Parallel (embedded) architecture implementations for U2

Giovanni Punzi
Università di Pisa & INFN

LHCb Upgrade-II tracking workshop

7/03/2024

From Matteo's slides: RTA scenario for U2

Scoping scenarios: Real Time Analysis

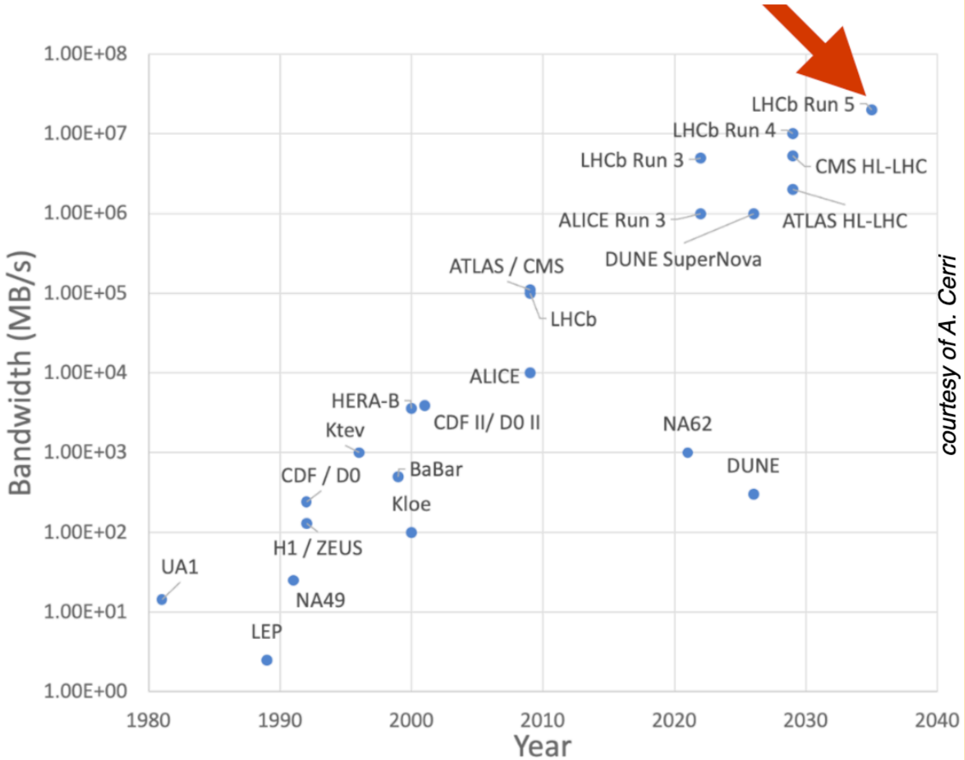
Largest throughout in HEP

Evolutionary path

Scale up the Run 3 processing model → full detector reconstruction on GPUs

Revolutionary path

Finding sub-detector primitives, for example tracks or calorimeter clusters, on FPGAs



Ensure critical mass of skills across all architectures. Decide the architecture mix for Run 5 at the right moment to maximize both technology and long-term support

Why even consider a different path ?

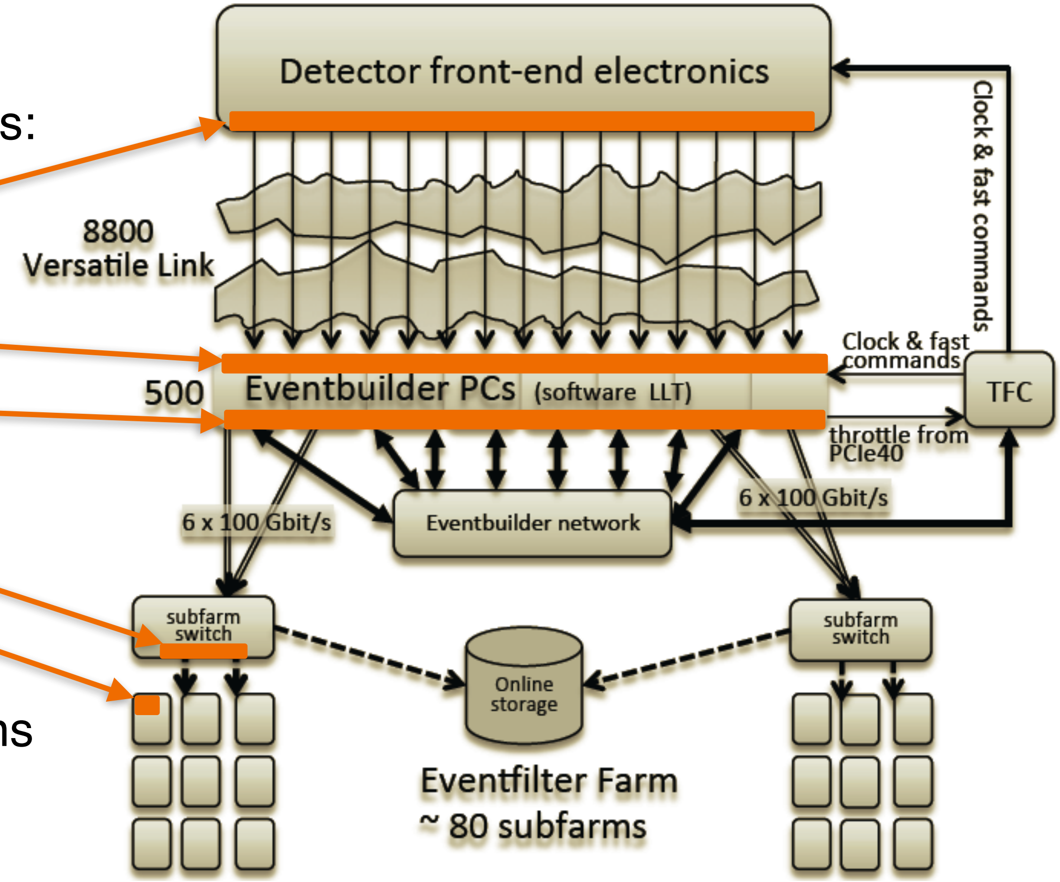
- Why is it worth considering to enhance the current reco system with other means?
  - Because Moore's law is slowing down, and we can't afford to do the same.
  - Lack of progress in reconstruction may render useless detector progress
  - Biggest problem is evolution of memory speed -> affects pattern recognition
- The idea of a further level of pre-processing is not actually that revolutionary
- Actual novelty here is introducing transparent "embedded reconstruction"
  - Next slide discusses this
  - First ideas in this direction: see GP talks at TTFU and 1st FCC workshop for discussion of Real-Time analysis, embedded reconstruction, and use of timing.

Different stages where track reconstruction can be performed

- Several (non exclusive) possibilities:

  1. On the Front-End
  2. Pre-Build
  3. Post-Build
  4. Within the EFF
  5. Within CPUs.

- At each stage, there may be options to reduce event size OR rate



Stages 1 and 2 not much exploited yet - most potential is there

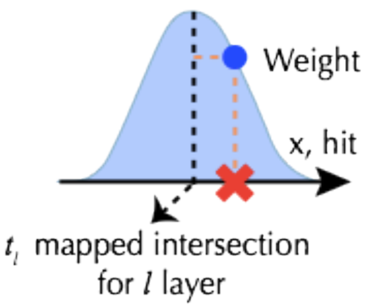
Front-End processing

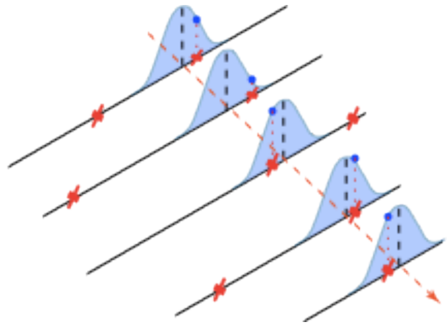
- Processing 'on the detector' entails several limitations:
  - Space, locality, accessibility, radiation...
  - The main point of full event readout was to break free from limitations
- Nevertheless, technology progress allows some data (pre)processing in the FE
- ***Cluster finding***, for instance, could certainly be helpful (see later)
- Basically it only makes sense if can reduce data flow and save $$
- Best known example in CMS Phase-II, with doublet matching as a way to filter high-pt tracks from loopers, hugely reducing the data flow into the tracking trigger
  - Possibility that has been considered for LHCB as well (es MT, Muons, MC)
- Timing info processing also a possibility in principle, but gain to be proved [see Vava's "fine print" at the U2-VELO workshop]
- Technologies: ASICs or FPGAs
- Expected to be of some use, but unlikely to be a U2 game changer

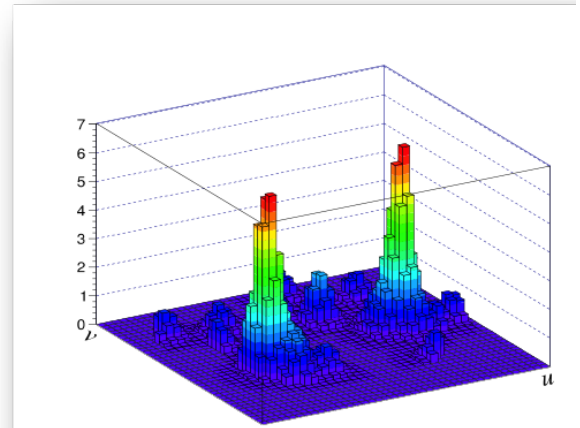
Processing at the Pre-Build level (near to readout)

- Some practical advantages:
  - Easier to upgrade/stage/modularize than in the FE
  - Much easier to establish at least local connections within tracking sections
  - Avoid need for "unpacking" the event to access the portions of interest
  - Can use just the amount of electronics needed for the job and no more
- Possibility of data reduction/compression, reducing EB load.
- Most attractive application is pre-tracking reconstruction, i.e. ***primitives***
- The concept of reconstruction primitive can apply to every detector, but for tracking this means mainly pattern-recognized track stubs, to be used as seeds for accelerated track reco in HLT
- Embedding primitive reconstruction **in the readout** allows to deal with them as **raw data**
  - Primitive reconstruction transparent to EB and HLT - as if produced by the detector
  - Part of the raw event, persisted to both HLT1 and HLT2, that will refine the reconstruction.
- ***Requires*** capability to handle the full collision rate
- FPGA most suitable technology: combination of power, flexibility, bandwidth (and low-power)
- Can exploit the power of FPGA in the readout boards.
  - Needs them anyway to read data at high speed.
  - Modern FPGA devices carry large processing power per unit bandwidth: put it to good use !
- Lots of development work in RTA WP6 [see Peilian talk, and my talk at 106th LHCb week].
  -> Today will focus on prospects from 'artificial retina' technique, that is in DAQ-TDR for LS3E.

Architectural organization: the "artificial retina" approach



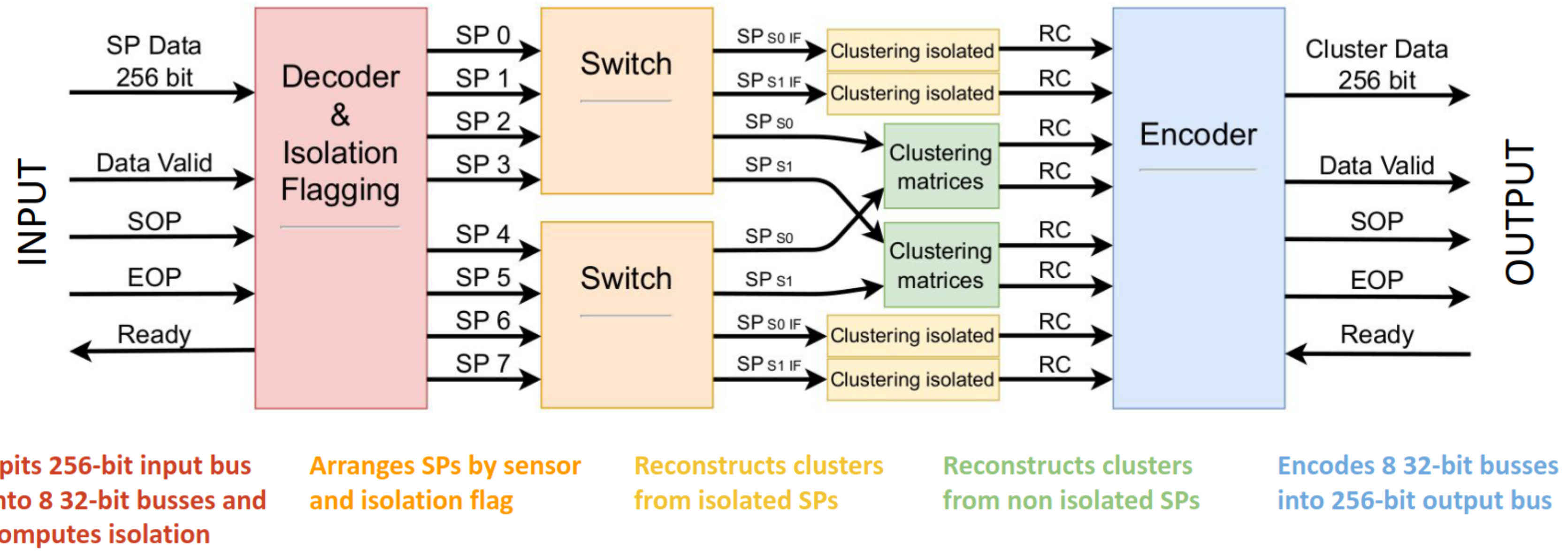




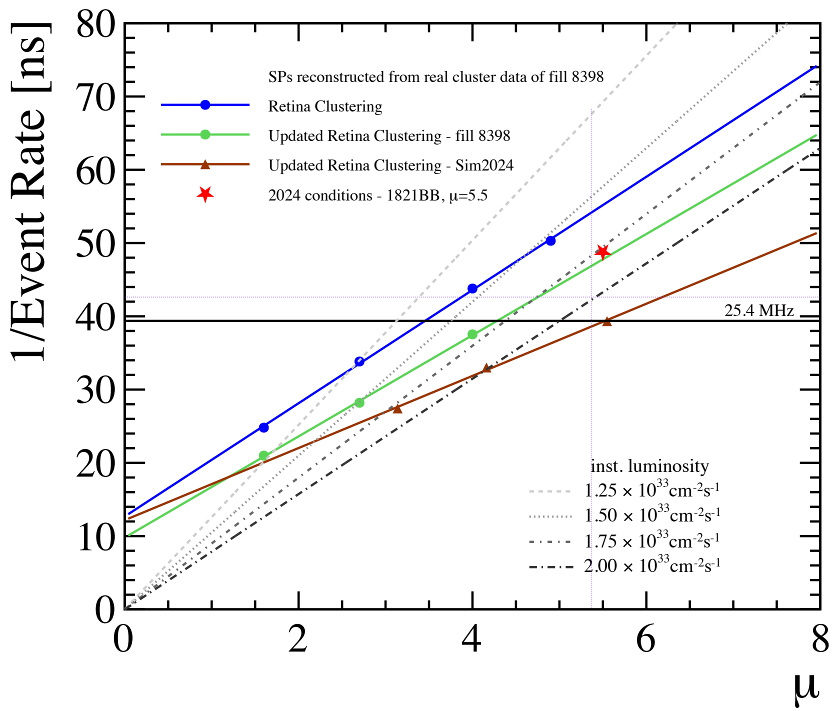
- RETINA is a *logic architecture* for very fast reconstruction, addressing real-time reconstruction by ***extreme parallelism*** and ***high connectivity,*** imitating features of natural neural systems.
  - **NO RAM = no memory access.**
    - "Future-proof", as memory improves more slowly than processing speed.
  - Perform a computation similar to a Hough-transform. Strong pattern recognition capability (but not particularly strong on precision parameter determination.
  - Ultra-low latency, allowing reconstruction to happen transparently "on the fly" while data is being readout, with no buffering overhead. Ideal for Pre-Build processing.
  - Allows to exploit every available FPGA gate, leading to ~maximum theoretically achievable computational efficiency
- Can be applied to essentially any tracking (also beyond tracking).
- While other architectures may be viable, this is one for which we have strong proof-of-principle

Existing Run3 application: VELO hit finding

- VELO hits produce clusters of pixels
- Cluster finding, initially intended to be performed in HLT1, has been embedded in the firmware, running in VELO readout boards (TELL40)
- Proved to work with same performance, but saves time in HLT1, and power consumption. Simpler form of primitive than a track, evaluated within a board
- Proves the concept, and at the same time prepares for a next tracking stage
- In the future, might be candidate for pushing further down to the front-end

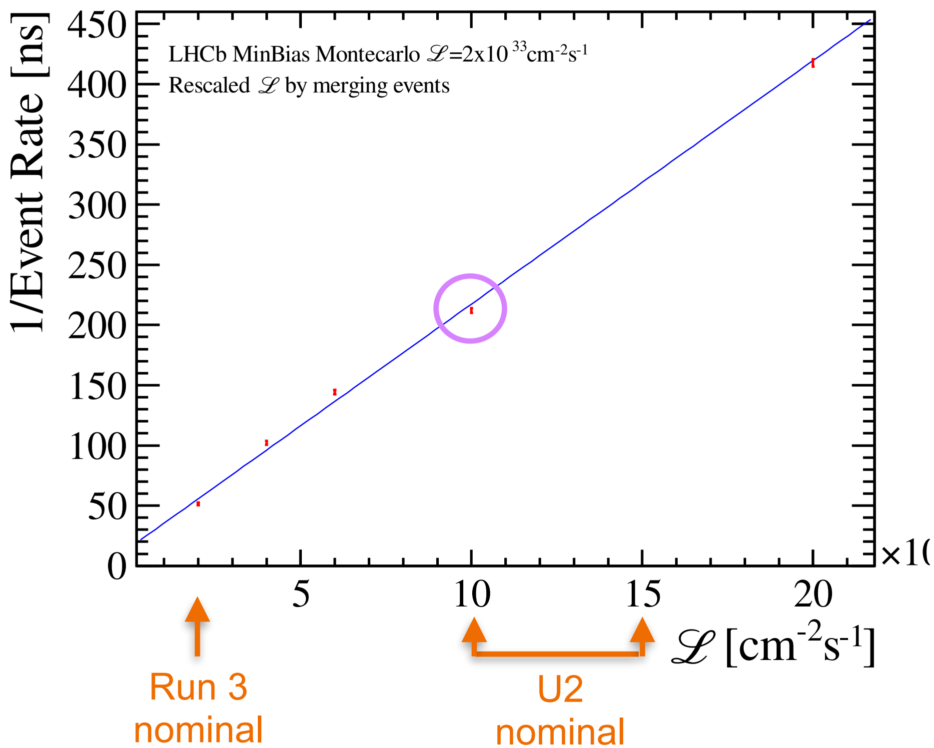


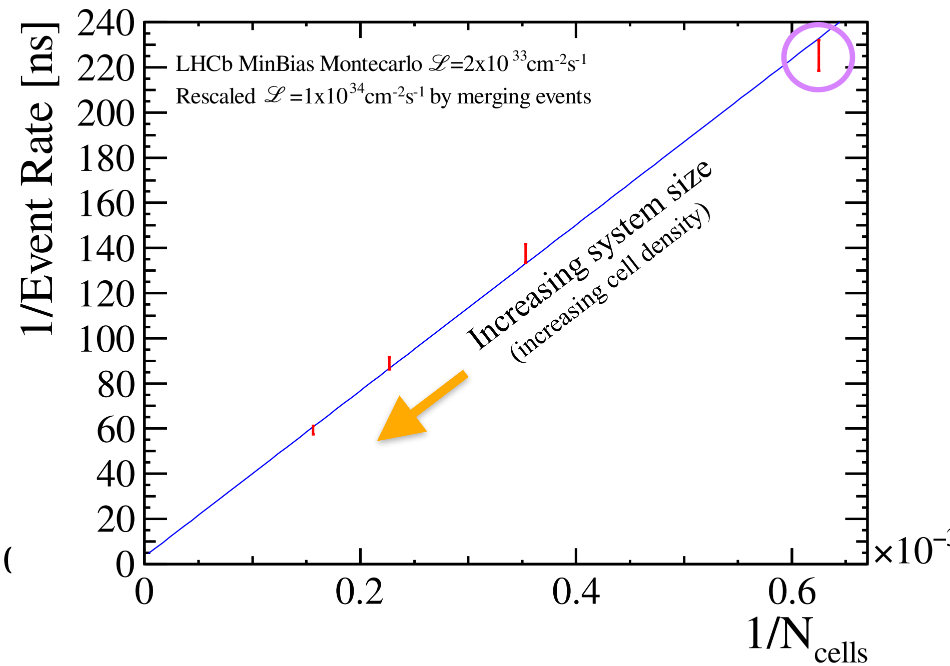
Linearity of retina processing: VELO clustering vs μ



- Plots shows latest measurements of VELO clustering throughput vs Luminosity (μ)
- Very linear behavior, for various conditions, up to the highest L expected in Run 3
- Question is what happens with more complex, multi-board processing, at even higher luminosities ?

Linearity of retina processing: VELO tracking*





- Good behavior, intrinsic to the retina methodology
- Here tested with LHCb VELO simulated events, merged to emulate large μ
- Throughput also linear in hardware size: see right plot

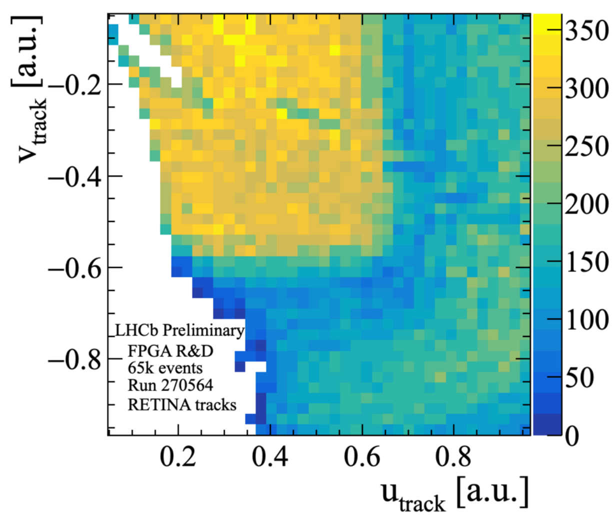
➡ **Size scales ∝ Luminosity** (Remind HLT2 size grows as L^2)

NB: Data from **real hardware** retina processor @LHCb testbed

✳ [special thanks to F. Lazzari and F. Terzuoli]

Working prototypes

- The system may look complex, but working prototypes exist already, complete with optical network at full speed
- A complete Retina demonstrator has been installed and tested at the LHCb TestBed facility (near Control Room).
- Reconstruct a quadrant of the VELO detector.
  - Detector chosen for compactness, small test system
- Implemented in 8 PCIe-hosted, commercial FPGA cards (Stratix-X)
- Tested with:
  - LHCb MC data at nominal luminosity (2x10^{33}).
    - Event rate: 19.6 MHz. Power 550W
    - Longest continuous run: 27 days w/o errors
  - LHCb live data during 2023 run (July and September)

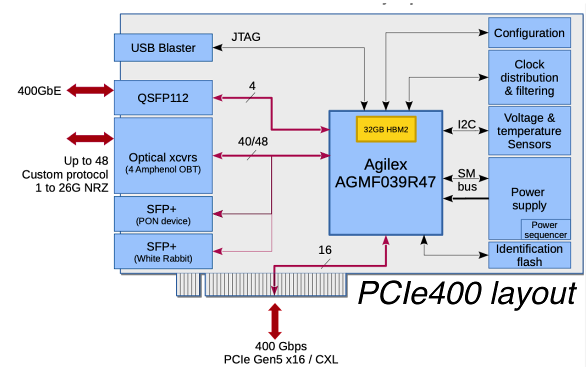


On the path to U2: Data Acquisition Enhancement TDR

Recently submitted to LHCC:

- PCIe400: new readout board with 400 Gb/s / 4 MLE FPGA
  - 48 GBT/lpGBT links compatible with PCIeGen5 or 400GbE
    → output bandwidth x4 compared to present generation

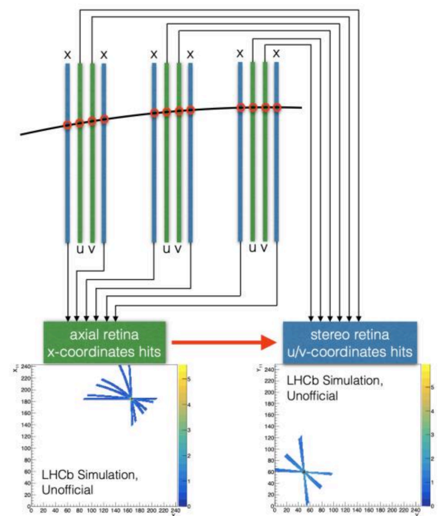
  <u>U2 vision:</u> <u>keep pace with FPGA technology evolution</u>



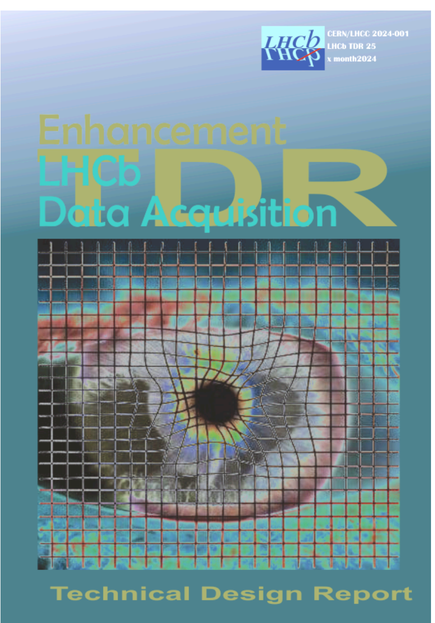
PCIe400 layout

- Downstream tracking with FPGA (DWT)
  - Retina-based T-track primitives reconstruction from SciFi hits
  - Accelerate HLT1/2 tracking (→enhance reconstruction of Long-Lived)

  <u>U2 vision:</u> <u>experiment with reconstructing tracking primitives at 30 MHz</u>



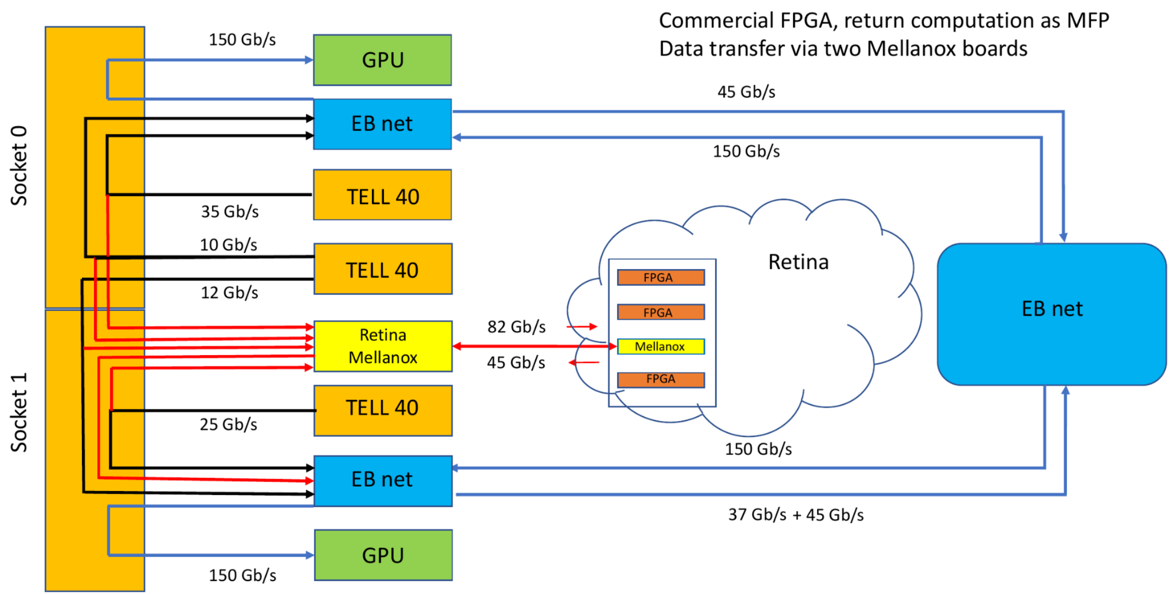
DWT tracker

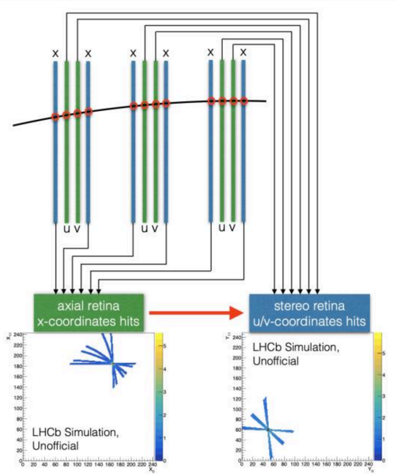


- Connected developments: DWT to be deployed in PCIe400 boards
- In Run 4, use a separate set of boards to avoid disturbing current EB
- Run 5+ prospect: **integrate in readout** and **extend to further detectors**

What the system will look like in Run 4

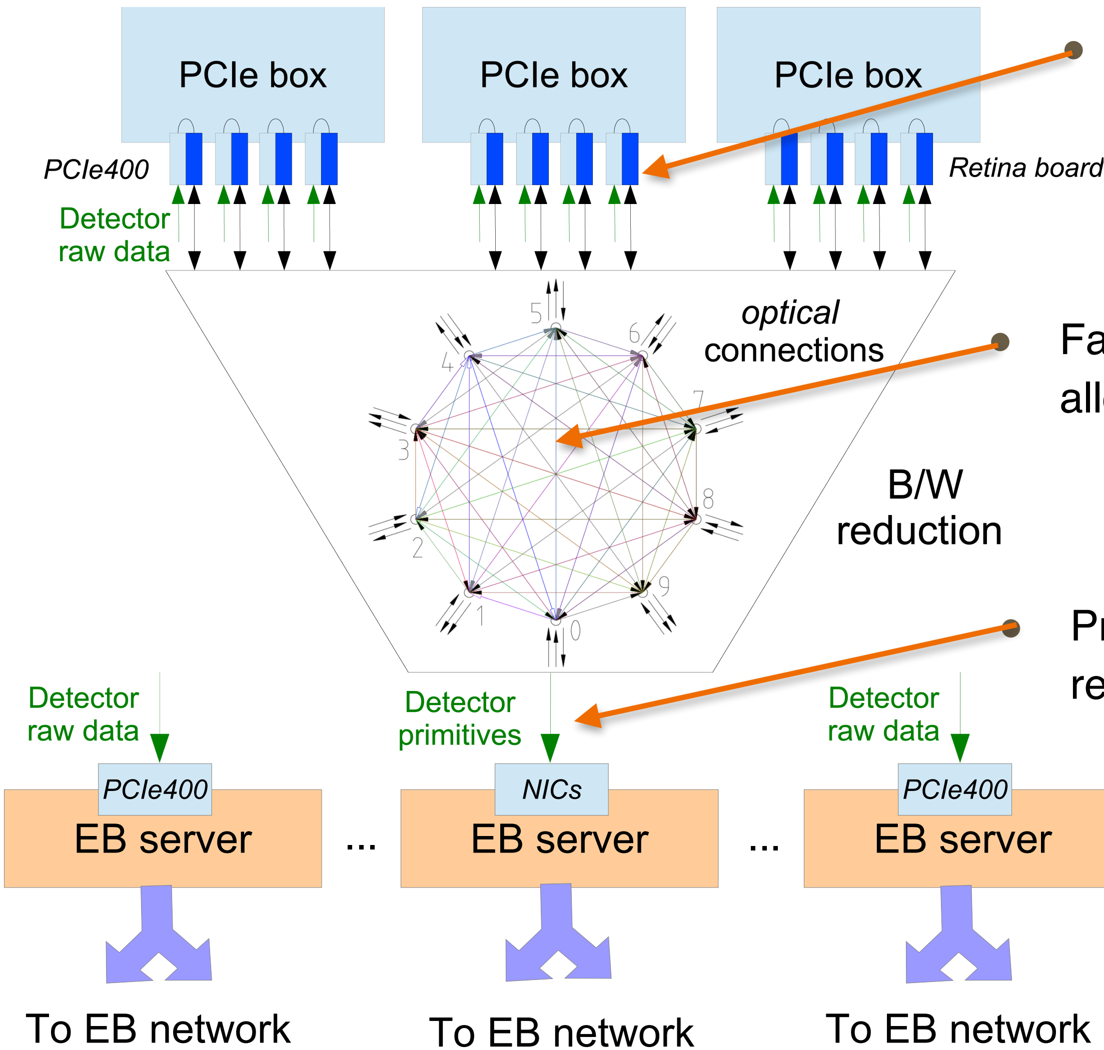
- 24 DWT Server Boxes (up to 6 FPGA each) connected to SciFi EB nodes.
  - Provide T-track primitives (seeding) to HLT1 (and HLT2)
  - Connected through InfiniBand cards (same as internal EB connections)
- Minimal interference with current system (that will still use same TELL40 boards)
  - Drawback is significant extra hardware needed (~100 PCIe400)
- Frees ~1/3 of HLT1 power for other uses + an indetermined amount of HLT2 power
  - Fit downstream tracking comfortably in HLT1 (plus any desirable extras)
  - Enhance trigger efficiency for Long-Lived Particles





DWT tracker

What the system could look like in U2



- PCIe400 readout boards "expanded" with reconstruction functionalities. Exploit readout boards (+ extra ?) [increase number, or add-on]
- Fast optical point-to-point interconnections allow finding track primitives across layers
- Primitives fed to EB nodes (as from readout) for building and delivery to HLT1
- EB nodes as usual (w/ PCIe400)

TBD: Evaluate additional hardware needed (expect $\lesssim$ readout needs)

Applications to U2 tracking

- A 'minimal' set of Primitives available to HLT1 and HLT2 could be:
  1. VT tracks
  2. T-tracks from MT (pixels)
  3. Muon stubs
- FT and UP might be used at the hit (cluster) level
- Reconstruction could then use them to:
  - Matching T-tracks / VT tracks + add UP hits
  - Attach VT + UP hits; then FT
  - Match Muons /T-tracks + add UP hits
- Augmented set might additionally have:
  - Standalone UP and FT primitives
  - VT-UP primitives (as a single entity)
  - VT-MS primitives
  - Muon-MT stubs
- Some more details on next slide

Applications to U2 tracking

- **VT tracks:**
  - Advanced stage: first implementation exists already as Run 3 prototype (also an early VELO-UT)
  - Resources needed pretty much known. But timing to be studied (see later)

- **T-tracks from MT-pixels:**
  - A lot will be learned in the making of the DWT
  - Pixel geometry algorithmically easier than SciFi - similar to VELO algorithm

- **Muon stubs:**
  - Preliminary: first attempts exist
  - Local data reduction: reduce bandwidth while keeping efficiency high
  - Good physics potential: enable upfront μ trigger/downstream/low-pt muons.

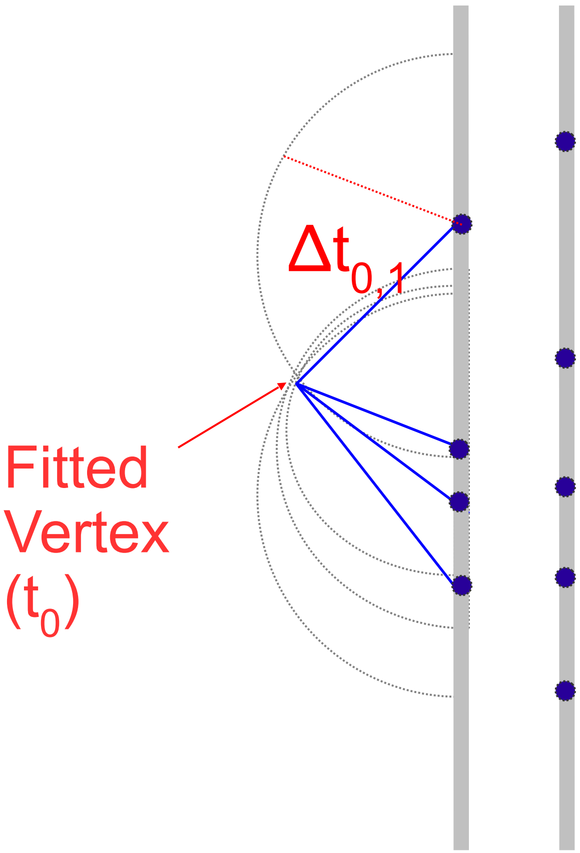
Alignment considerations

- FAQ: Can we keep misalignments into account in primitive reconstruction ?
- The short answer is yes.

More in detail:

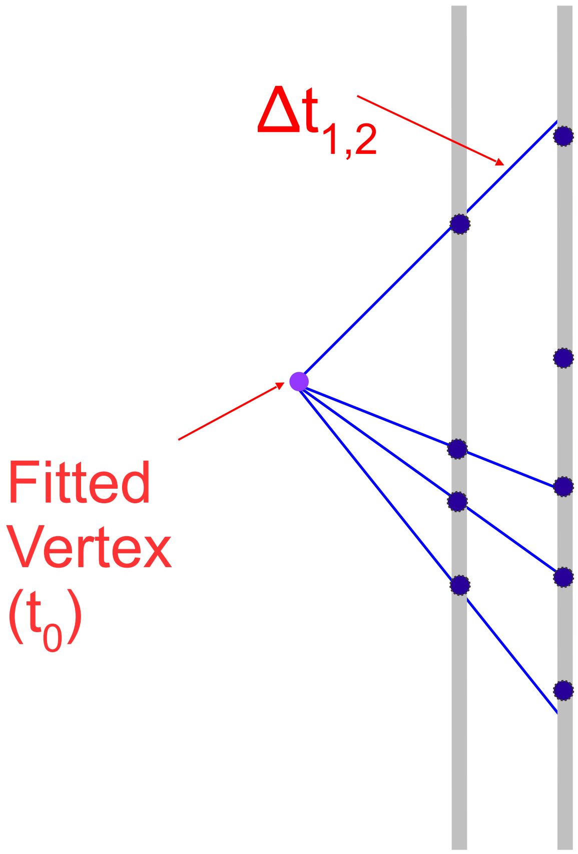
  - Primitives less sensitive to alignments: precision fit demanded to HLT. Still, large misalignments affect pattern recognition besides parameter determination
  - Alignment constants CAN be stored/updated in FPGA boards, and applied on-the-fly while performing primitive reconstruction (no need to reload firmware)
  - The mechanism of getting updated constants from the current LHCb setup and applying them has been tested already on the hardware demonstrator for VELO
  - Indeed, a possible enhancement is to have local/internal alignment constants evaluated within the FPGA boards themselves, and applied directly at that level
  - That is all to be demonstrated, but if successful is potentially very helpful

A word about time-based tracking



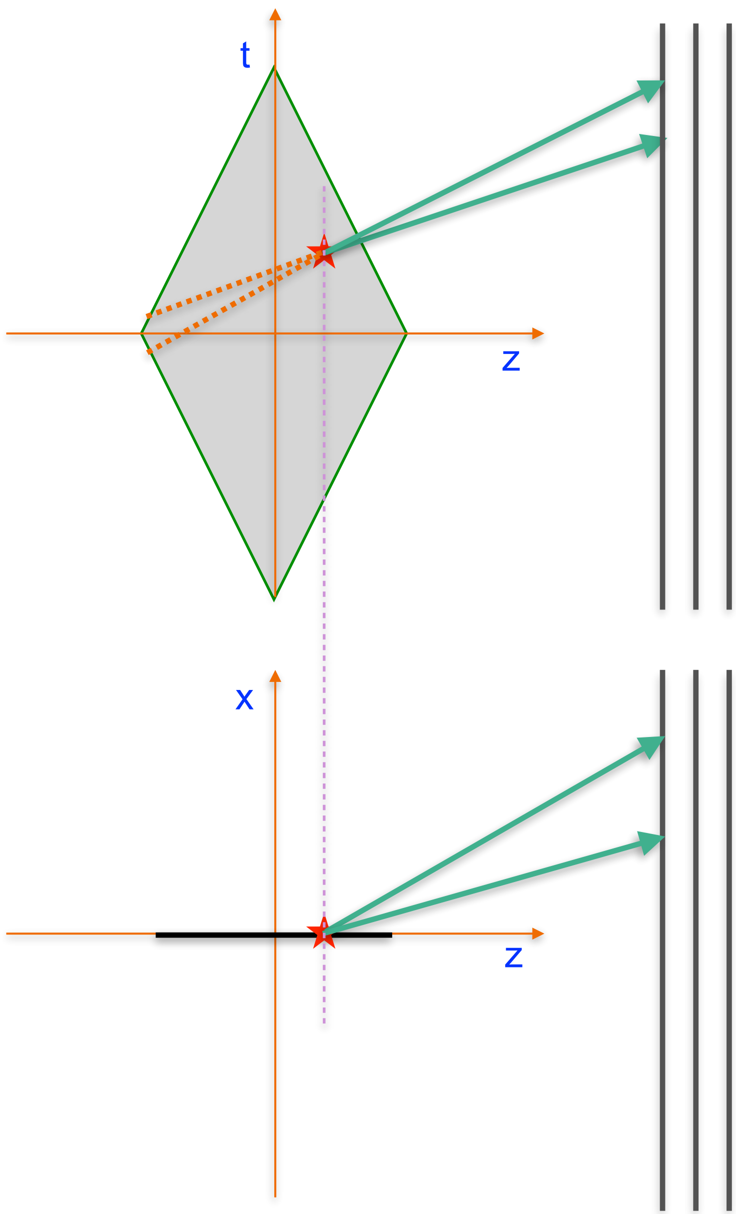
- Time information often seen just in the light of vertex separation
- However, its role in pattern recognition might be even more important
- Consider a single layer of planar pixel detector, where each hit has (x,y, z0, t) information
- In case the time t0 of track production known, an half-sphere is defined by the hit
- If > 3 points on the same layer, can determine the space and time position of the origin vertex, without other information, simply with a local fit
- Opens the possibility of single-layer vertex finding, even before track finding

A word about time-based tracking

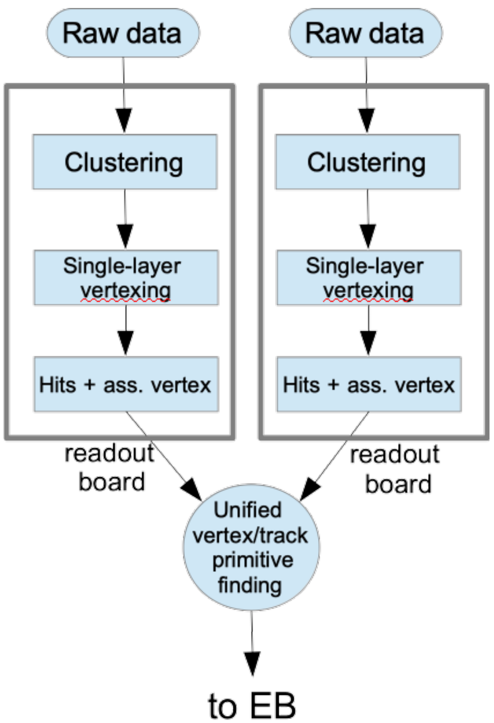


- When moving on to a further layer, the knowledge of the vertex and the time different with hits in the previous layer allow to restrict the possibile associations
- If can exploit this mechanism well, could perform better pattern recognition, and all based on *local information* only (=suitable for primitive finding in FPGAs)

Tracking in space-time



- In this scheme, vertex candidates are found *before* tracks, within a readout board
- In a second stage, board communication allows to put everything together in 4D
- Expect to help pattern recognition, HLT1 does the precision fit
- Quantitative studies TBD



Final considerations

- The large data rate at LHCb-U2, unprecedented in HEP, and likely new advanced detectors, will put us in a very new situation, in which real time data processing takes a very central role in the success of the experiment.
- While commercial processor powers will still grow, the large luminosity and possibly timing data will require the use of specialized tracking devices
- Pre-reconstruction at readout level is a processing stage we have only began to explore, and could be crucial to the success of complex tracking in U2
- Moving most of pattern recognition to the pre-build stage, enables better reconstruction in HLT1 / HLT2, yielding a more affordable overall system
- Need to write primitive-based Allen code, and revisit alignment -> work starting
- **Needs the effort and ingenuity of many people, but will enable LHCb to do the frontier physics we want in U2**

Good people & good machines together will succeed