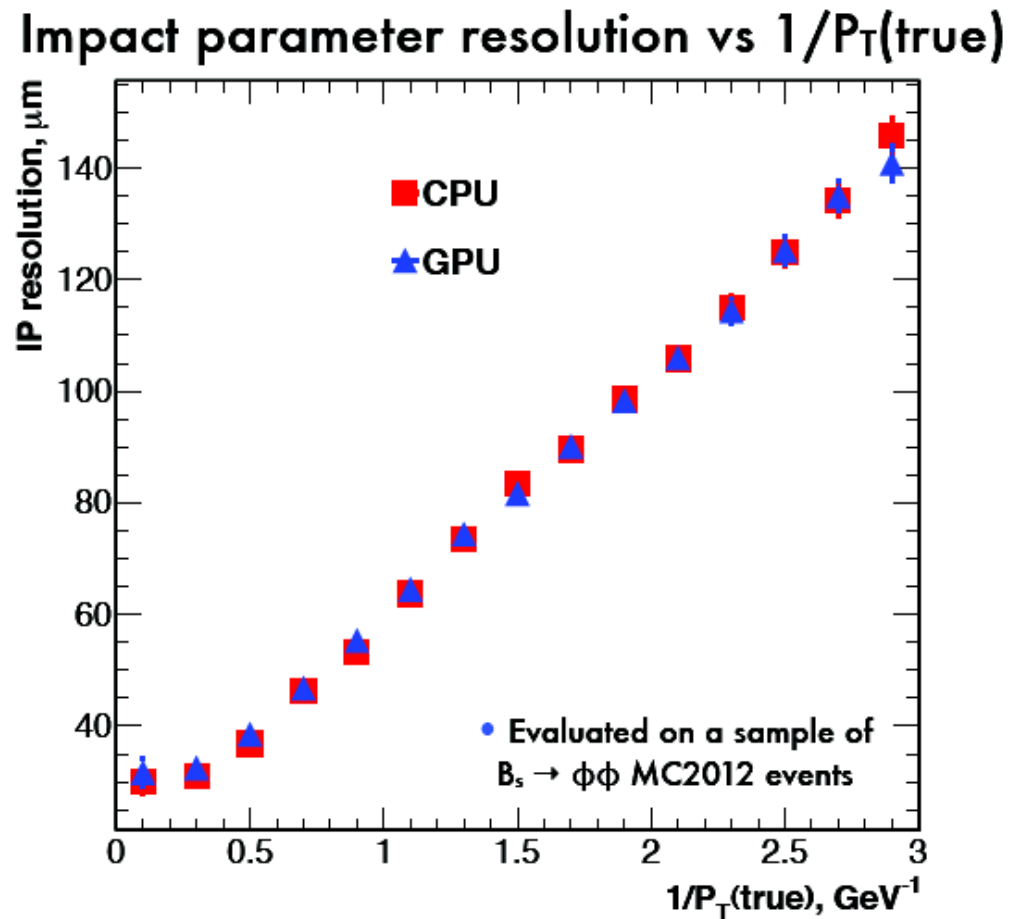
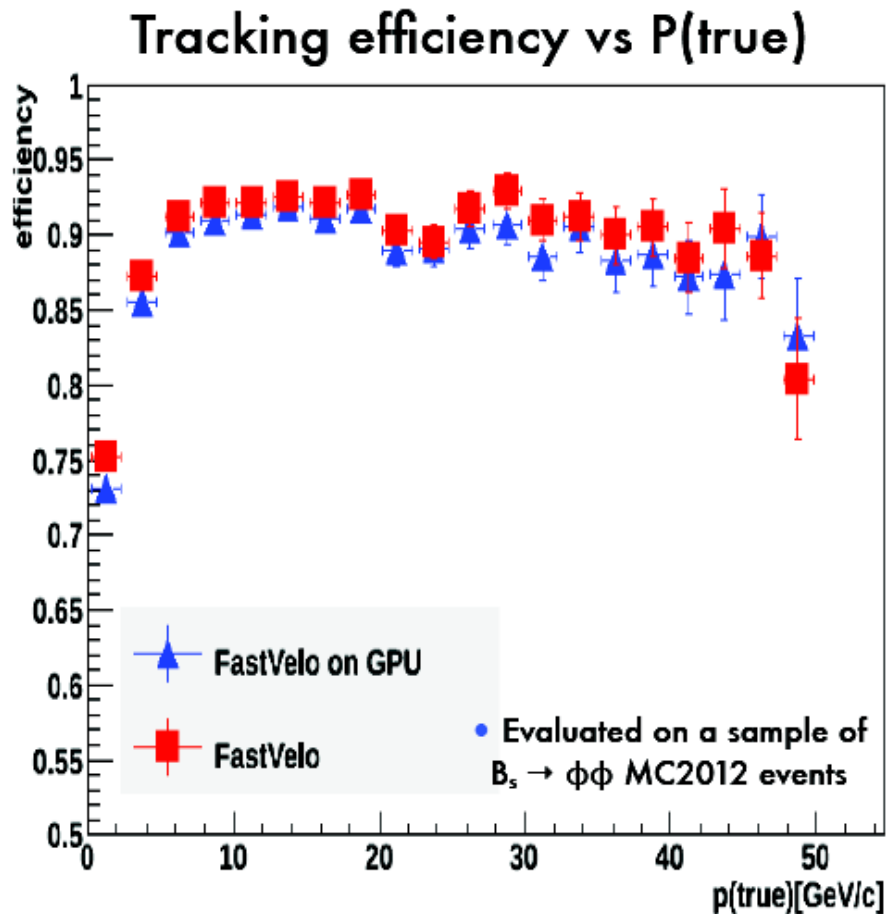
- **RZ tracking:**
- Only R-sensors are used
- The algorithm looks for quadruplets of hits in four contiguous R-sensors (seed) on both halves.
  - Each thread works on a set of four contiguous R-sensors and find all quadruplets.
- Then each quadruplet is extended in parallel as much as possible adding the remaining R-sensors



# FastVelo on GPU (5)

- **Space tracking:**
- Add hits on  $\phi$ -sensors
- Each RZ track is processed concurrently by assigning a space-tracking algorithm to each thread:
  - Search for a triplet of hits: for each hit in the first two  $\phi$ -sensors, the candidate hit in the third sensor is the one most compatible with predicted position (best  $\chi^2$ )
  - The track is extended and its parameters are found by minimizing  $\sum_{\text{points}} \chi^2$  (linear system solved by substitution)
  - This part is almost a re-writing in CUDA language of the original space-tracking code

# Results (2)



Tracking performances close to the optimized CPU code

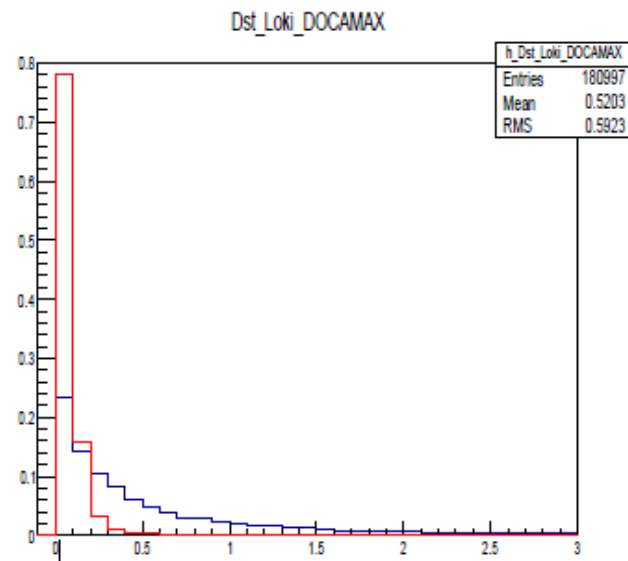
# Inclusive $D^*$ trigger

## Signal vs background variables distributions

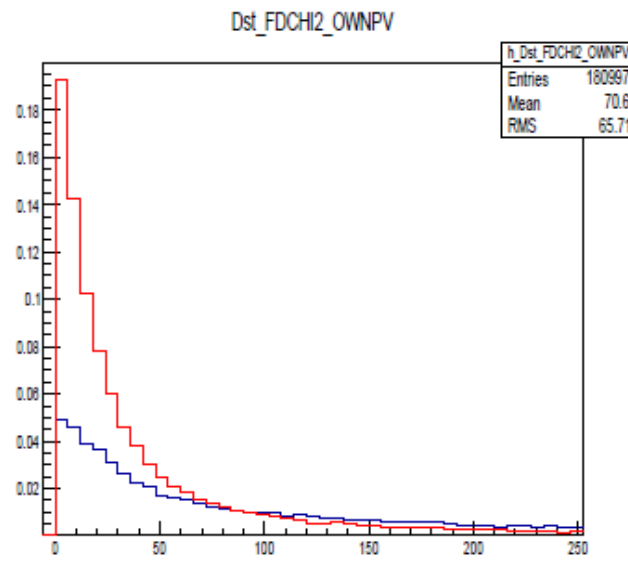
example:  $D^0 \rightarrow K K \pi \pi$

Blue background, red signal

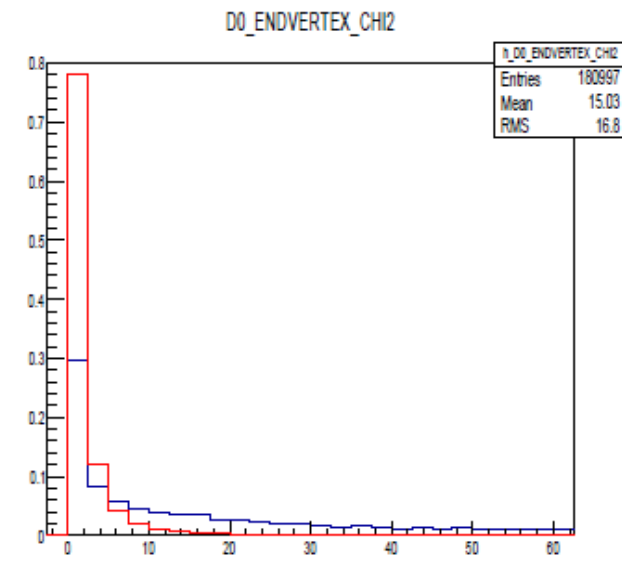
Chosen input variables



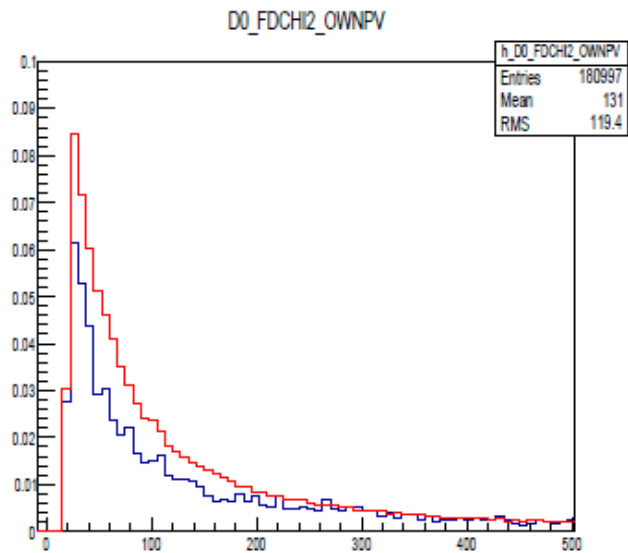
(a)  $D^*$  DOCA



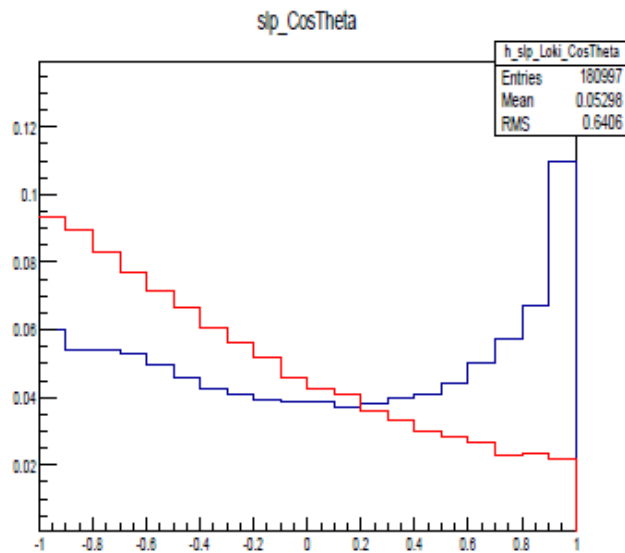
(b)  $D^*$  flight distance



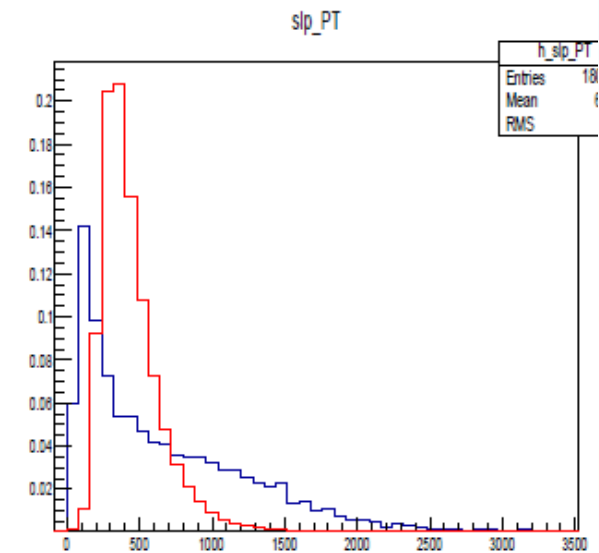
(c)  $D^0$  vertex  $\chi^2$



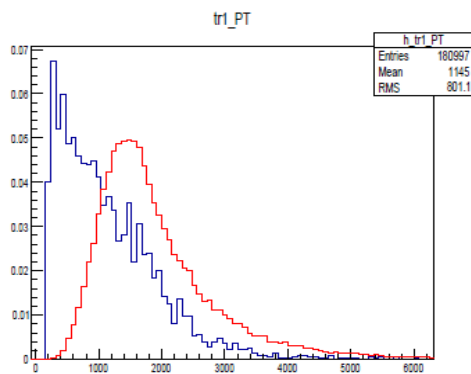
(d)  $D^0$  flight distance



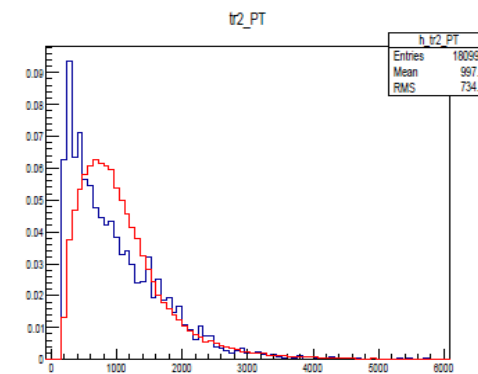
(e)  $\pi_{sl} \cos \vartheta$



(f)  $\pi_{sl} P_T$



(g)  $tr1 P_T$



(h)  $tr2 P_T$

The input variable is the sum of the two tracks  $P_T$