Timestamps and segfaults – a brief overview of dataflow in LHCb

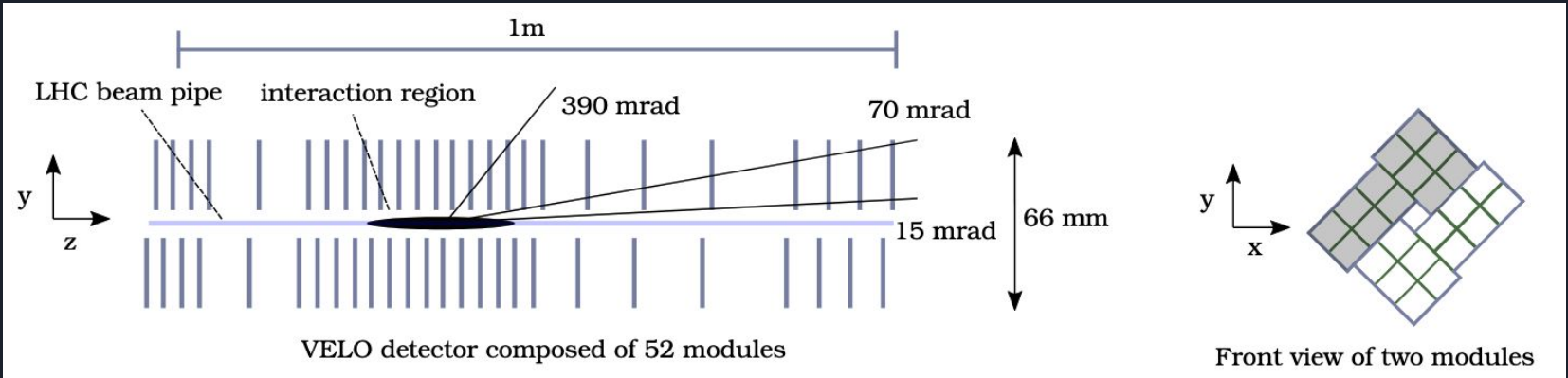
FASTER Meeting Maastricht 20/09/2024

Justus Rudolph

Motivation – LHC Upgrades

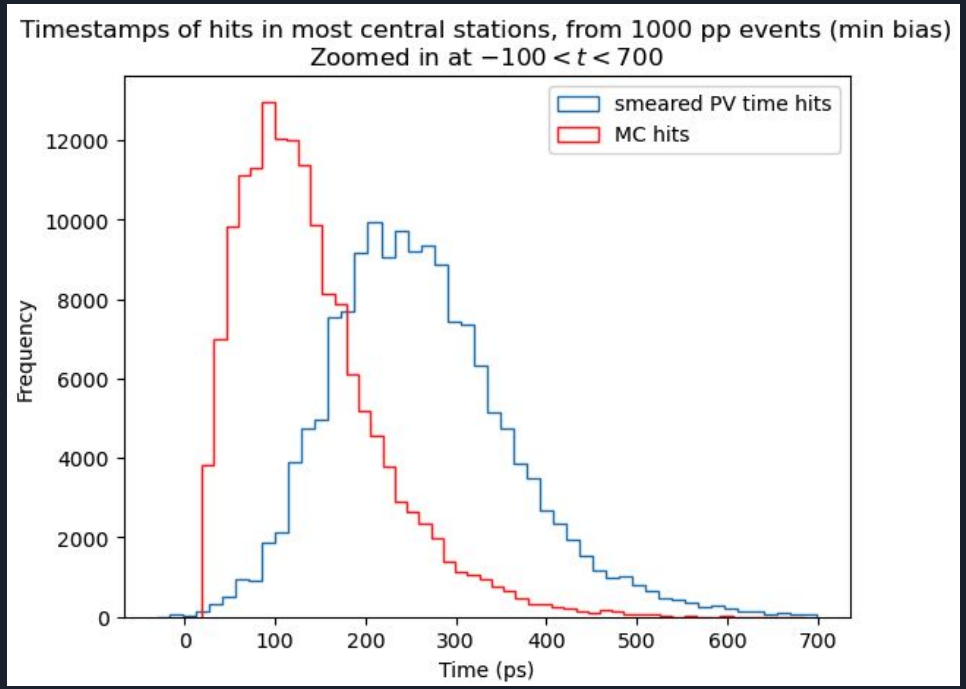
- HL-LHC: Vastly increased expected pile-up
- → Much higher detector occupancy
- → Increasing need to separate tracks with a fourth dimension

The VELO detector – Schematic



Pérez, Daniel & Neufeld, Niko & Núñez, Agustín. (2022). Search by triplet: An efficient local track reconstruction algorithm for parallel architectures.

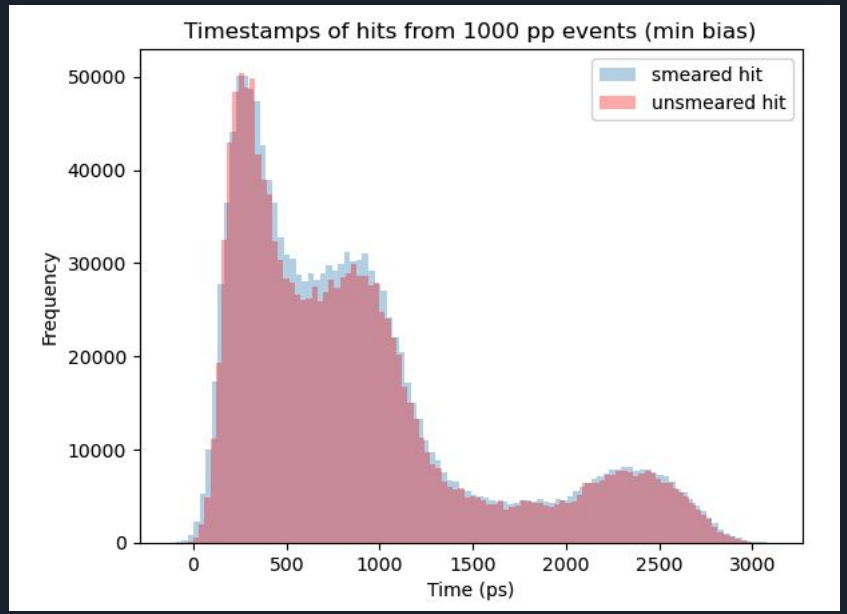
Data to work with

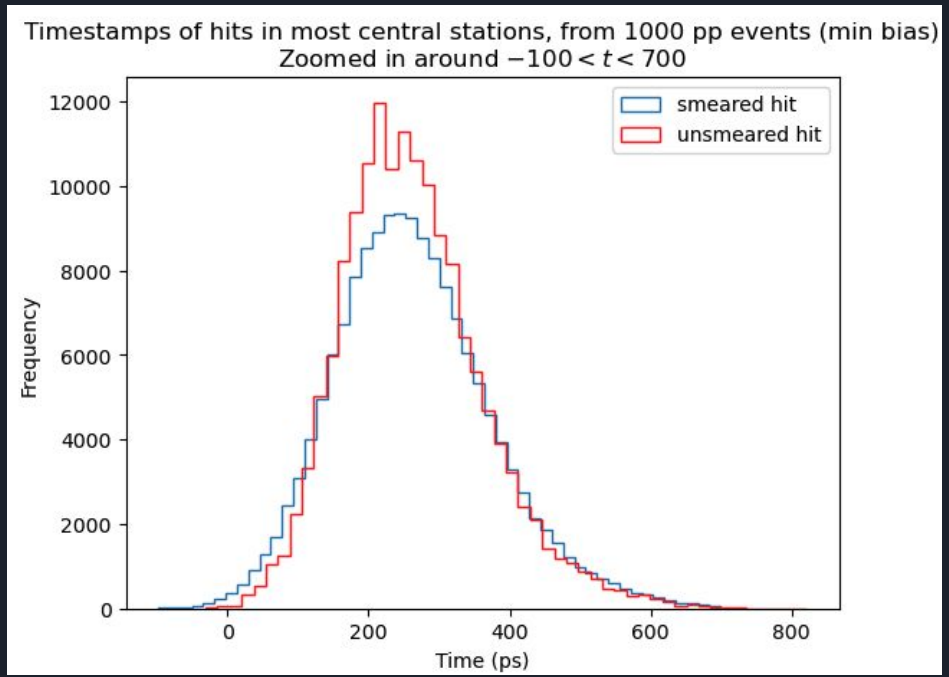


- Current MC hit data contains only relative time to primary vertex
- Every primary vertex (PV) is at t = 0
- → Introduce smearing over PVs
  - Width of bunch at collision ~7.5cm
  - → Back of envelope calc: 250ps crossing time
- → σ = 60ps reasonable: >95% within nominal bunch crossing

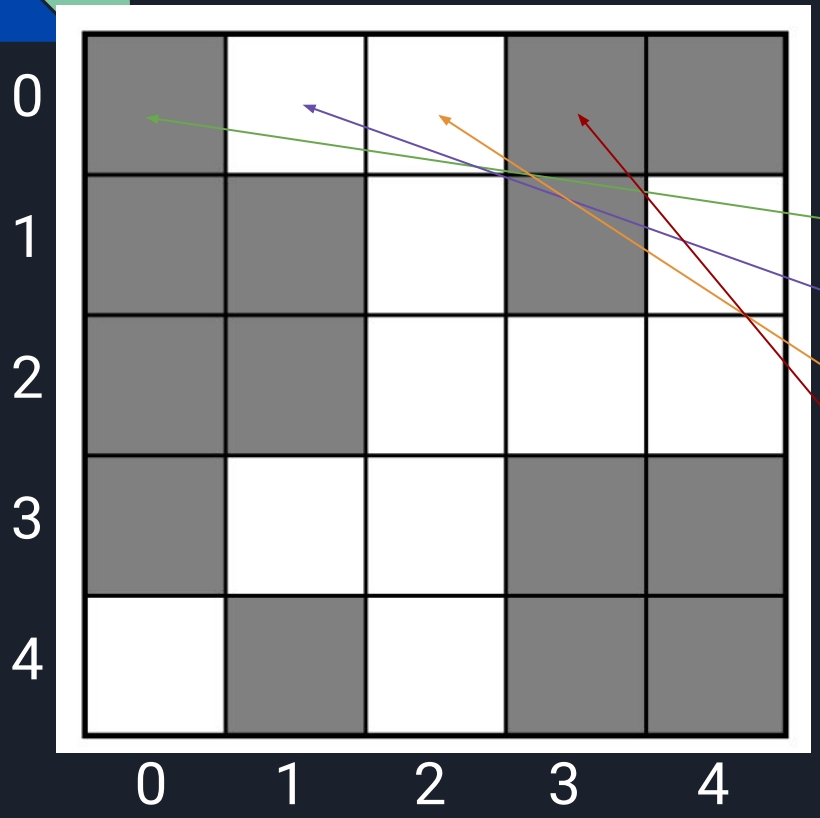
Data to work with – smeared PV timestamp

Also add 50ps smear on hits:





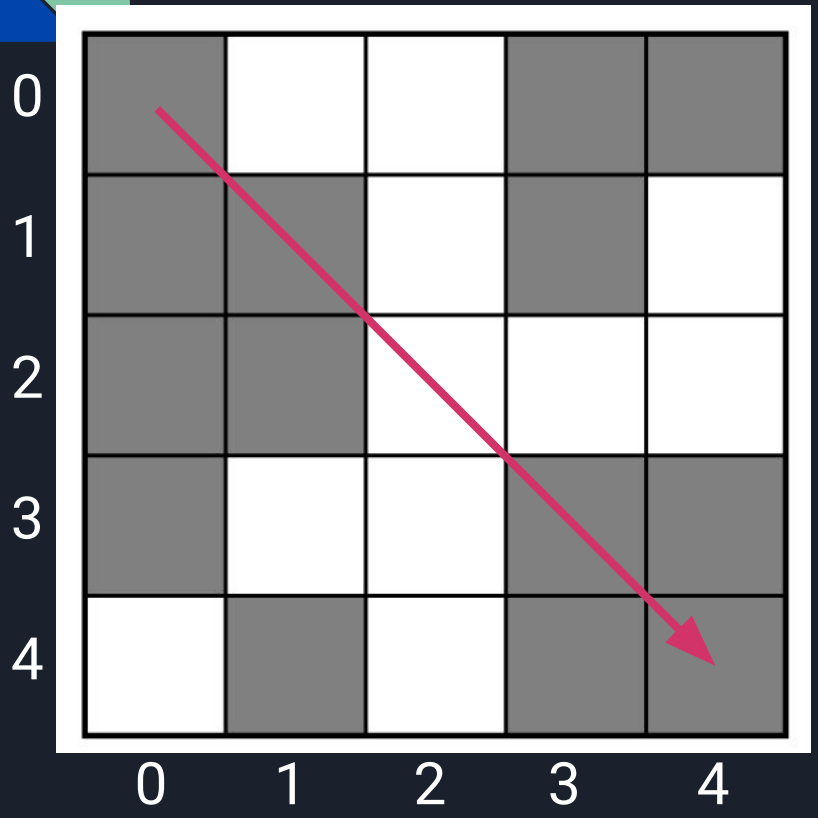
How to pass pixel information?



Position:x	Position:y	Value	Total bits
000	000	1	7 (8)
001	000	0	7 (8)
010	000	0	7 (8)
011	000	1	7 (8)

8 bits per pixel →
25× 8 = 200 bits

How to pass pixel information?



Sender and receiver agree on order:

Position:x Position:y Value

000 000 10011 11010 11000 10011 01011

31 bits in total
→ ONE 32 bit “word” to send vs 6 or 7 with naive method

How to pass pixel information: Superpixels

Superpixel format: Settle on info of 8 pixels shipped together

6.1 SuperPixel format

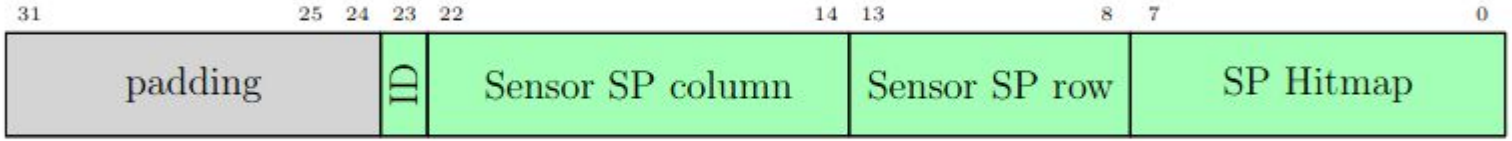


Figure 7: Velo SuperPixel data fragment

A. Fernandez Prieto, K. Hennessy, G. Bassi; LHCb Upgrade VELO
TELL40 data processing; EDMS 2086526 v.5 status In Work access Public;
VELO_TELL40_Data_Processing_v5.pdf modified 2023-03-29 09:28

Superpixel hitmap

3	7
2	6
1	5
0	4

How to pass pixel information: Superpixels

Superpixel format: Settle on info of 8 pixels shipped together

6.1 SuperPixel format

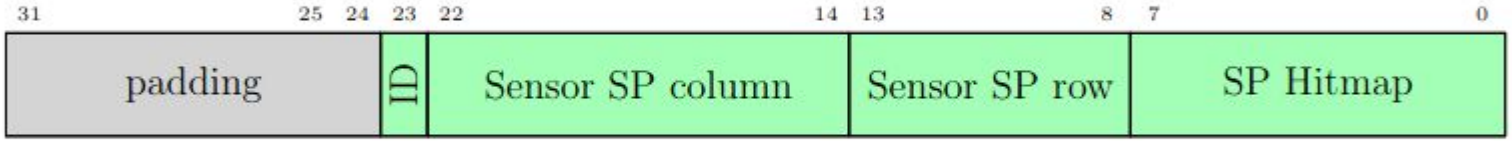


Figure 7: Velo SuperPixel data fragment

A. Fernandez Prieto, K. Hennessy, G. Bassi; LHCb Upgrade VELO
TELL40 data processing; EDMS 2086526 v.5 status In Work access Public;
VELO_TELL40_Data_Processing_v5.pdf modified 2023-03-29 09:28

Superpixel hitmap

3	7
2	6
1	5
0	4

Problem: Space: 1 bit/pp, while Time: O(10) bits/pp. Why?

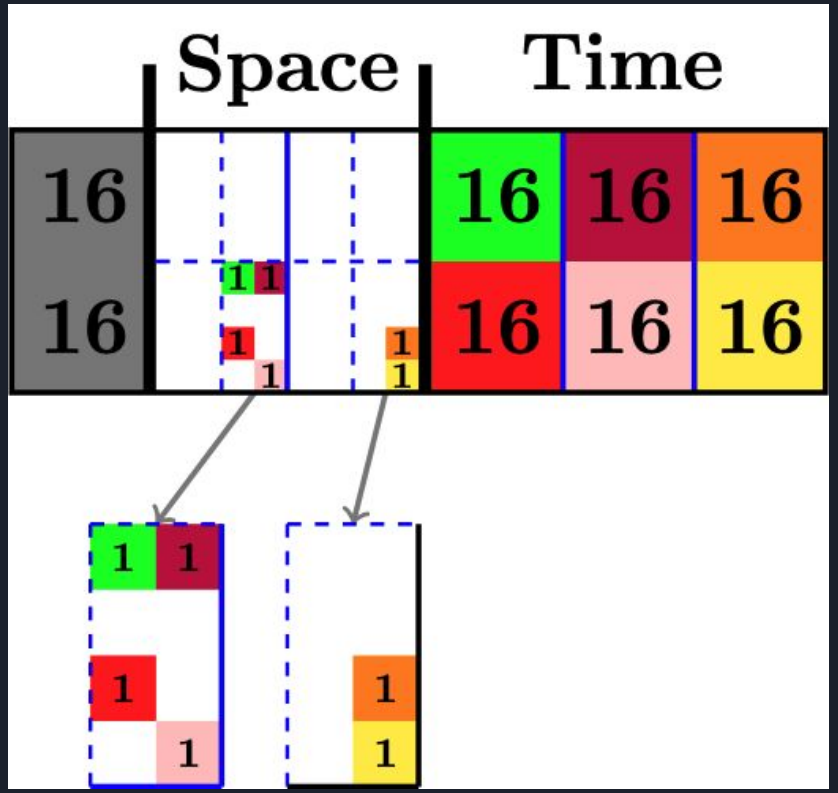
Bit requirements for timestamps

- Goal is 50ps resolution
- Design choice: for some wiggle room: 16ps resolution on software side
  - I.e. the 16 bit word 0000 0000 0000 0001 corresponds to 16ps
- Bunch spacing: 25ns
  - → Allowing for spillover: use a 50ns window
- → Bits needed: 50ns / 16ps ~ 3000 > 2^{11}, so 12 bits needed.
- In processing, 12 bits means 16 bits → work with 16 bits per timestamp
  - Subject to change on the data transfer side. How?

Encoding of superpixel words

Superpixel hitmap

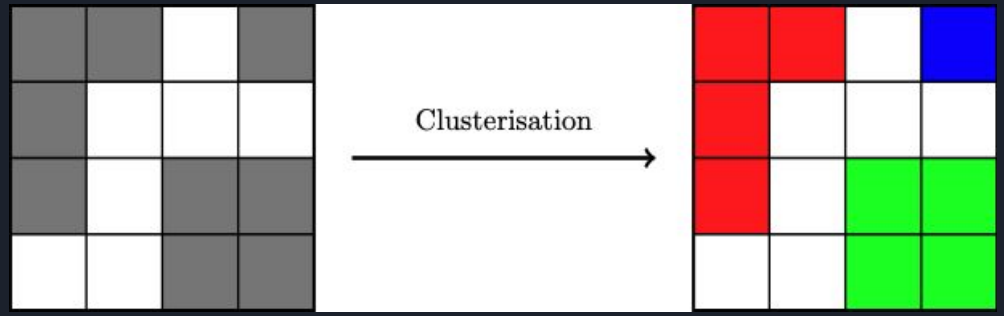
3	7
2	6
1	5
0	4



- Data is passed as one large contiguous chunk of memory
- → Writing 12 bits per timestamp is thus technically possible

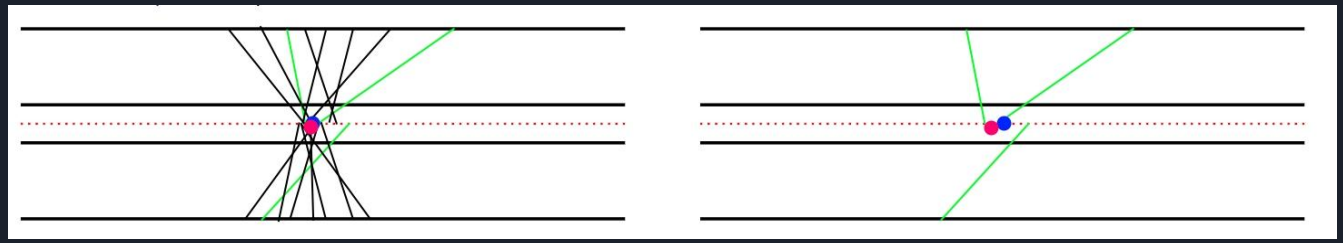
Encoding of TimeClusters

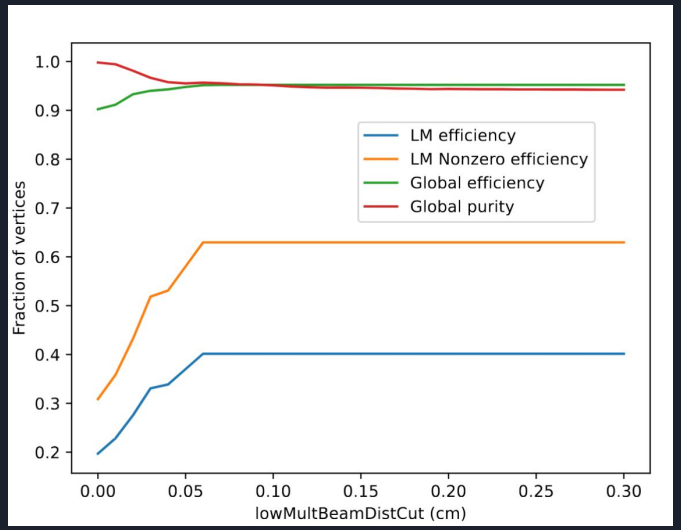
- Same as with superpixels, but 32 bits per cluster.
- 16 for the time, 16 for the uncertainty
  - This is to ensure ease of use in future addition of times in clusterisation



Work in ALICE: Service Task

Low multiplicity vertex creation: should be the easiest case right?



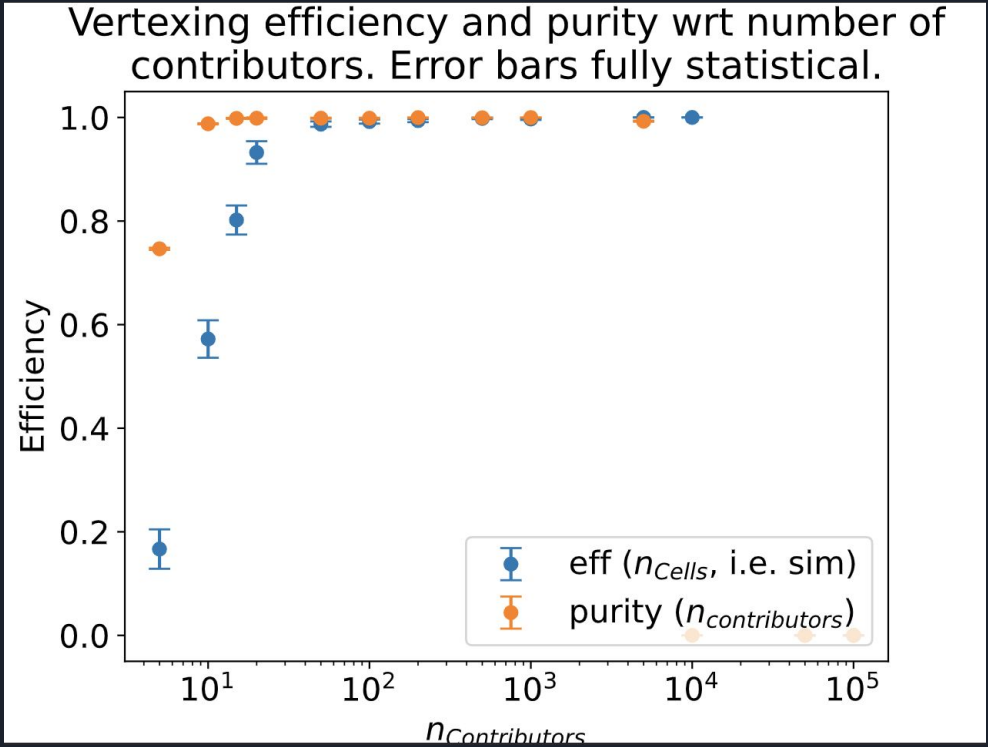


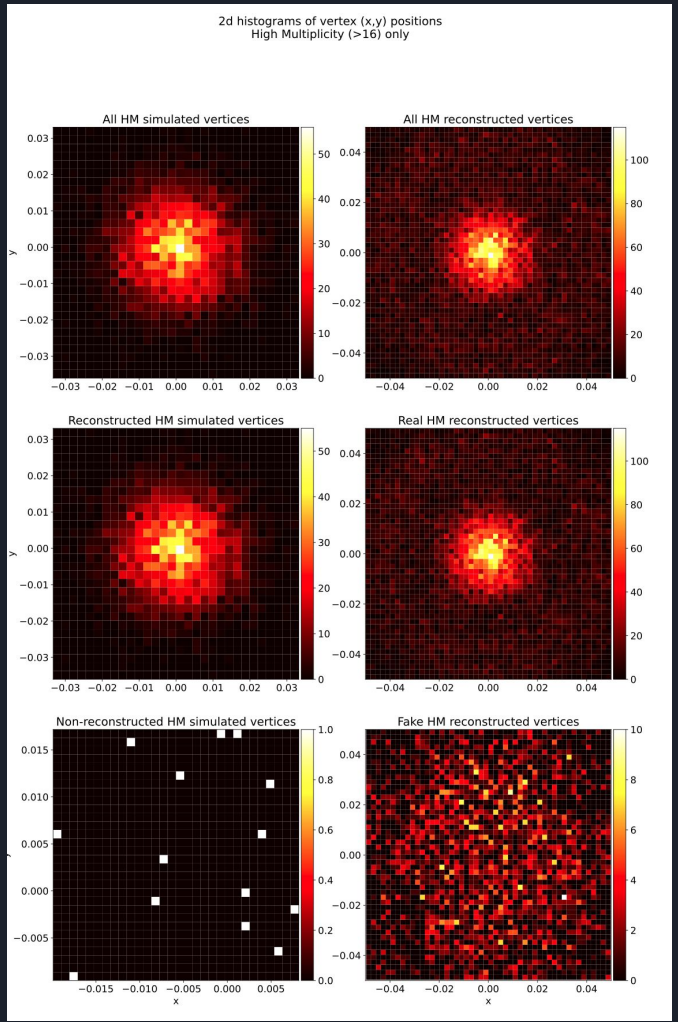
Vertexing parameter to investigate: lowMultBeamDistCut

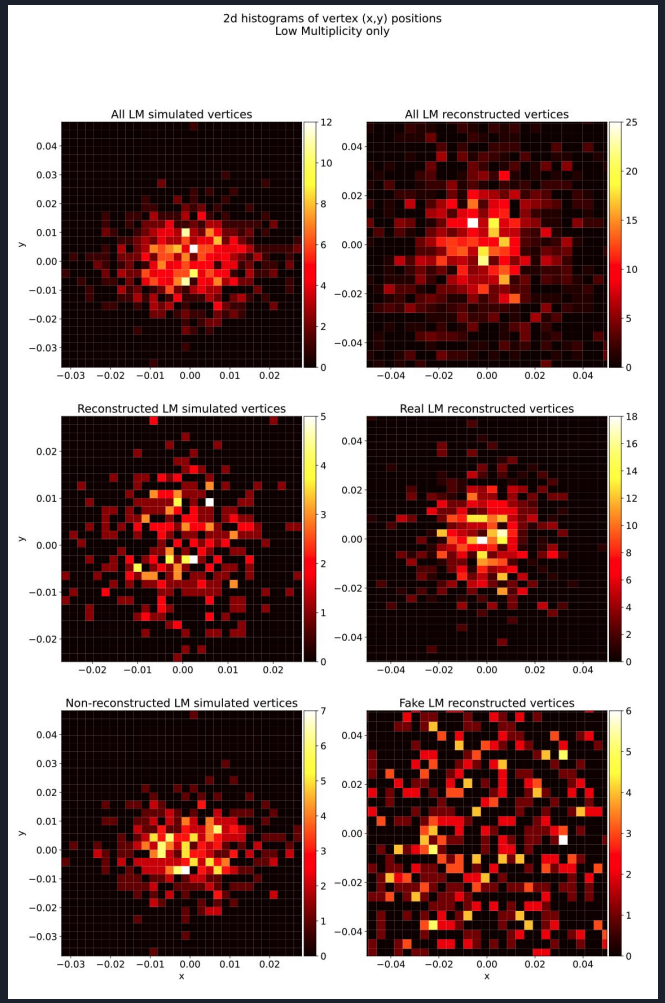
Summary and plan

- Investigate timestamps in finalised VELO tracks (consolidated and Kalman filtered)
  - Discrepancies somewhere? (outside of statistical and simulated timestamp uncertainties)
    - first hit time > last hit time?
    - Faster than speed of light travel?
- Look into data with higher pile-up: optimally with Run 4 parameters & expected lumi
- Prepare infrastructure for the point where MC timestamps are more physical
  - Fairly simple: just remove my own smearings
- Start implementing (simple) cuts on clusterisation and tracking with timestamps
  - E.g. a cluster cannot have two pixels with $\Delta t > 1000$ps
  - E.g. two subsequent hits in tracklet creation need to be within $\Delta t <$ Xps, where X can be determined by the slope of the 3-hit tracklet

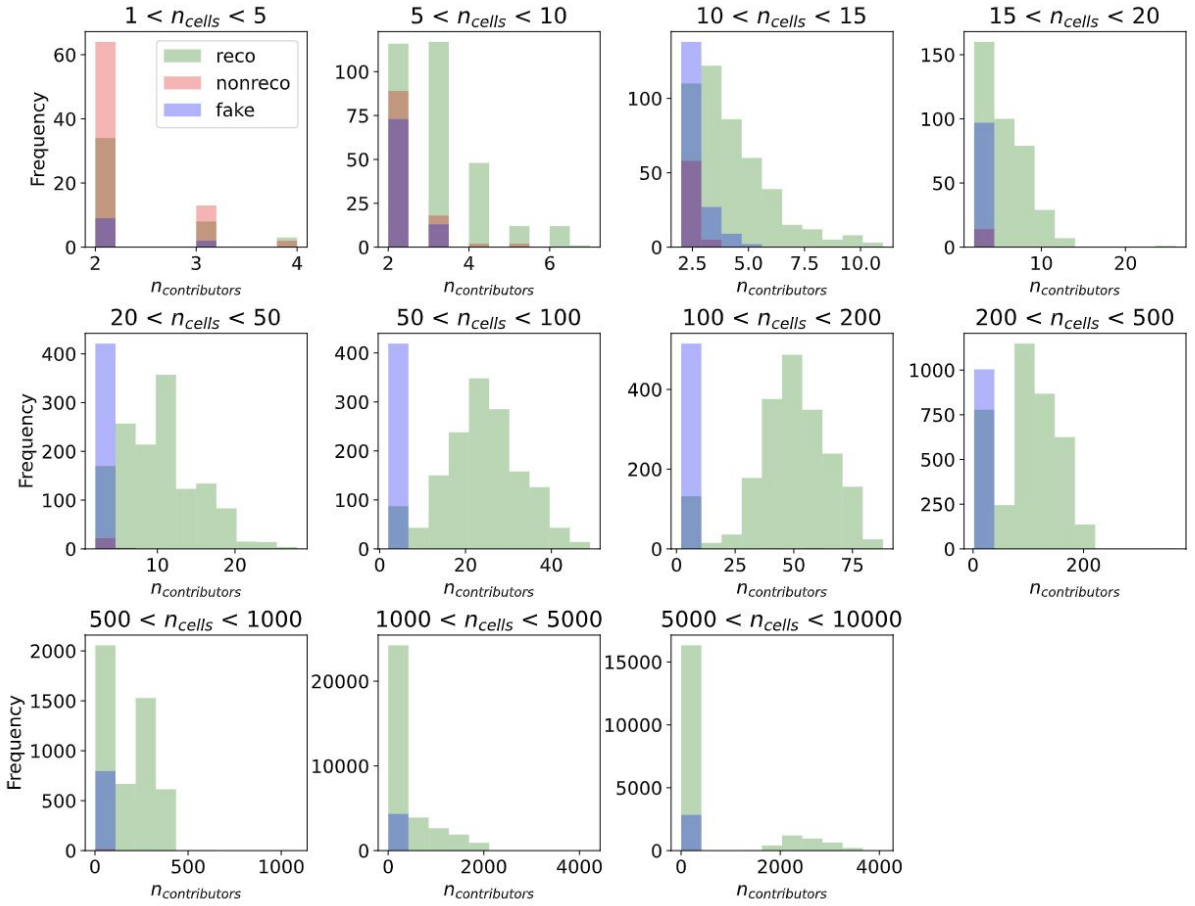
Backup: Service Task plots




2d histograms of vertex (x,y) positions
High Multiplicity (>16) only
All HM simulated vertices
All HM reconstructed vertices
Reconstructed HM simulated vertices
Real HM reconstructed vertices
Non-reconstructed HM simulated vertices
Fake HM reconstructed vertices
x
y


2d histograms of vertex (x,y) positions
Low Multiplicity only
All LM simulated vertices
All LM reconstructed vertices
Reconstructed LM simulated vertices
Real LM reconstructed vertices
Non-reconstructed LM simulated vertices
Fake LM reconstructed vertices
x
y

Distributions of number of tracks of reconstructed vertices
for various bins of number of tracks of the simulated vertex.



Distributions of distance from reconstructed vertices to their simulated vertex
for various bins of number of tracks of the simulated vertex.