First experiences with a parallel architecture testbed in the LHCb trigger system

Stefano Gallorini

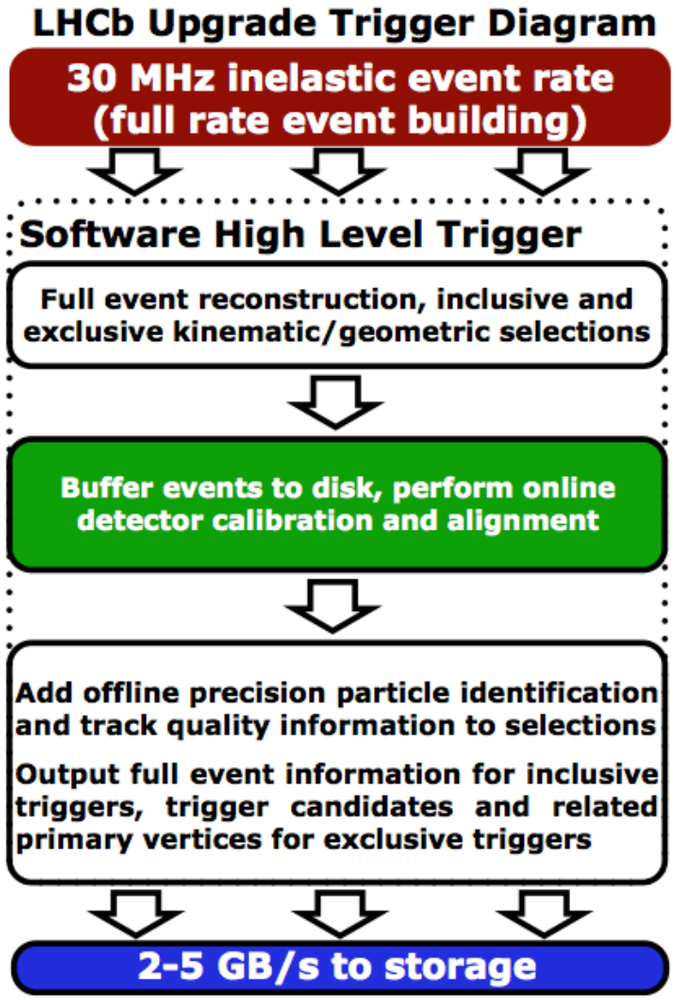
On behalf of the LHCb collaboration

CHEP 2016 - San Francisco (Ca) - 13/10/2016

Introduction

- To match theoretical precision on the CPV observables, LHCb will be upgraded during 2018 to run at higher luminosity ($2 \cdot 10^{33}$ $cm^{-2}s^{-1}$)
- **Problem:** current L0 hardware trigger limits the output rate to 1 MHz:
  - Hadronic trigger efficiencies saturate at higher luminosity!
- **Solution:** detector readout at 40MHz collision rate, full software trigger!

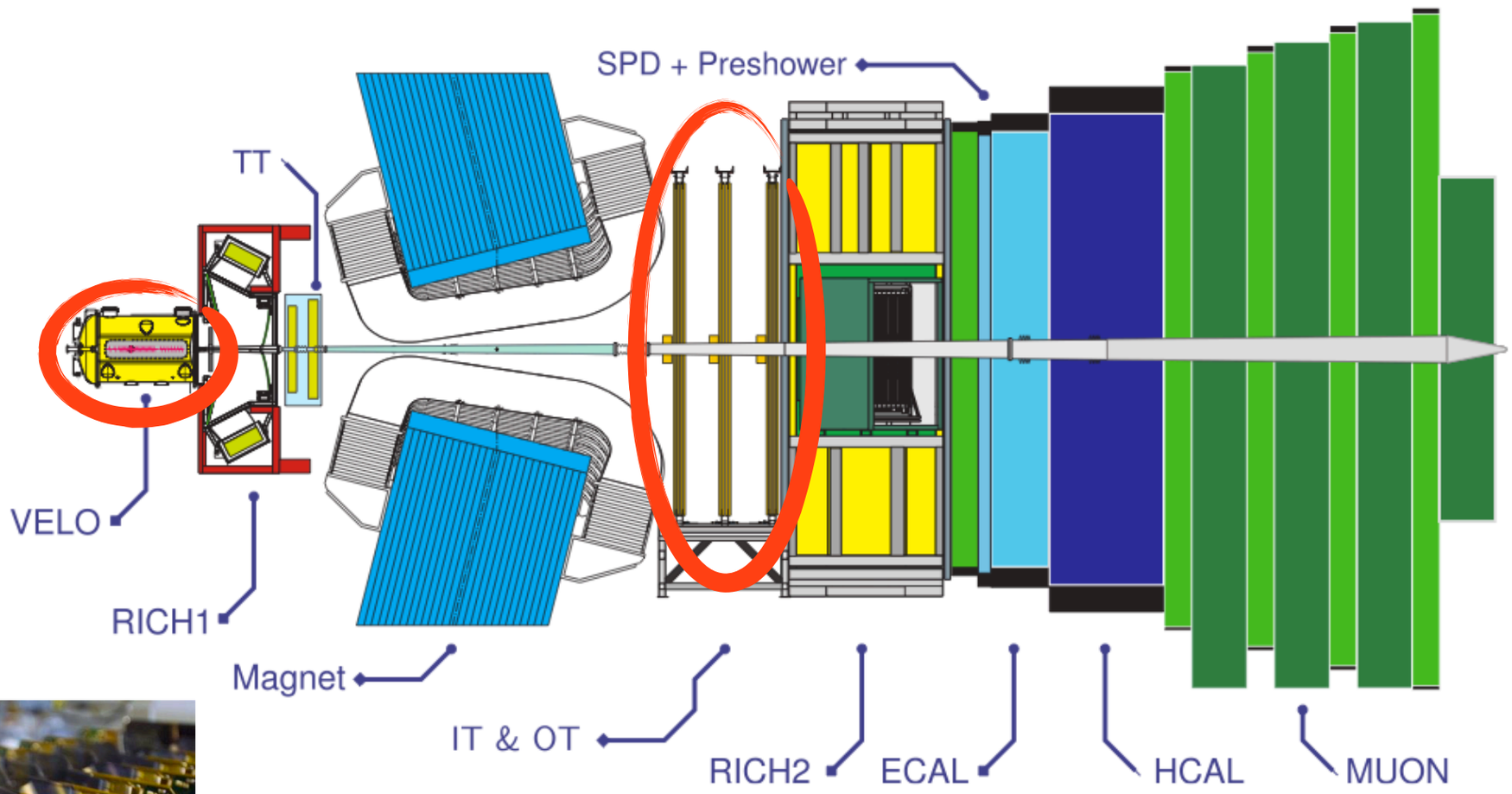
- The trigger and tracking systems need to be revised to face high luminosity conditions
- Tracking at 30 MHz!*
- Full event reconstruction online!

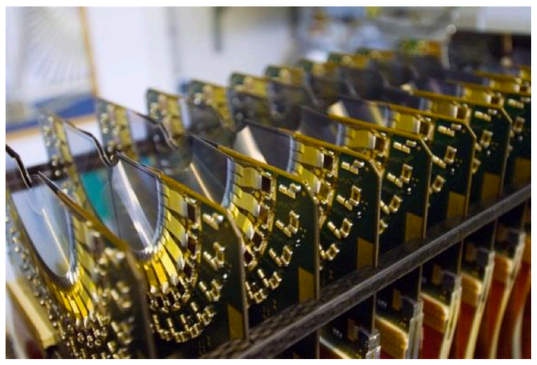


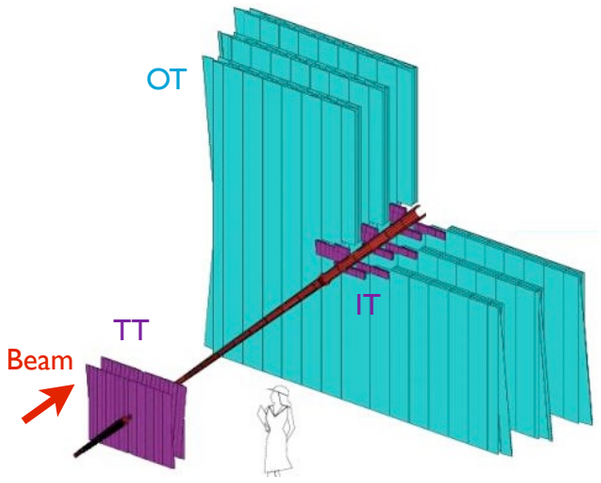
* Rate of visible interactions

2

The LHCb tracking system







- **VELO**
  - Surrounds the interaction point
  - Made by 21 silicon micro-strip stations with r-ϕ geometry
  - 2 retractable detector halves (8mm from beam when closed)
- **T-stations**
  - Consist of Inner Tracker (IT) and Outer Tracker (OT)
  - **IT:** planes of silicon micro-strip sensors ($\sigma(\text{hit})_{x,u,v} \approx 50\mu m$)
  - **OT:** planes of straw tubes ($\sigma(\text{hit})_{x,u,v} \approx 200\mu m$)

3

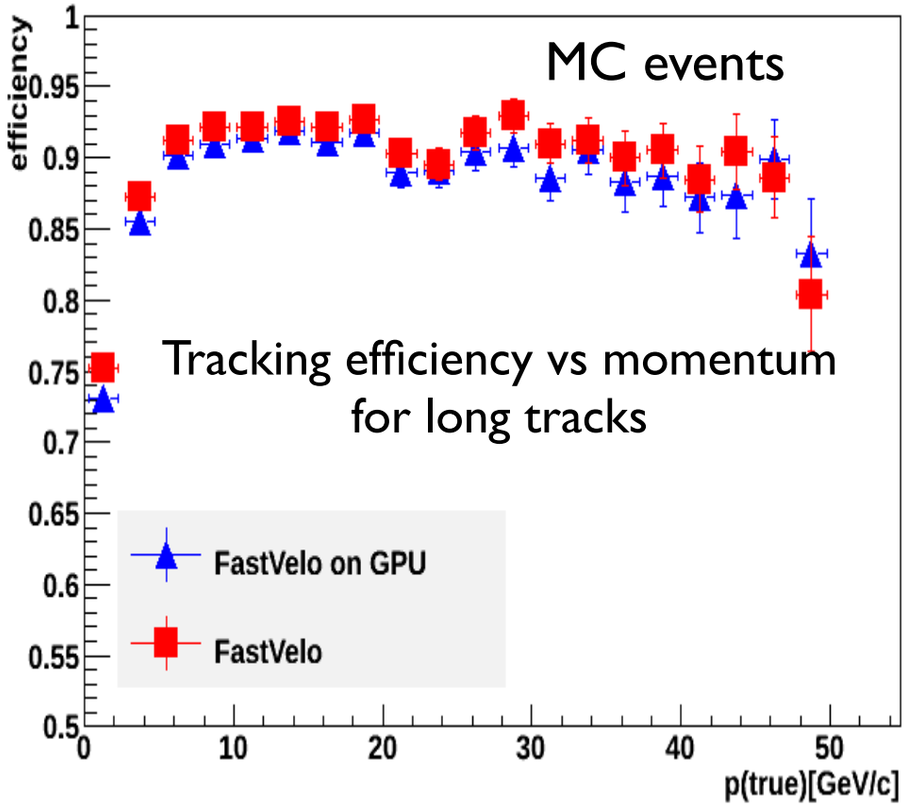
Many-cores in LHCb

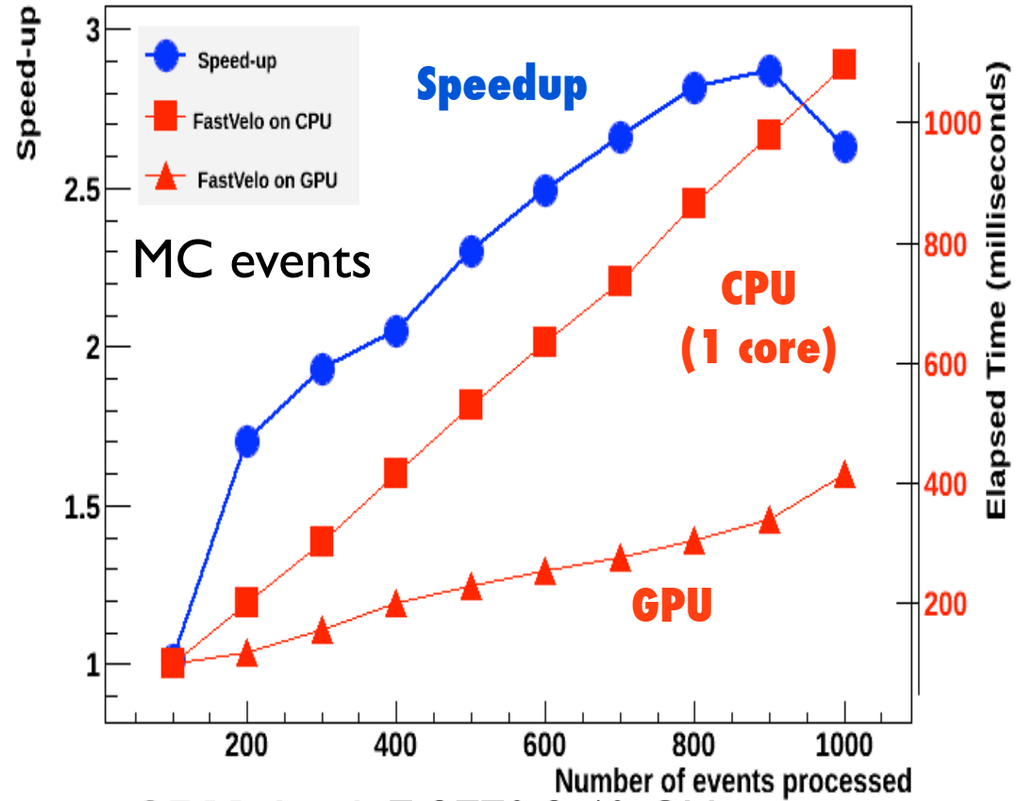
- GPGPUs can be used in the upgraded HLT to speed-up trigger selections, reducing the cost and increasing the computing power of the HLT farm
  - Due to a small event size (O(100 kB)) a huge speed-up could be obtained by processing **many events in parallel**
  - Tracking algorithms are usually well suited for parallelization
- Run II offers an unique opportunity to test new hardware and new algorithms in a realistic environment!
- Many questions to answer:
  1. What issues will we face inserting new hardware in a CPU oriented environment?
  2. How can we fully exploit the computing power of this hardware?
  3. How shall we move data to and from this hardware?
  4. How can we alleviate offloading latencies?

4

VELO tracking on GPU

- "FastVelo" is the algorithm for the tracking of the VErtex LOcator (VELO)
- It ran online in the HLT farm:
  - Written to be fast and highly efficient to cope with high rate and hit occupancy. Several conditions and checks introduced throughout the code to speed up the execution.
- FastVelo has been rewritten in CUDA and partially modified to run concurrently on GPU*





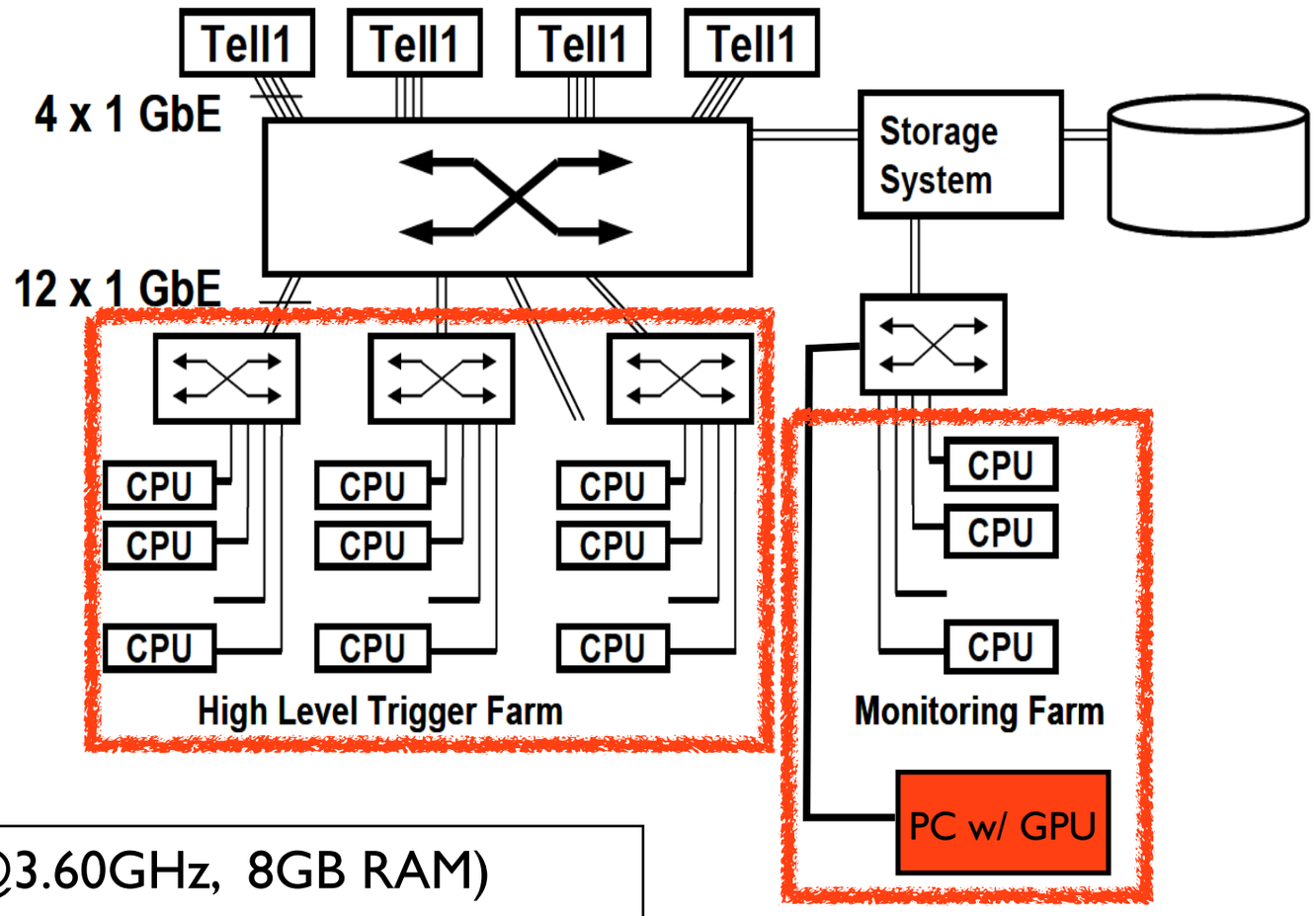
CPU: Intel i7-3770 3.40 GHz
GPU: NVidia GTX Titan (14 SMX, 6GB)

* More details on back-up slides

5

Parasitic test-bed

- At the end of last year a node equipped with a GPU has been inserted in the LHCb online monitoring system
  - No interference with data taking!
- During data taking, a subset of events was sent to the node and processed in parallel by GPU and CPU-based VELO tracking to allow a direct comparison
- Rate of incoming events: O(10Hz)



- **Host:** desktop class PC (i7 - 4790, CPU@3.60GHz, 8GB RAM)
- **Device:** NVidia GTX Titan X (3072 cores, 12 GB ram, 250/300W)

6

Integration with framework

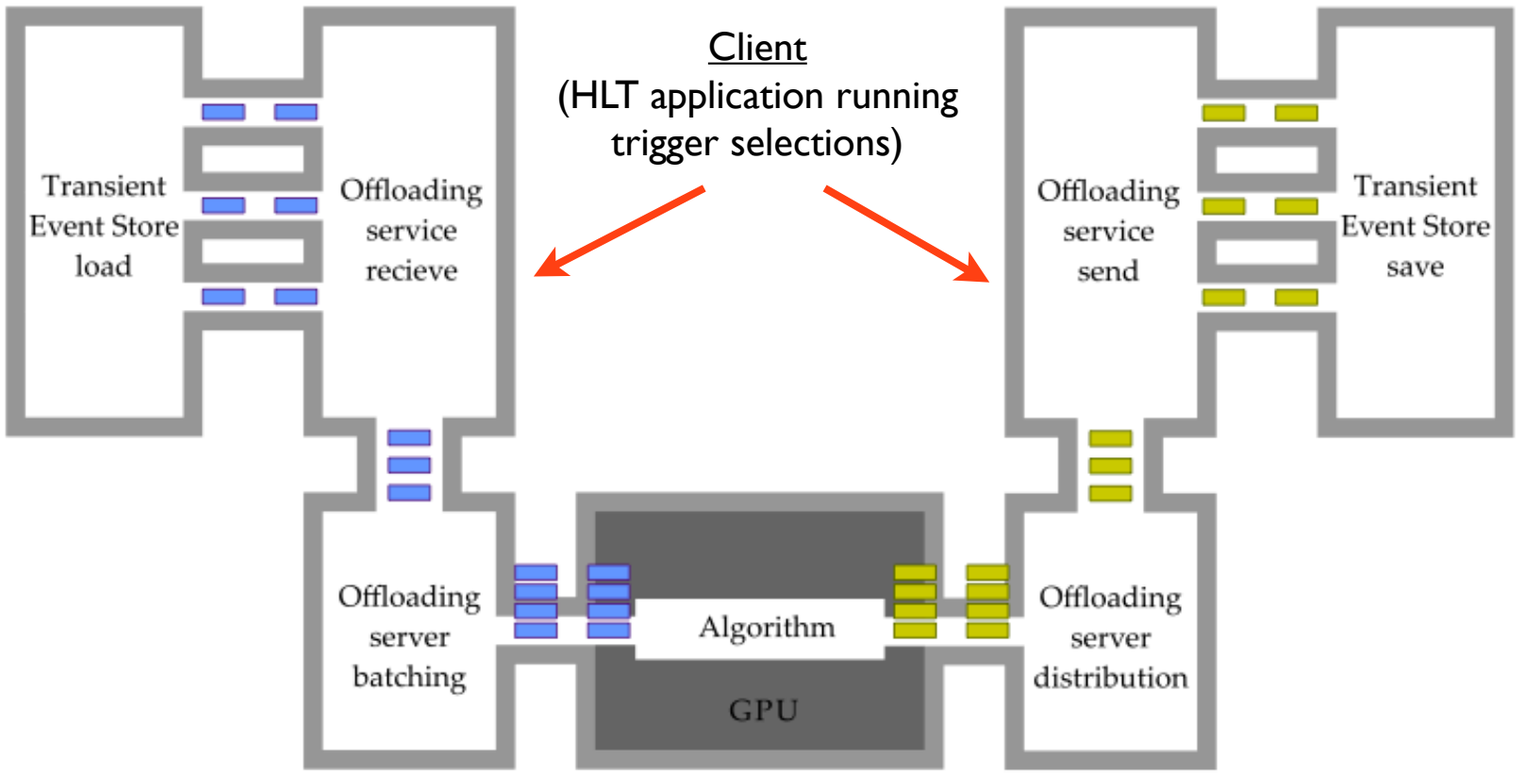
- A critical issue for the use of accelerators is the integration with the software framework of LHCb ("Gaudi")

- Gaudi schedules one event at time so a client-server approach has been adopted ("Co-processor manager")

- Data are serialized by each client (e.g. an HLT node) and submitted to the server which runs one or more algorithms on the device

- The server checks continuously its queue

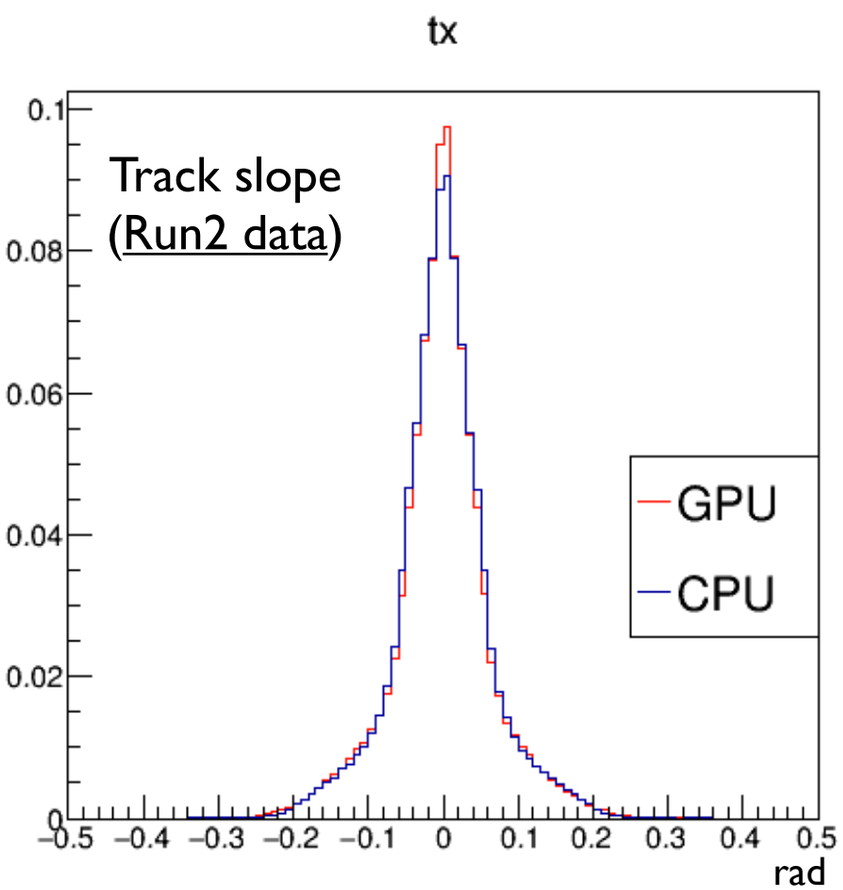
- Two running modes:
  1. Wait for N events before submitting ("batch")
  2. Send data as soon as the queue is not empty

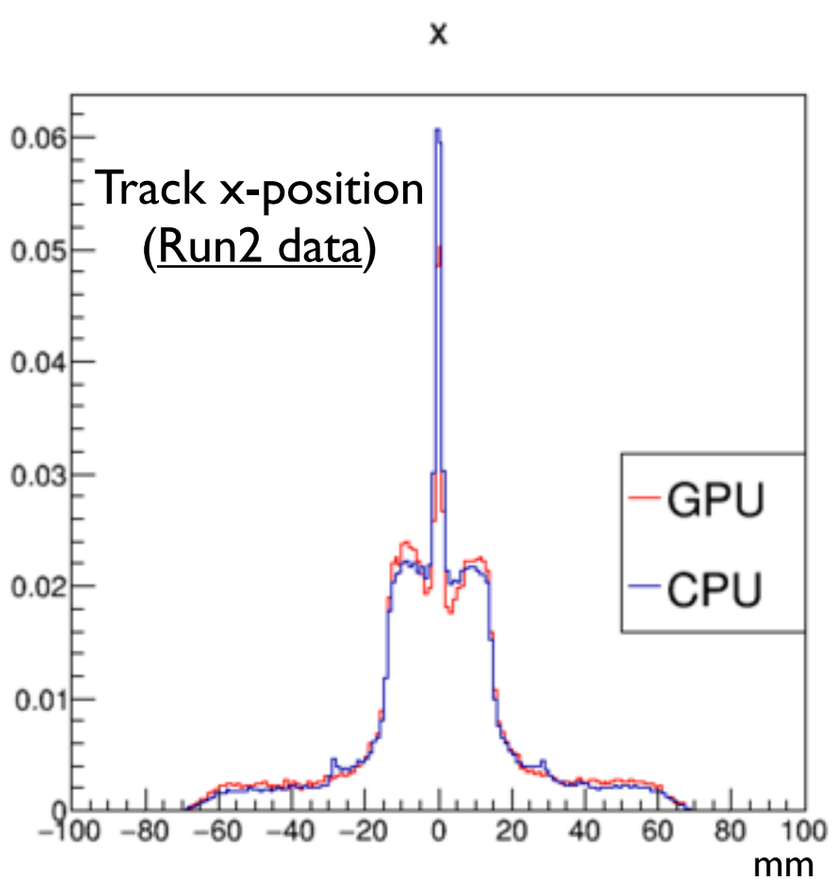


7

Results (I)

- During the parasitic test, client and server were in the same machine (only one node available for the test)
- Data size higher than the one used during the development of the algorithm ➜ need to cure size overflow of data structures
- Track parameters quite in agreement w/ official reconstruction

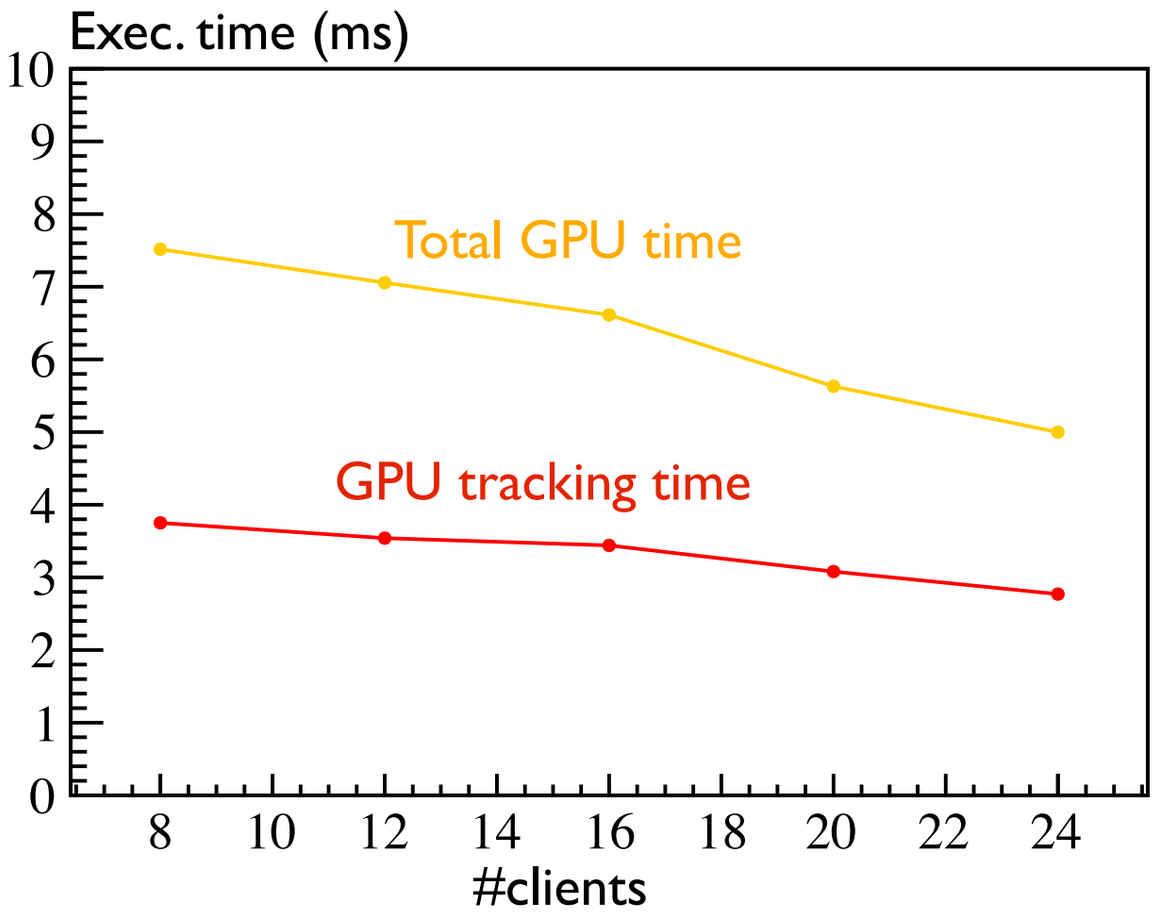


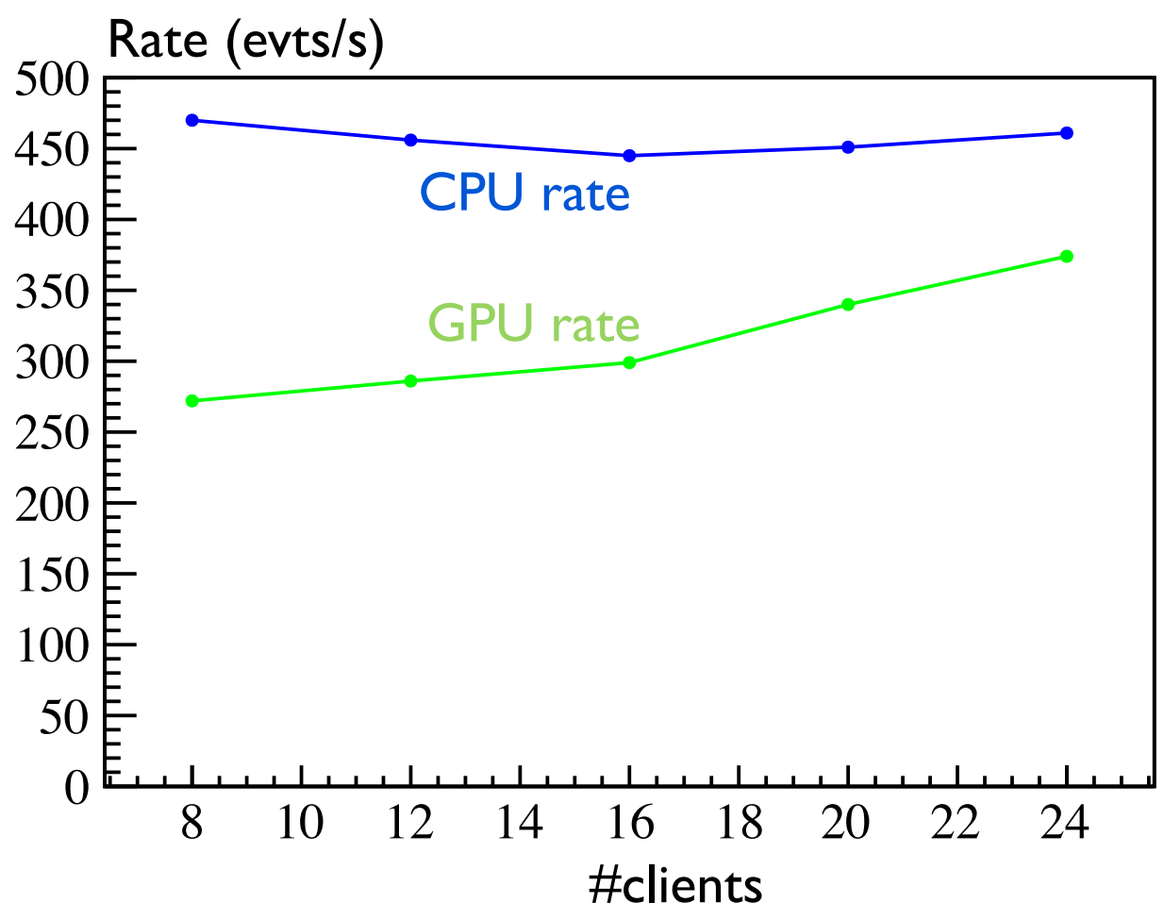


8

Results (II)

- The rate of processed events grows with the #clients as expected
  - But the number of events in flight limited by the #logical cores of PC
- More clients* (nodes) would be needed to get better performances and stress the system in a more realistic scenario





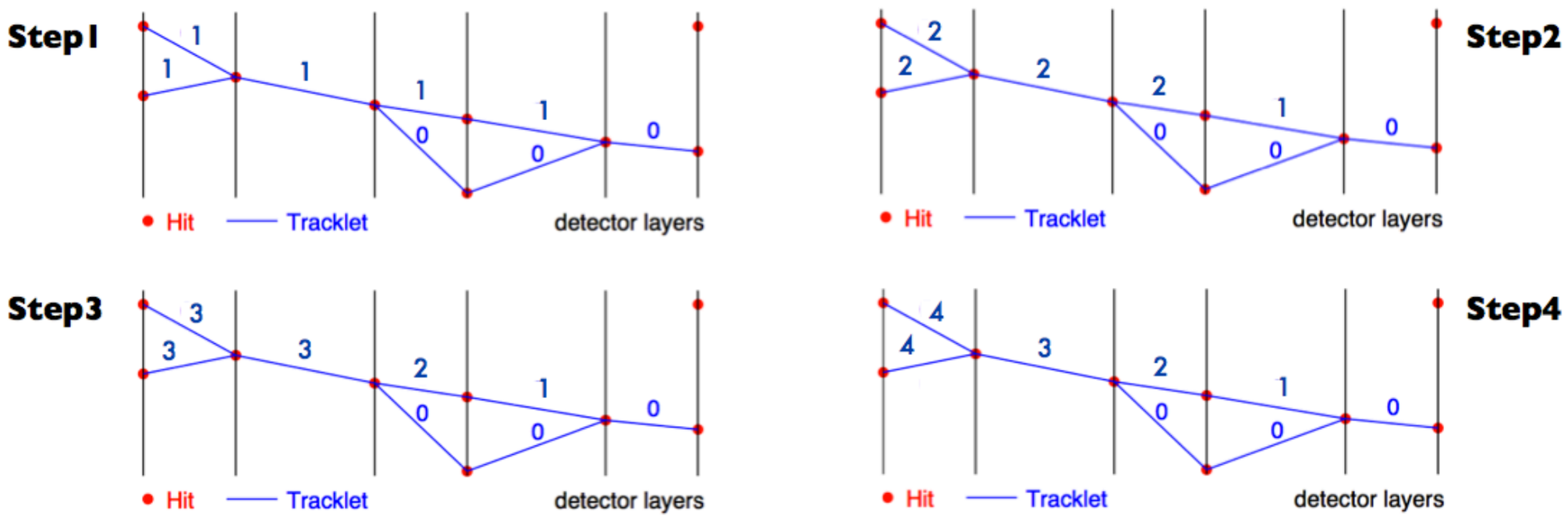
* Connected with server e.g. via TCP-IP

9

Future work

- In addition to FastVelo, a new parallel algorithm for the tracking of the T-stations, inspired to Cellular Automata, is under developing on GPU
- Work done for the OT detector (to be extended to include IT)
- The algorithm proceeds in three steps:

1. **Tracklets generation**: each tracklet has a counter, initially set to 0. Tracklets are the "cells" of the automata and counters their states
2. **Neighbour finding:** connect adjacent tracklets according to track model
3. **Evolution and track formation:** all cells evolve in **parallel** in time steps



10

Conclusions

- First successful integration of a GPU gaming card in the LHCb online system during Run2

- The throughput of the GPU was limited by the number of clients available in the test. More nodes are needed to fully exploit the GPU computing power

- On the other hand, the client-server approach could introduce too latency if the number of clients increases

- A viable solution would come from a concurrent version of Gaudi ("GaudiHive") which will allow to run on multiple events
  [see Concezio's talk: The LHCb software and computing upgrade for Run3]

- A tracking algorithm based on cellular automata is under study for GPU. It will be tested in our testbed during Run2

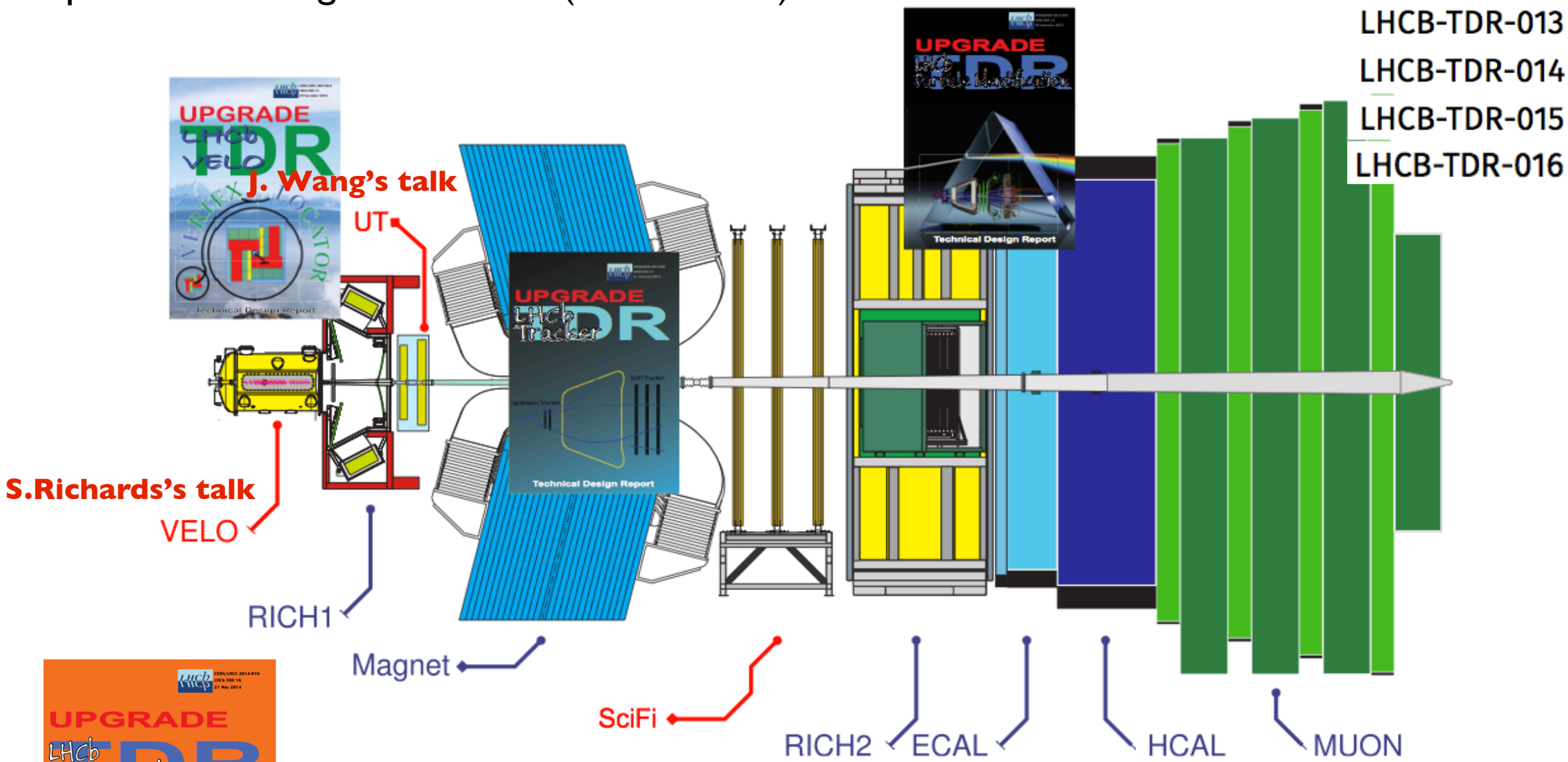
11

Back-up slides

12

The LHCb Upgrade

The upgrade of the detectors will take place after Long Shutdown 2 (2018 - 2019)



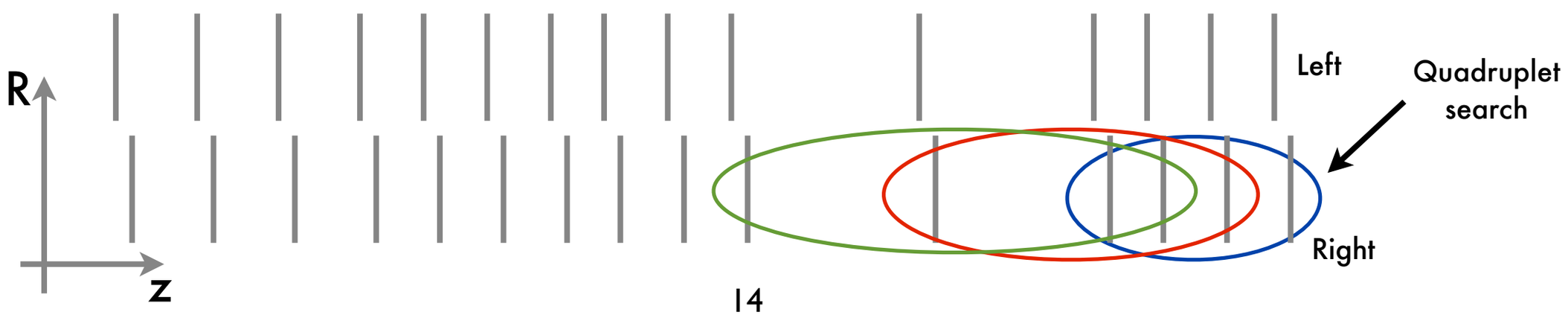
13

VELO tracking

- Tracks inside the VELO are straight lines (magnetic field negligible)
- The VELO track reconstruction is done in two steps:

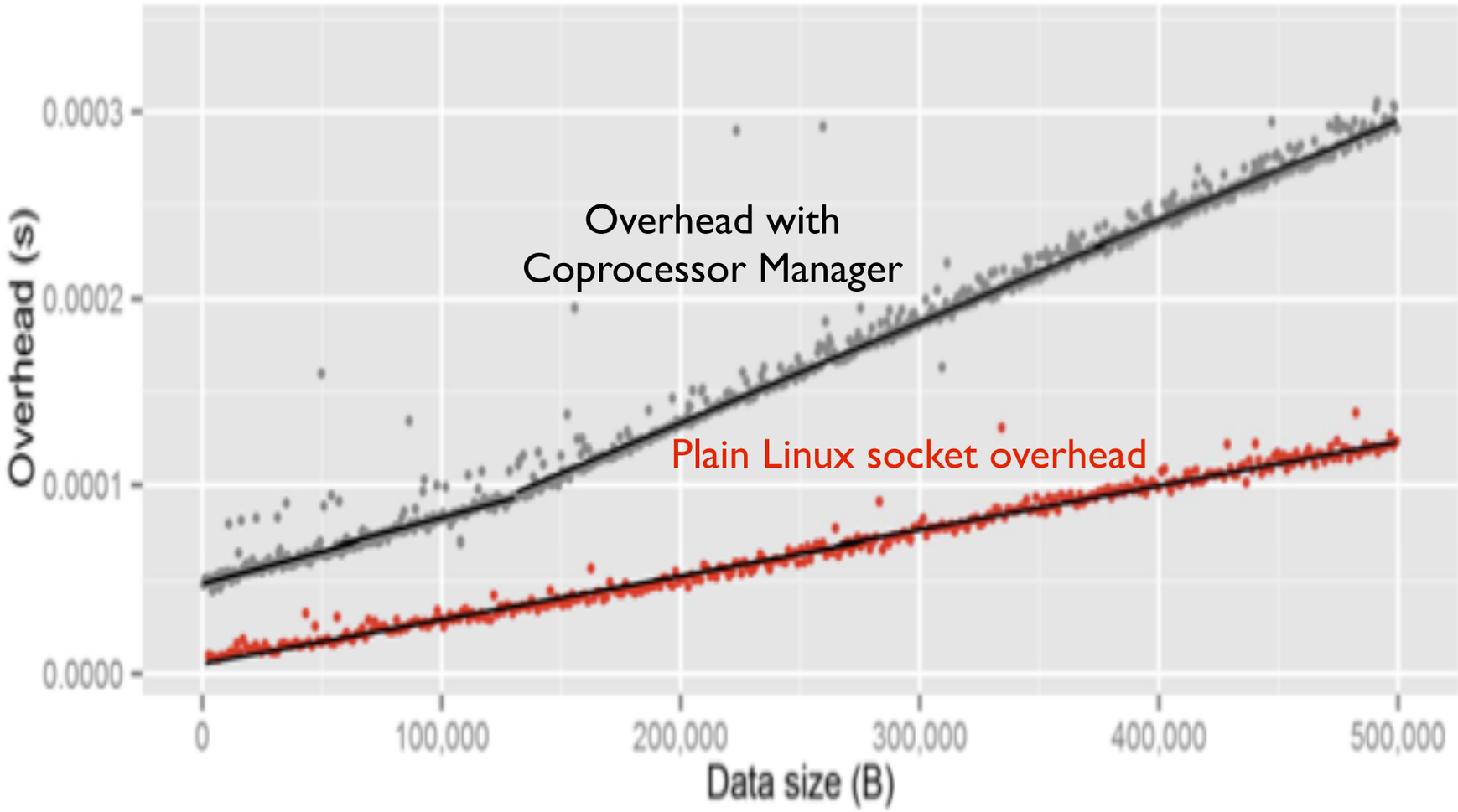
1. **R-Z tracking:**
   - It looks for quadruplets of hits in four contiguous R-sensors on both halves ("seeds")
   - Quadruplets are extended by adding hits in the remaining sensors ("R-Z tracks")
   - The process is repeated for triplets of R-hits

2. **Space tracking:**
   - The R-Z tracks are promoted to 3D-tracks by adding hits in ϕ-sensors
   - Fit tracks using a χ^2 minimization



Data transfer overhead

- Data transfer overhead added by Co-processor manager communicating via two Linux local sockets



15