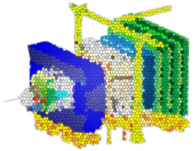
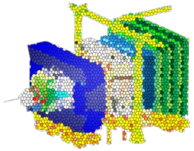


Evolution of the LHCb software trigger (RTA): past, present and future

Sascha Stahl, CERN,
on behalf of the LHCb RTA project

LHCb Upgrade II workshop,
29-31 March, Barcelona

RTA – Real-time analysis



Real-time or real time describes various operations in computing or other processes that must guarantee response times within a specified time (deadline), usually a relatively short time. … (Wikipedia)

Goal of the trigger system (inspired by wikipedia, trigger (particle physics)):

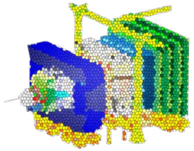
Make it possible to

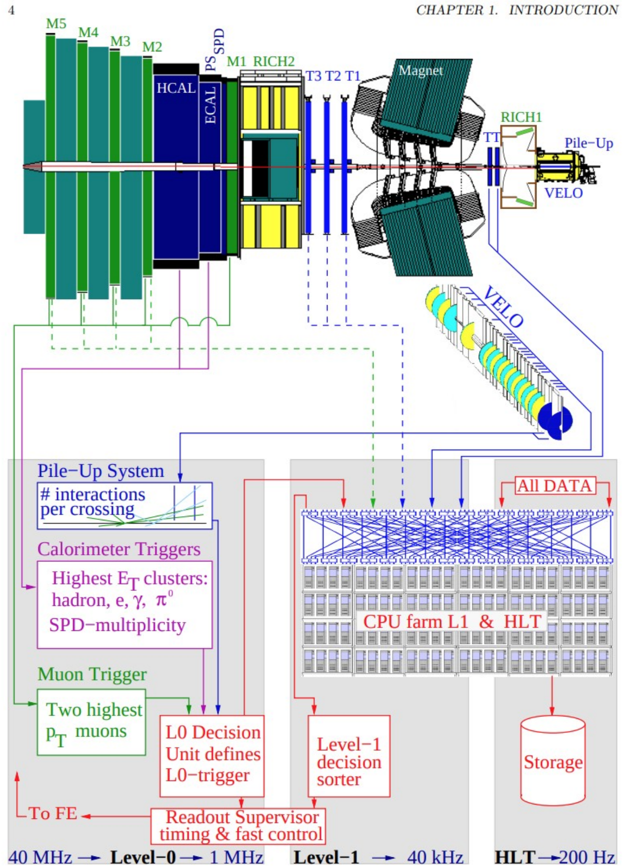
- Analyze detector data in real-time,
- to keep only the small fraction of events and data which is “interesting”

Necessary due to real-world limitations in computing power, data storage capacity and rates.

(I’ll spare you the Bing/ChatGPT answers.)

Very distant past (2003)



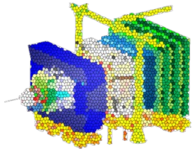


LHCb Trigger system : Technical Design report

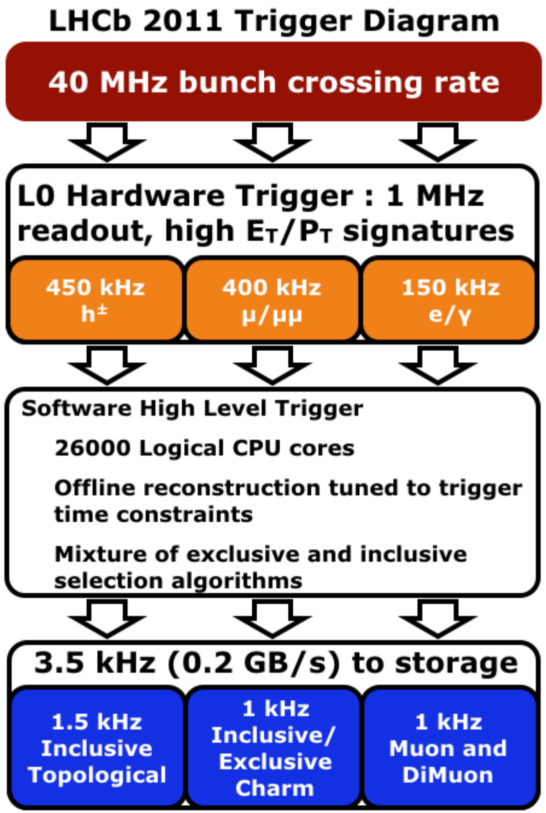
LHCb detector design:

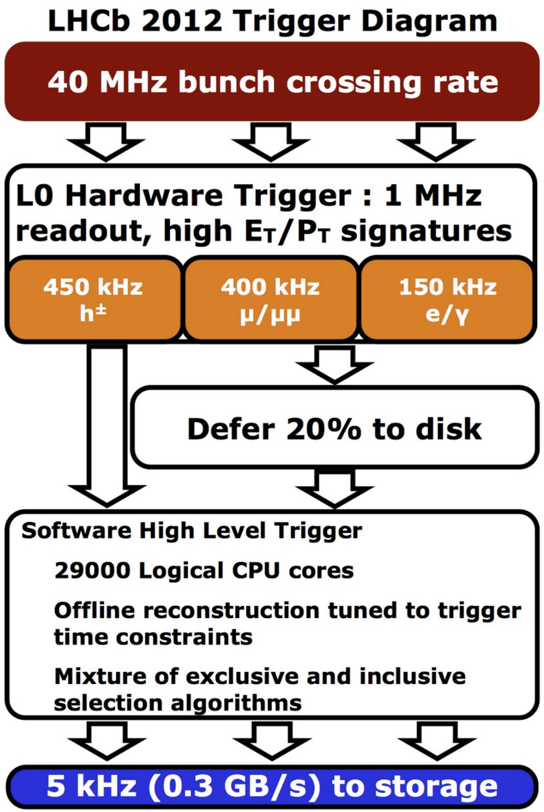
- L0 trigger reduced data rate from 40 to 1 MHz
  - Only based on muon and calorimeter information
- Level-1 (software but tight timing requirements) acted as L0 confirmation (whatever that means), and reduced rate to 40 kHz
- HLT used exclusive selections of beauty decays to get rate down to 200 Hz
- Running on 1800 CPUs

Distant path (Run 1)



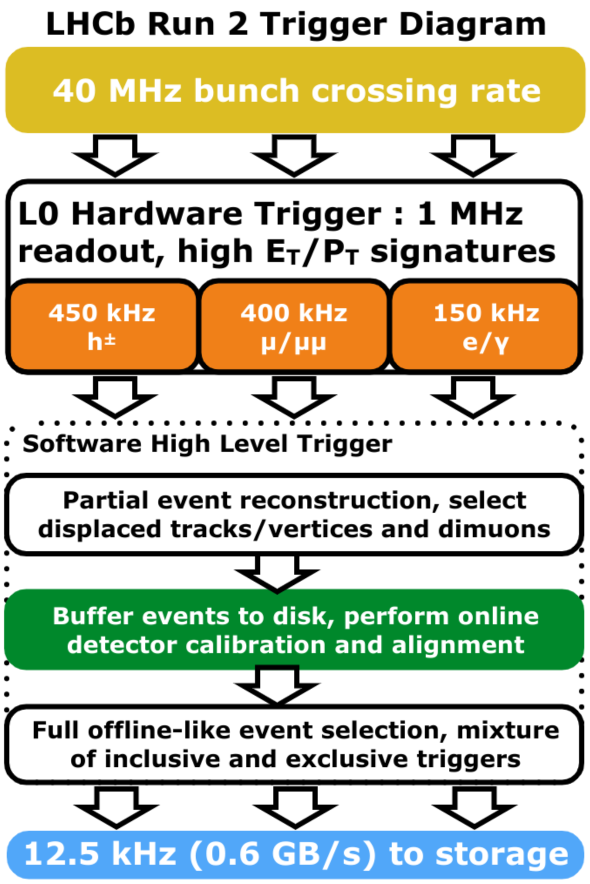
Eventual trigger system in Run 1:



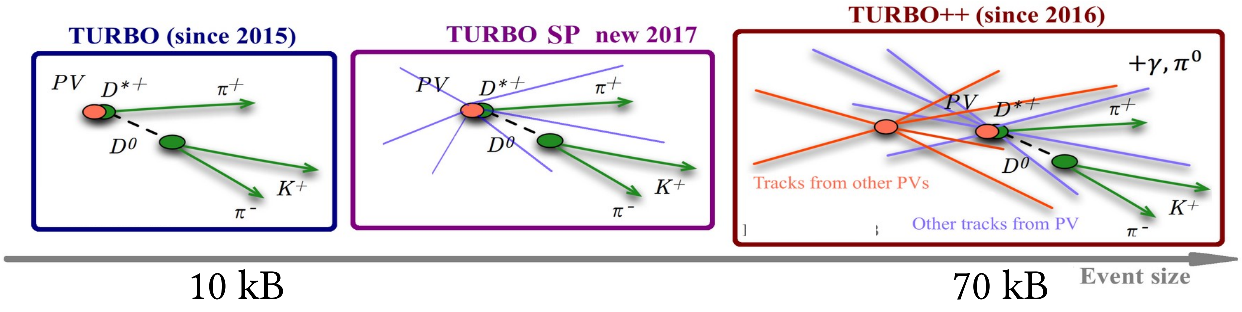


- HLT1 had to run (high-pt) tracking including Kalman filter, HLT2 almost full event reconstruction
  - Possible as much more computing power available, continuous work on algorithms
- Much bigger output rate than originally foreseen.
  - Based on inclusive selections for full beauty programme
  - Charm selections were not part of the TDR :D.

Past (Run 2)



- Moving disk buffer between HLT1 and HLT2, increased number of CPUs and <u>hard work by people</u> enabled
  - Real-time alignment and calibration
  - Real-time reconstruction with analysis quality reconstruction
  - Ability to use trigger output for analysis and discard raw detector information in trigger



Innovations enable more physics (biased choice)

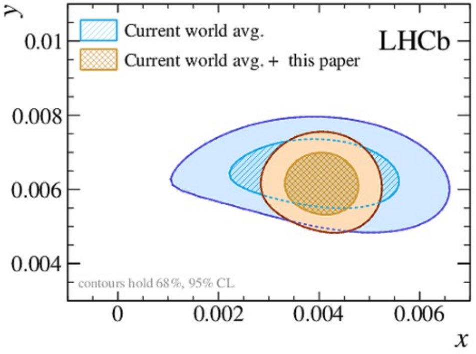
<u>There is a charm programme</u> :D

<u>Observation of Charm CPV</u>:
Increased statistics due to fewer bandwidth restrictions in Run 2

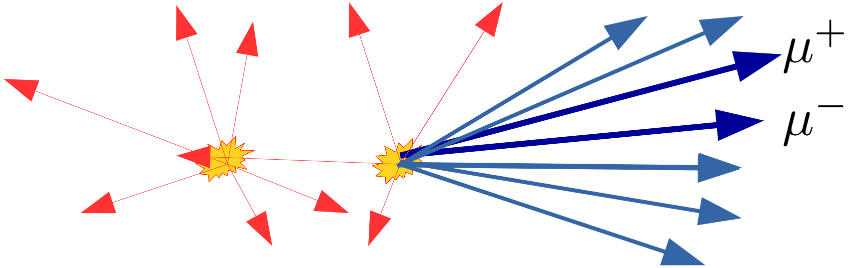
$$\Delta A_{CP}(D^0 \to hh) = (-15.4 \pm 2.9) \times 10^{-4}$$

<u>Observation of mass difference between neutral Charm-Meson Eigenstates</u>:
20 times more events in Run 2 due to much better tracking in HLT2 (esp. Downstream tracks)

$D^0 \to K^0_S \pi^+ \pi^-$

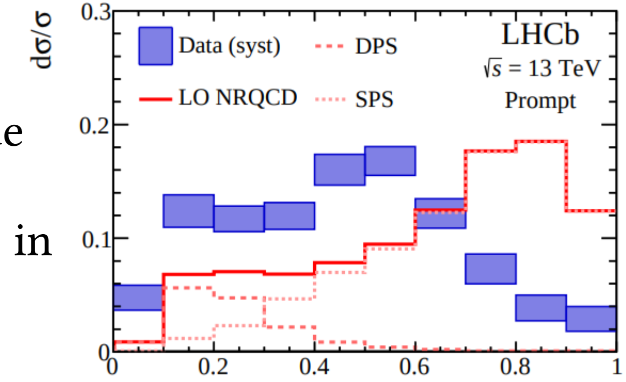


<u>J/Ψ production in Jets</u>



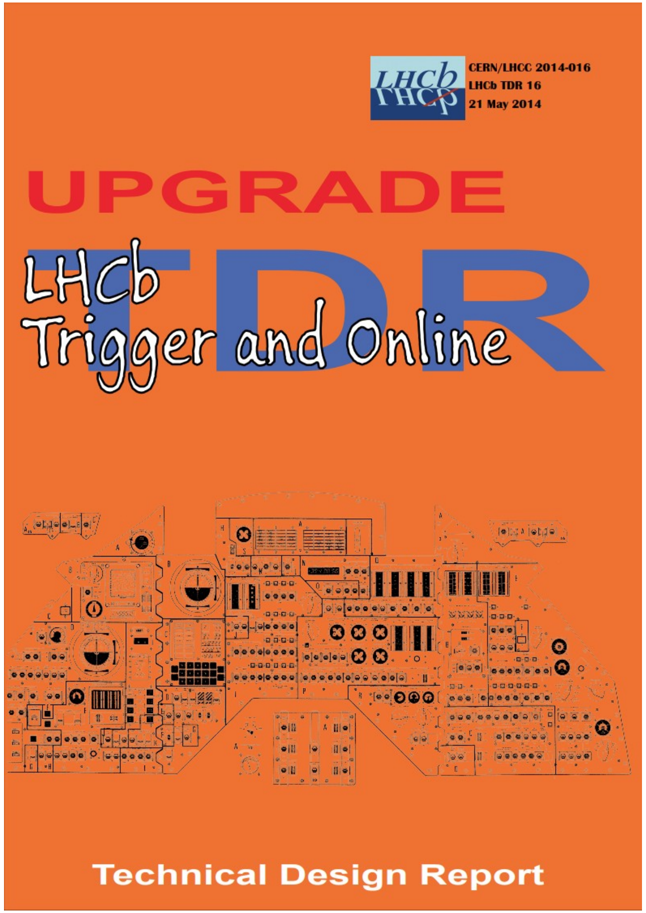
10.1103/PhysRevLett.118.192001

Possible due to flexible persistence in trigger.



$$z = \frac{p_T(J/\Psi)}{p_T(\text{jet})}$$

Present as planned in 2014

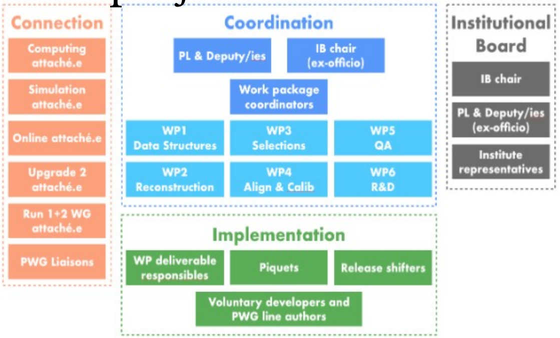


- Removal of L0 trigger bottleneck to increase efficiency.
- Full software trigger for maximum flexibility.
- Split of HLT1 and HLT2 and real-time alignment.
- Output bandwidth 2 to 10 GB/s (2 being bad for efficiency)

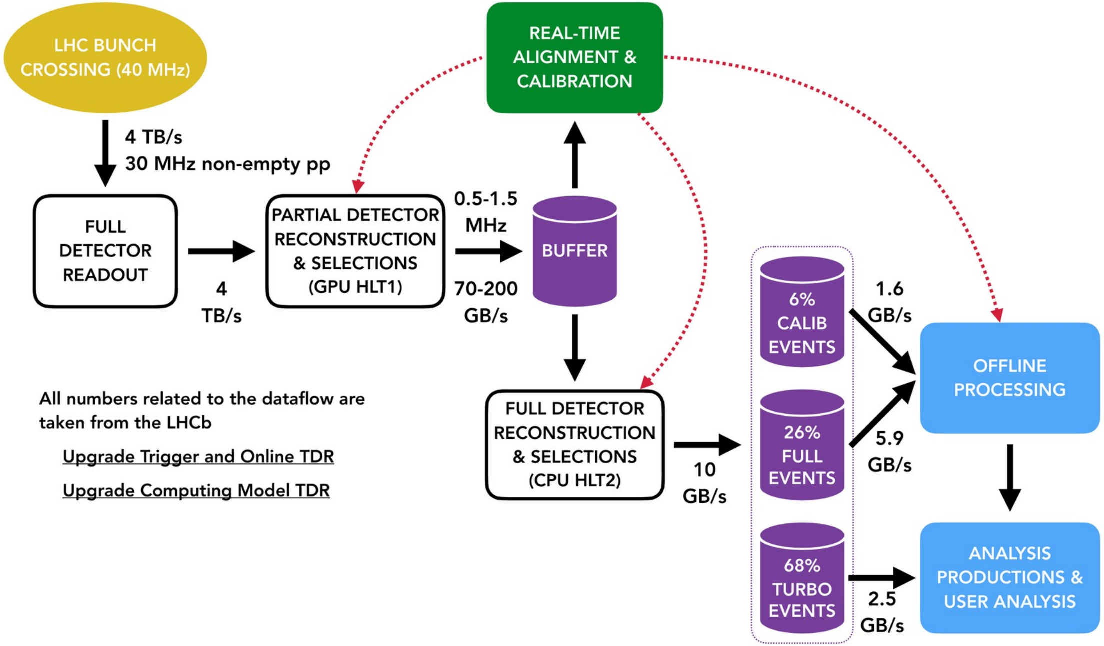
Notable omissions (or only mentioned as ideas):

- GPUs in Event builder running HLT1.
- Full reliance on Turbo model to increase efficiency while keeping same output bandwidth.

RTA project



Present and near future



All numbers related to the dataflow are taken from the LHCb

Upgrade Trigger and Online TDR

Upgrade Computing Model TDR

Numbers will look different in 1 or 2 years, definitely in Run 4.

- Hard work by many people to go from TDR to current trigger system.
- Took first data with HLT1 and HLT2 in 2022 and ran alignments and calibrations.
- Still a long way to go until the new detector and the new trigger system are understood.
- Possibilities and challenges of
  - No L0 ⟶ trigger almost fully track based.
  - HLT1 on GPUs.
  - Flexibility of HLT2 persistence.

Future

<u>How to handle increased luminosity?</u>

The baseline:

- HLT1 performs a fast (HLT1) reconstruction to identify inclusive beauty- and charm signatures as well as high pT muons at LHC frequency
- HLT2 performs full reconstruction allowing to discriminate between signals of interest
- Disk buffer in between to increase compute power and allow for alignment and calibration

The evolution:

- HLT1 output rate increased by factor 7.5
- HLT1 <u>and</u> HLT2 accelerated by GPUs to handle bigger and more events

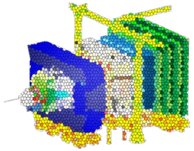
Future

Assumptions	Value
Run 3 event size (kB)	100
HLT1 input rate (Hz)	30×10^6
Run 3 HLT1 output rate (Hz)	1×10^6
Luminosity scaling factor	7.5
Event size scaling factor	4
HLT2 out-of-fill processing factor	0.5
Years until Upgrade II	10
Luminosity HLT1 output rate scaling factor	7.5
Physics HLT1 output rate scaling factor	1
Detector complexity scaling factor	1.0
Disk buffer size scaling factor	0.5
Run 3 disk buffer size (kB)	28×10^{12}
CPU server (2x32 EPYC)	6 600 USD
Consumer GPU	825 USD
PCIe slot per GPU	700 USD
Cost-performance evolution per year	1.1

Item	Cost [kCHF]
HLT1	620
HLT2	9 060
Disk buffer	7 720
Total	17 400

- Look at some of the assumptions and see what trials and errors are needed to finish the evolution.

Luminosity scaling of throughput

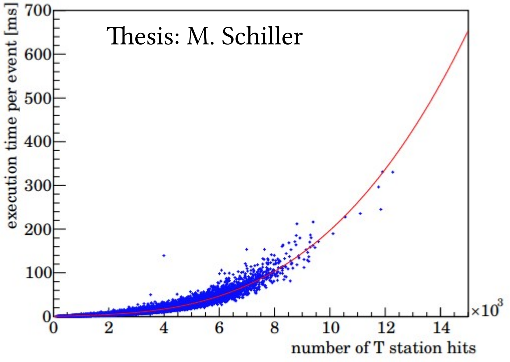


Luminosity scale factor (7.5): assume the throughput scales linearly with luminosity for all algorithms. (FTDR)

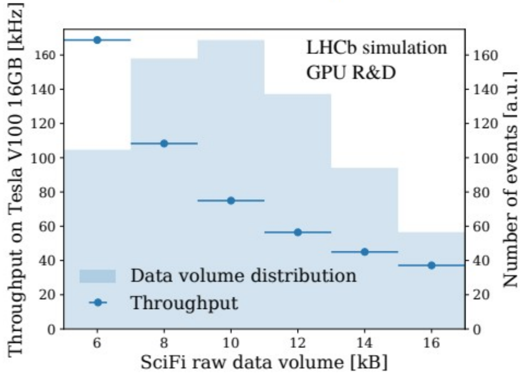
- Basically no reconstruction algorithms scales linearly with detector occupancy (see also Daniel's talk)
- How does Upgrade 1 work?
  - Improved detector technologies, e.g. pixel versus strip detector in Velo, no drift detector (OT) with ambiguities.
  - Scaling still not linear but pre-factors are completely different.

→ Study and optimize complexity of reconstruction for different detector options before making choice (to first order independent of compute hardware).

→ Needs collaboration among hardware and software projects



(a) *PatSeeding*

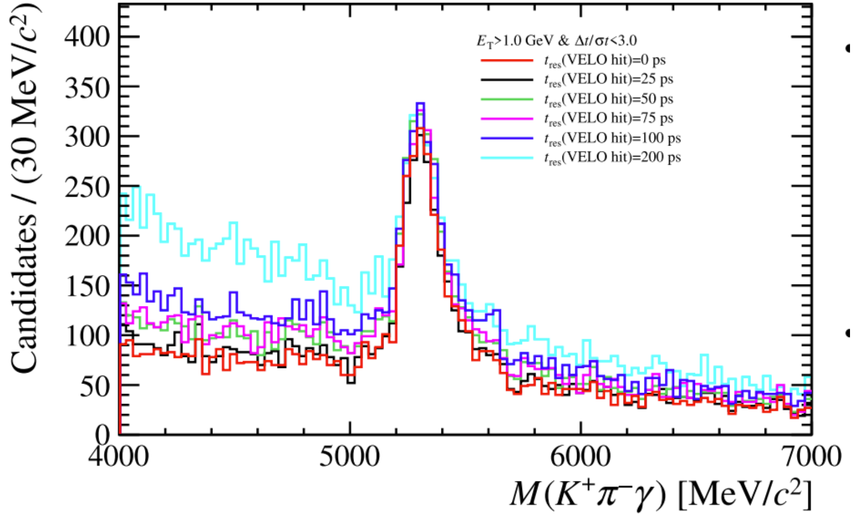


https://arxiv.org/abs/1912.09161

Timing information

> Timing information allows for a particularly fast separation of reconstructed objects according to the pp interaction that produced them, and the availability of precise timing information in multiple subdetectors (, as discussed above,) is therefore crucial. (FTDR)

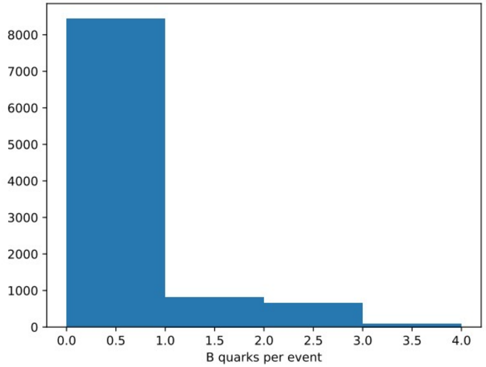
- While timing might not help everywhere to speed up things by factors (e.g. track reconstruction in the Velo), the impact on the full event reconstruction and on particle combinatorics needs to studied.
- See Tim’s talk “Time-resolution optimisation: a joint VELO & ECAL simulation”
  - Particle to PV association is much better (only possible) with timing in the Velo.
  - Matching of Calo clusters with secondary vertices based on time information significantly reduces backgrounds.

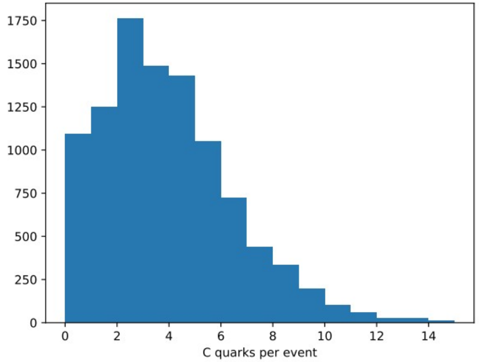


Luminosity scaling of HLT1 output rate

Luminosity HLT1 output rate scaling factor (7.5): from luminosity evolution, assuming that HLT1 output is signal dominated so the HLT1 output rate for a given physics programme scales linearly with luminosity. (FTDR)

- Swamped by signal during Upgrade 2
- Reminder events are also more complex, HLT2 processing will be hit quadratically.
- To be studied, what selections are necessary to reduce HLT1 output rate without loss in efficiency.
  - What about combinatorial background?
  - What about ghost rates in tracking system?
  - Exclusive selections in HLT1 needed?

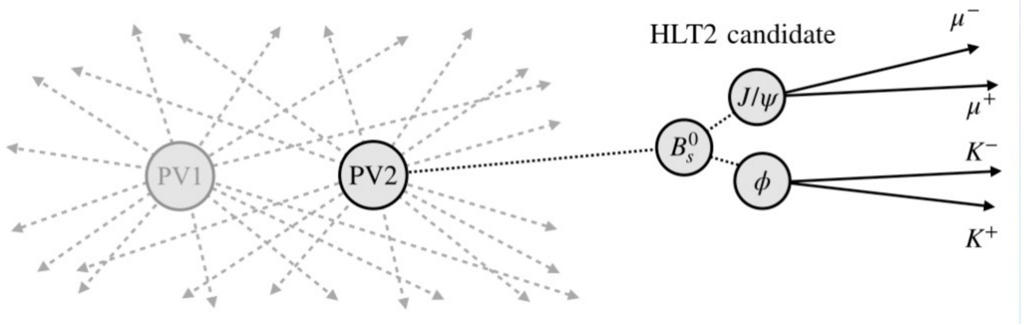




Category	In 4π	In $2 < \eta < 5$
b-hadrons	1.010 ± 0.010	0.243 ± 0.005
c-hadrons	13.959 ± 0.037	3.466 ± 0.019

Inclusive event selection

- Topological trigger essential to collect plethora of B-decays currently relies on persisting the full event reconstruction.
  - 50% of trigger output bandwidth
  - If we keep this model, output bandwidth scales quadratically for same efficiency
- Can we keep inclusive triggers but keep event size under control?
  - Possible if we could reliably isolate the information of individual collisions.
  - People are studying this approach could be tested in Run 3 + 4.
- Or do we have to go "back" to only exclusive selections?



Processing hardware (baseline)

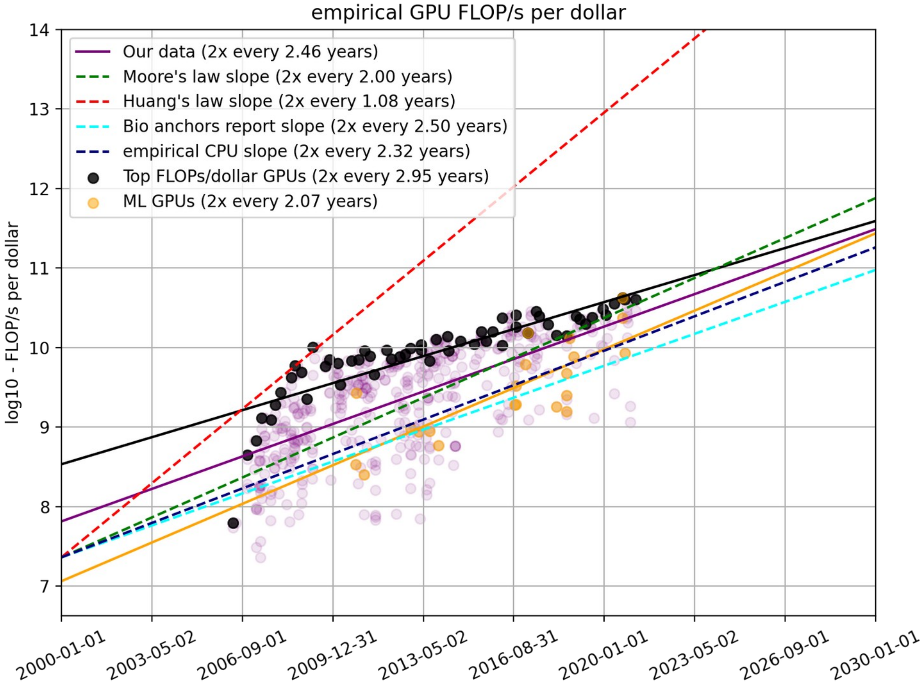
HLT1 and HLT2 processing are both performed on consumer grade GPUs. (FTDR)

- Seems like a no brainer (already for Run 4?)
  $\longrightarrow$ Requires a lot of R&D to see what is possible in a reasonable cost envelope (Daniel's talk)
  - Especially, particle combinatorics and persistence need R&D to profit from acceleration
- HLT1 and HLT2 are currently implemented in different frameworks (Allen and Moore).
  - Have common parts but not yet trivial to move algorithms between them.
  - Few people really know both.
  - Will combining both frameworks reduce maintenance costs? (Marco's Talk)

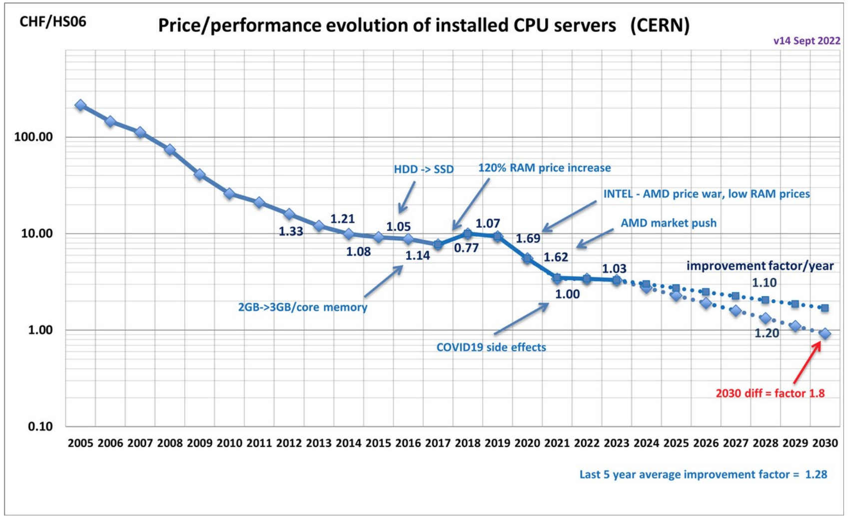
A disk buffer is implemented between HLT1 and HLT2. Therefore, HLT2 can continue to process data out-of-fill, increasing the utility of the farm resources. (FTDR)

- Depends on the costs of storage and compute hardware if that is the better solution.

Cost-performance evolution



- Use scaling factors discussed on previous slides together with Run 3 throughput to estimate needed computing power.
- Then use factor for cost-performance improvements per year to estimate costs (1.1 in FTDR)



Funding model of RTA project for Upgrade 2

- Processing hardware paid from Common Fund.
  → No MOUs binding institutes to the project
- Implemented a RTA Collaboration Agreement between institutes and the RTA project.
  → Soft agreement does not guarantee that enough personnel or that the required expertise will be available to develop efficient software satisfying high quality standard.
- Consequences can be felt every day
  - Critical parts of the software maintained on best effort basis, particle combinatorics, functors, persistence, MC association.
  - Few people giving support to ever growing user base.
- Improved model?
  - Make personnel costs more obvious, create possibilities that project can hire experts.

Conclusion

- (Software) trigger place of innovation when hardware fixed
  - During Run 1 and 2 but also during construction of Upgrade 1.
  - Probably haven't understood potential of full software trigger yet.
- Still, important to design trigger and sub-detectors coherently before hardware fixed.
  - Common/global optimisation needed

- Many open questions for Run 4 and 5.