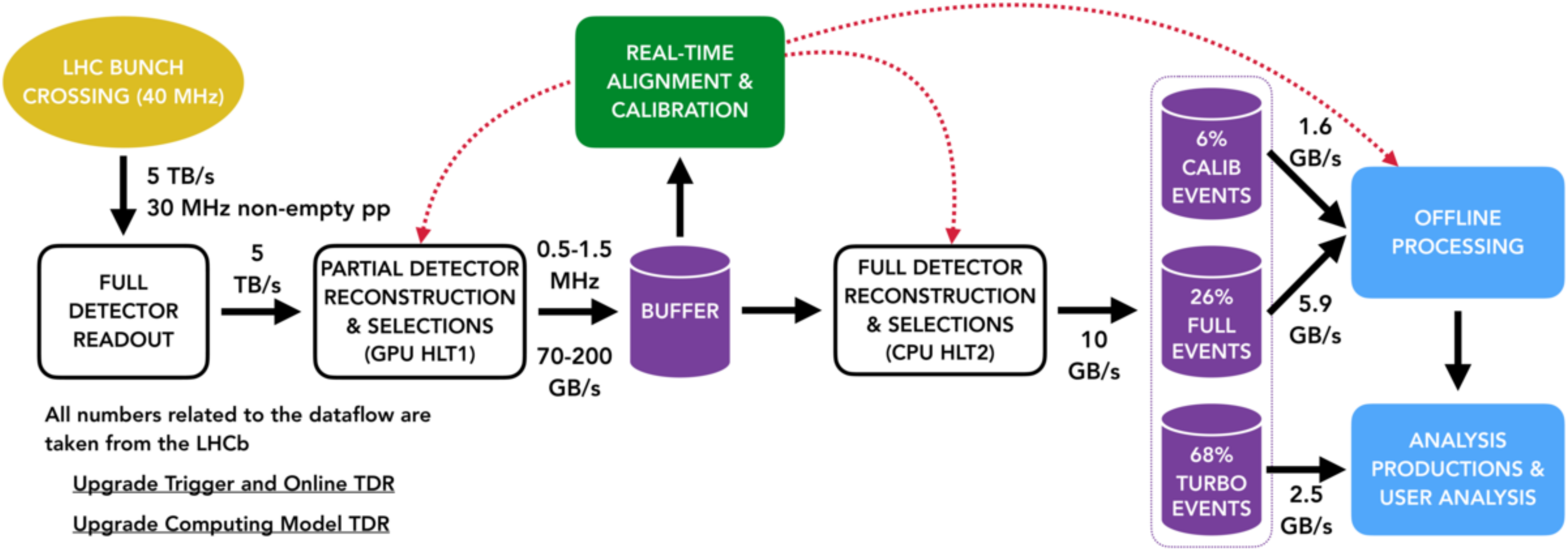
Software e computing in LHCb: la sfida di Run3 (e oltre)

Concezio Bozzi
INFN Sezione di Ferrara
CNAF, 23 Febbraio 2021

Outline

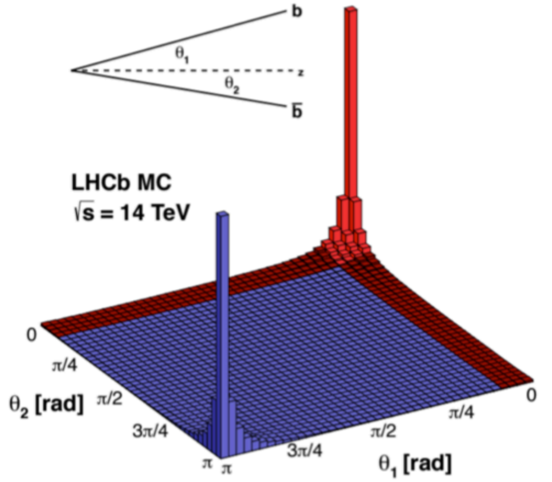


All numbers related to the dataflow are taken from the LHCb

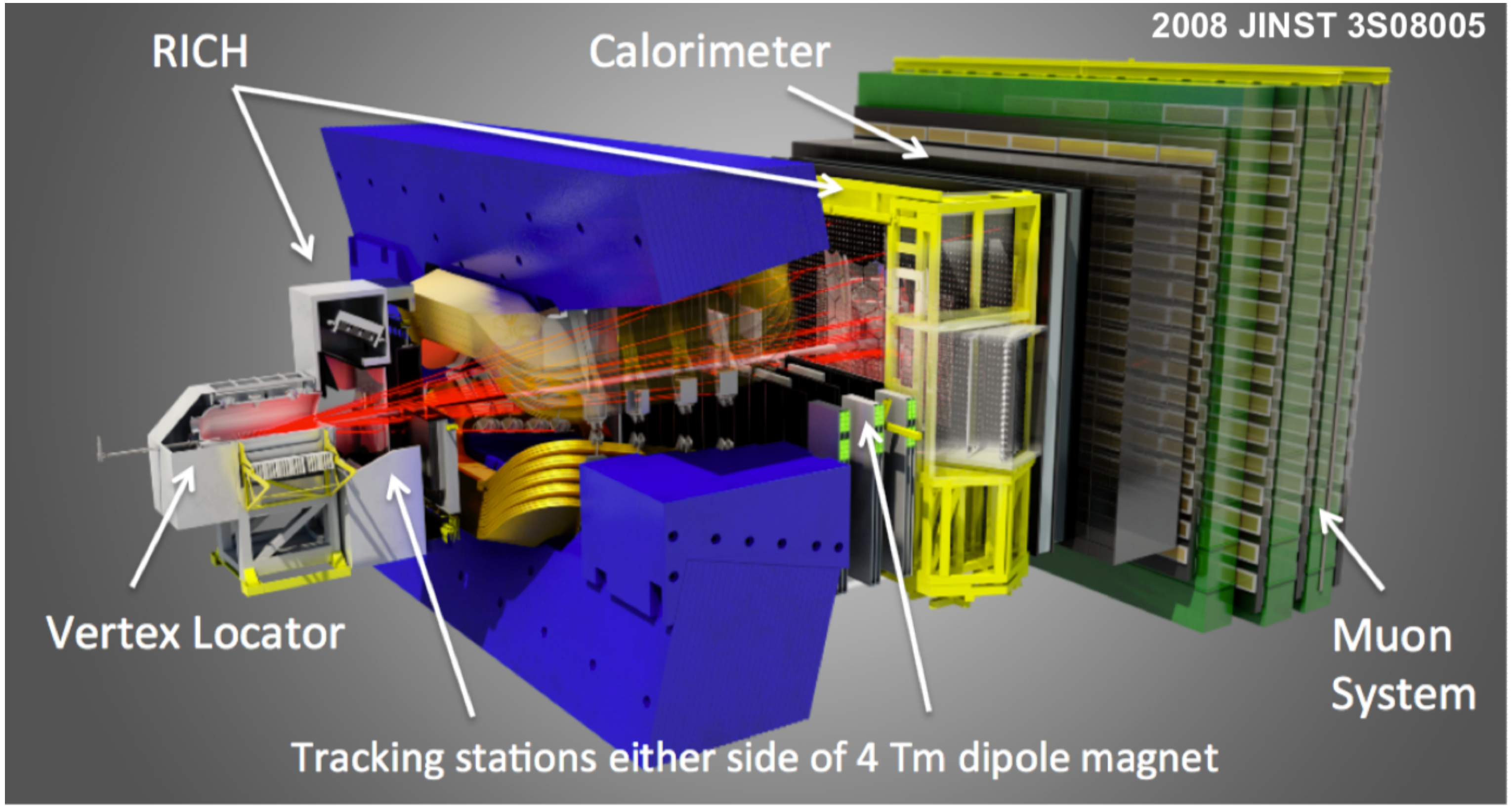
Upgrade Trigger and Online TDR

Upgrade Computing Model TDR

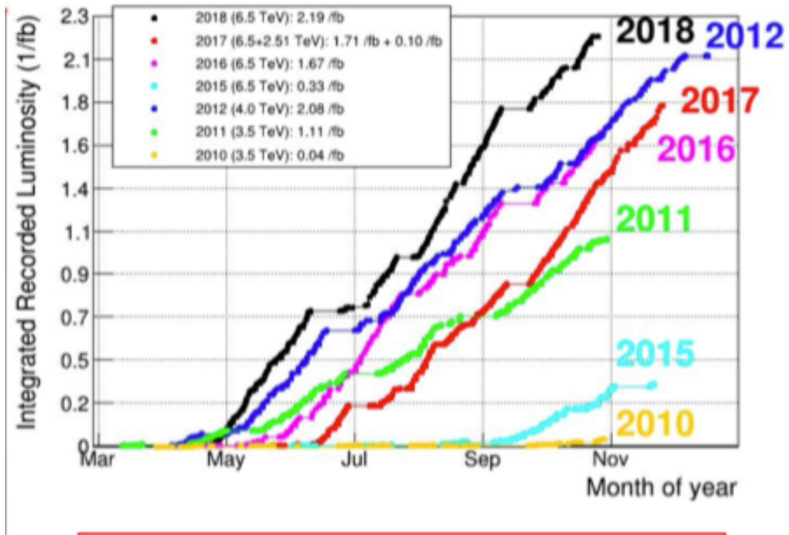
The LHCb experiment



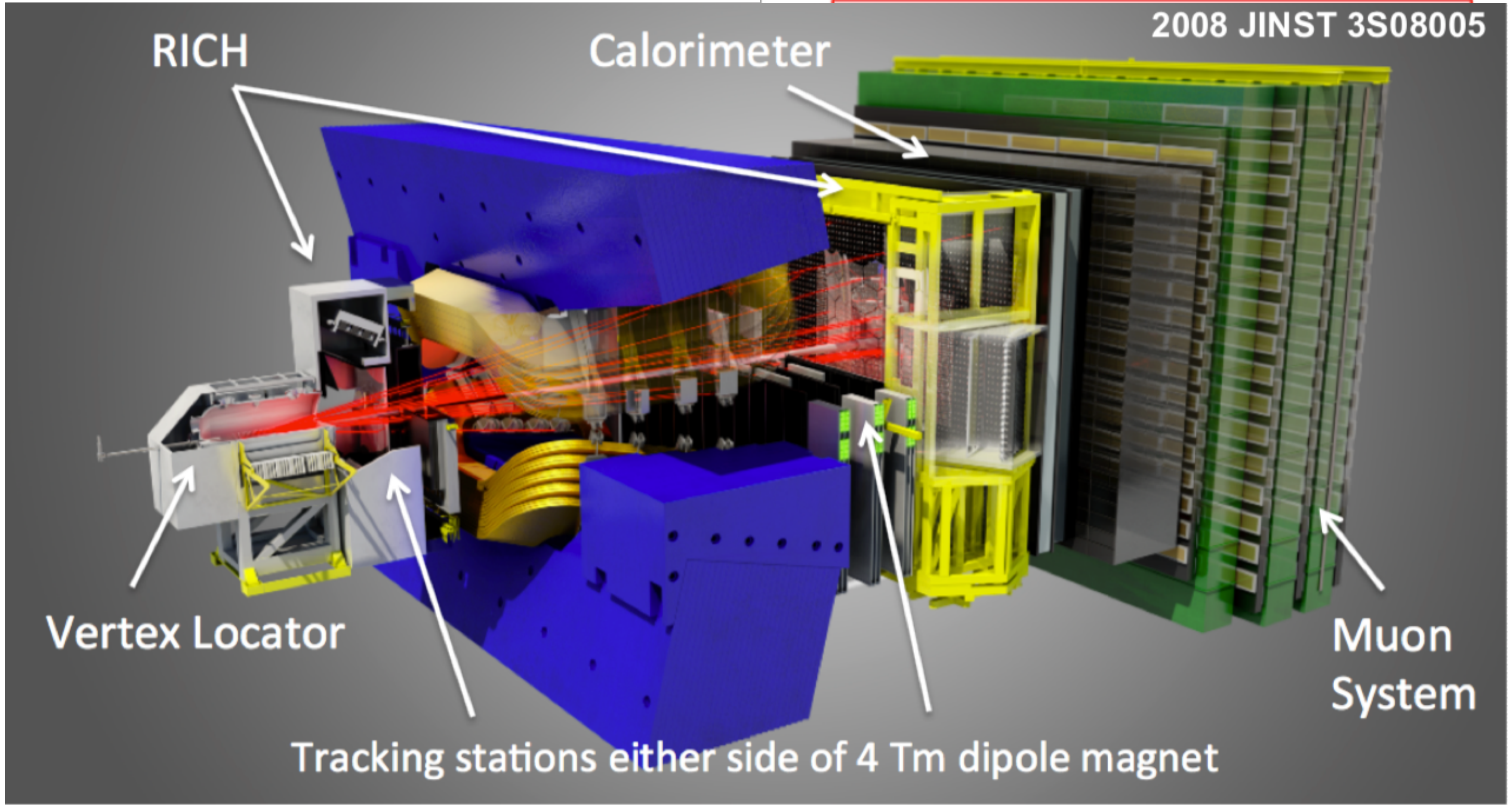
- LHCb was designed to study CP-violation and search for New Physics phenomena in heavy flavour (beauty and charm) quark sector
- Single-arm spectrometer, fully instrumented in pseudo rapidity range $2 < \eta < 5$
  - solid angle coverage ~ 4%, 40% B hadrons
- Thanks to its excellent performance, the LHCb detector also gave crucial insights and world-class measurements in other sectors e.g.
  - CP violation in charm
  - Hadron spectroscopy (tetraquarks, pentaquarks…)
  - Electroweak physics
  - Cross-section measurements in fixed-target mode
  - Heavy-ion physics



The LHCb experiment



- Many of LHCb results obtained in Run1 and Run2 are dominated by statistical uncertainties
- An upgrade of LHCb has therefore been planned and it is currently underway to take data in Run3 and beyond



The LHCb upgrade in a nutshell

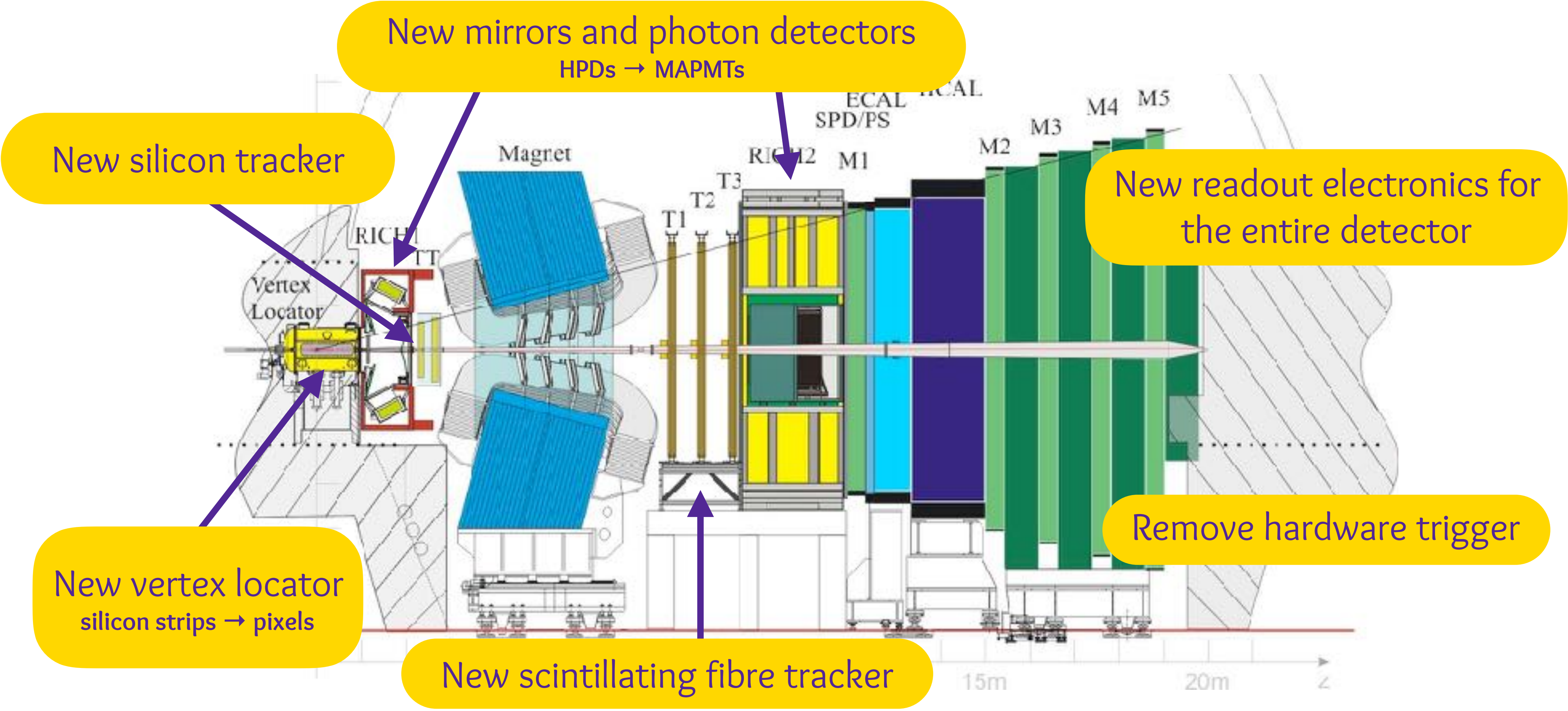
- An LHCb Upgraded detector is being installed in 2019-2021 (LHC LS2) and it will take data in Run 3 (2022-2024) and beyond.
- The motivation is to boost the physics output by taking advantage of the huge rate of heavy-flavour production at the LHC. This will be achieved by
  - Raising the instantaneous luminosity by a factor five to $2 \times 10^{33} cm^{-2}s^{-1}$
    - Number of visible interactions x5 larger
  - Implementing a full software trigger
    - to overcome the limitations of L0 hardware trigger

- Huge increase in precision, in many cases to the theoretical limit, and the ability to perform studies beyond the reach of the current detector.
- Flexible trigger and unique acceptance also opens up opportunities in other topics apart from flavour (‘a general purpose detector in the forward region’)

- Necessary to redesign several sub-detectors and their readout

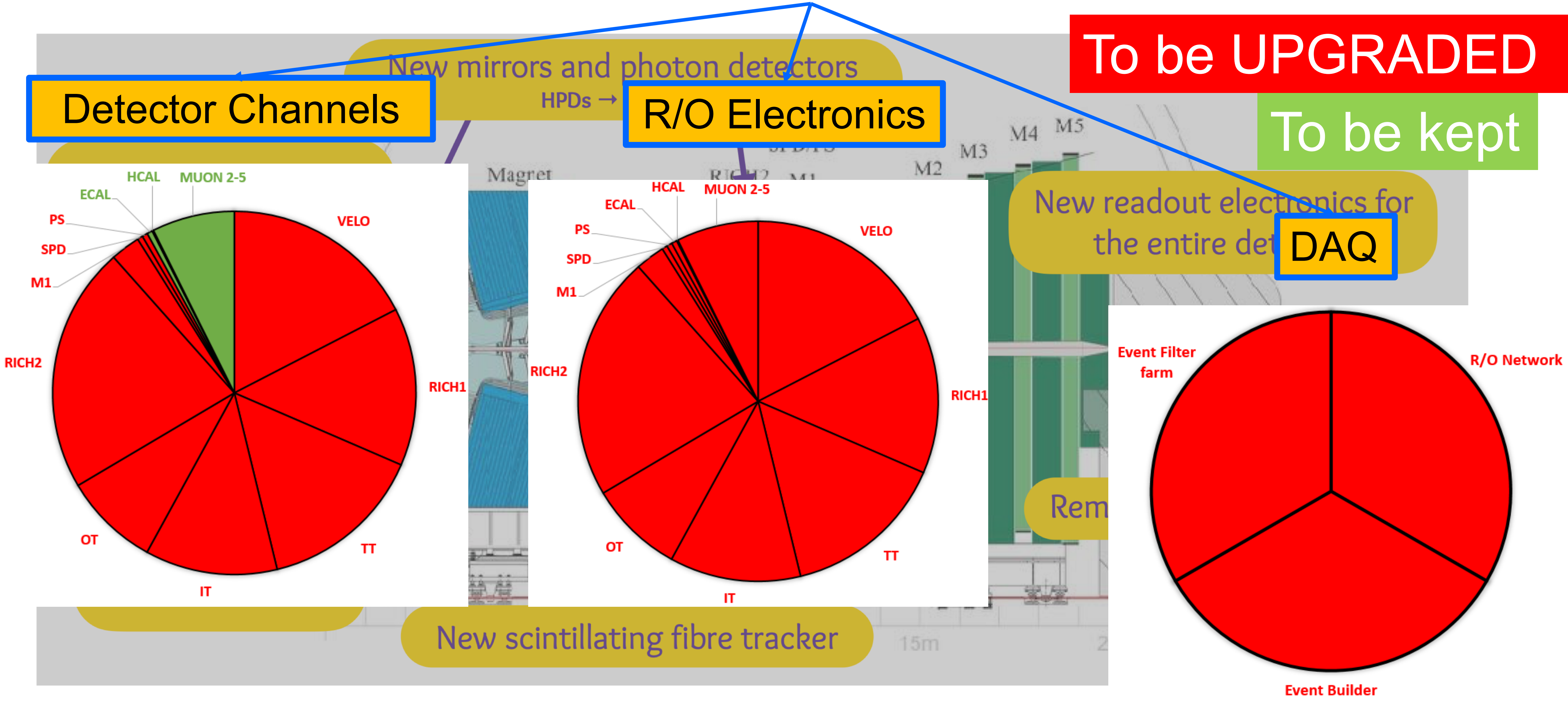
CERN-LHCC-2012-007

The upgraded LHCb detector for Run 3



CERN-LHCC-2012-007

The upgraded LHCb detector for Run 3



A big challenge in data handling

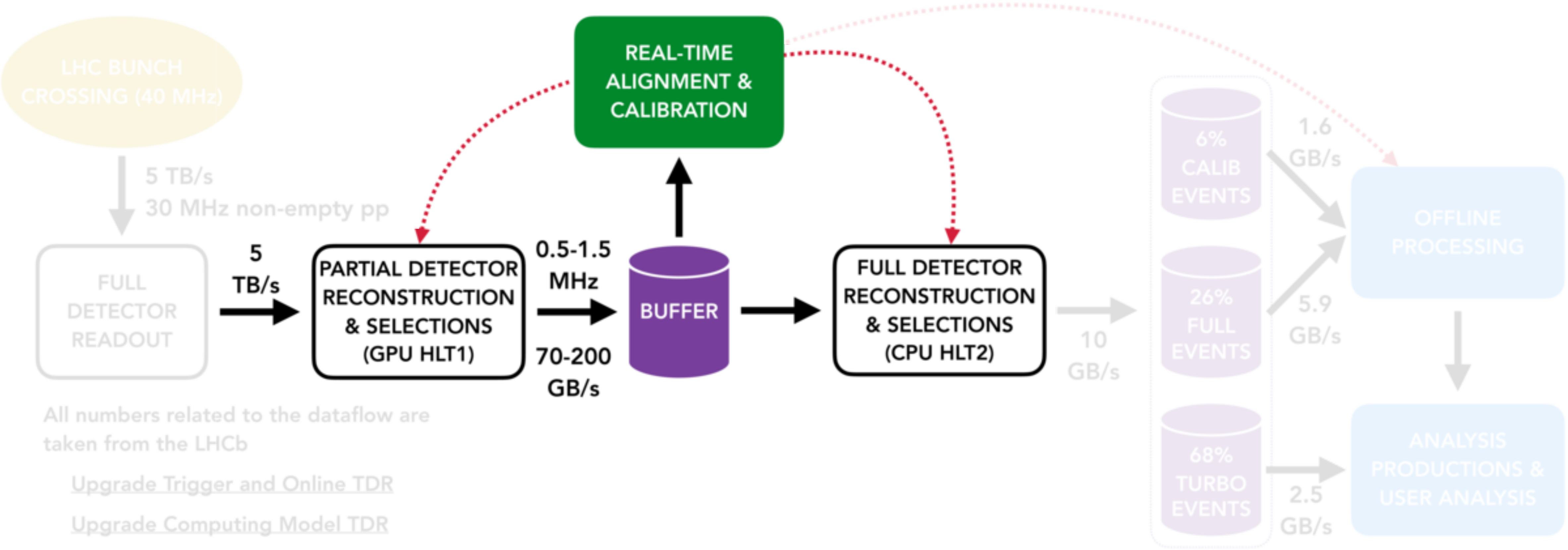
- Major expansion of LHCb physics programme through:
  - 5-fold increase in instantaneous luminosity
    - $4x10^{32}$ to $2x10^{33}$ $cm^{-2}s^{-1}$
  - Full software trigger at 30MHz inelastic collision rate
    - Factor 2 increase in trigger selection efficiency
- Order of magnitude increase in physics event rate to storage

- Pile-up increase
    - Factor 3 increase in average event size
- 30x increase in throughput from the upgraded detector
  - Without corresponding jump in offline computing resources

A big challenge in data handling

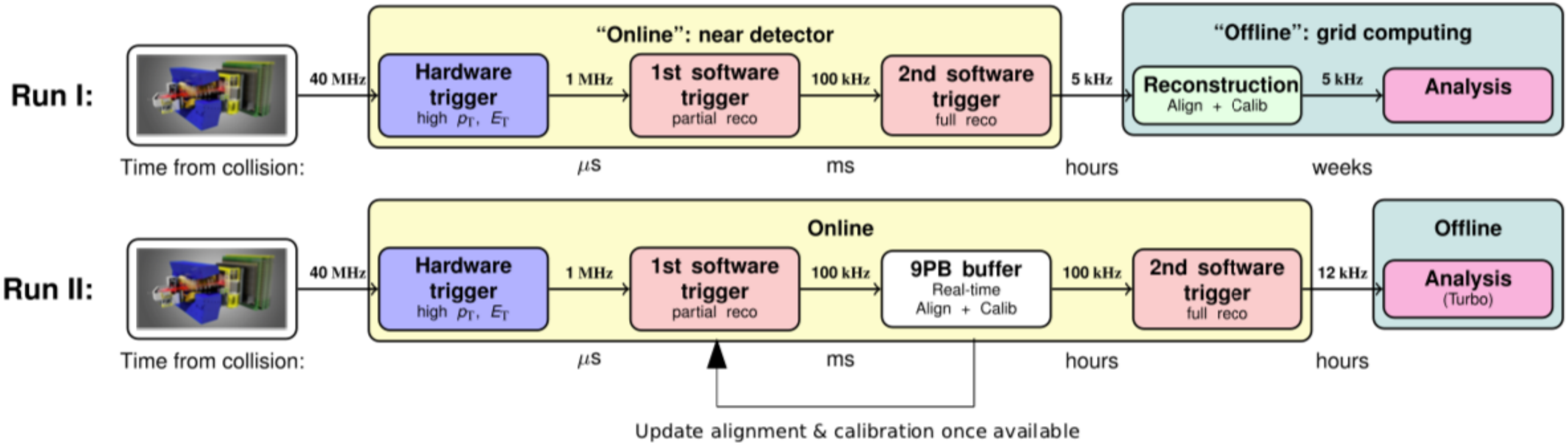
- Major expansion of LHCb physics programme through:
  - 5-fold increase in instantaneous luminosity
    - 4×10^{32} to 2×10^{33} $cm^{-2}s^{-1}$
  - Full software trigger at 30MHz inelastic collision rate
    - Factor 2 increase in trigger selection efficiency
- Order of magnitude increase in physics event rate to storage
- Pile-up increase
    - Factor 3 increase in average event size
- 30x increase in throughput from the upgraded detector
  - Without corresponding jump in offline computing resources

Outline



LHC BUNCH CROSSING (40 MHz)
5 TB/s
30 MHz non-empty pp
FULL DETECTOR READOUT
5 TB/s
PARTIAL DETECTOR RECONSTRUCTION & SELECTIONS (GPU HLT1)
0.5-1.5 MHz
70-200 GB/s
BUFFER
REAL-TIME ALIGNMENT & CALIBRATION
FULL DETECTOR RECONSTRUCTION & SELECTIONS (CPU HLT2)
10 GB/s
6% CALIB EVENTS
26% FULL EVENTS
68% TURBO EVENTS
1.6 GB/s
5.9 GB/s
2.5 GB/s
OFFLINE PROCESSING
ANALYSIS PRODUCTIONS & USER ANALYSIS
All numbers related to the dataflow are taken from the LHCb
Upgrade Trigger and Online TDR
Upgrade Computing Model TDR

Run1 + Run2 trigger



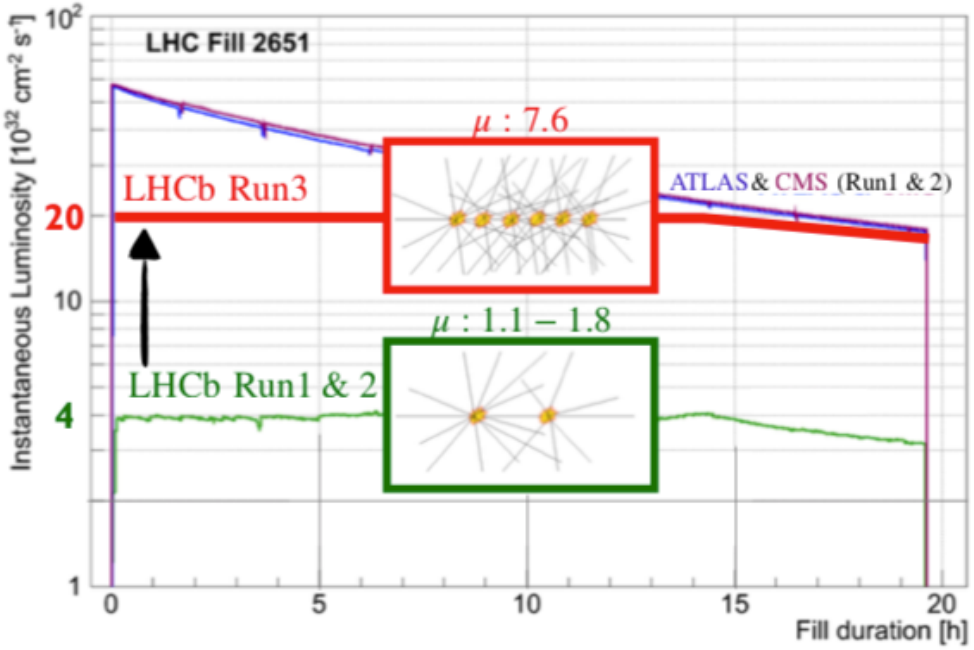
- Hardware trigger: based on muon detectors and calorimeters

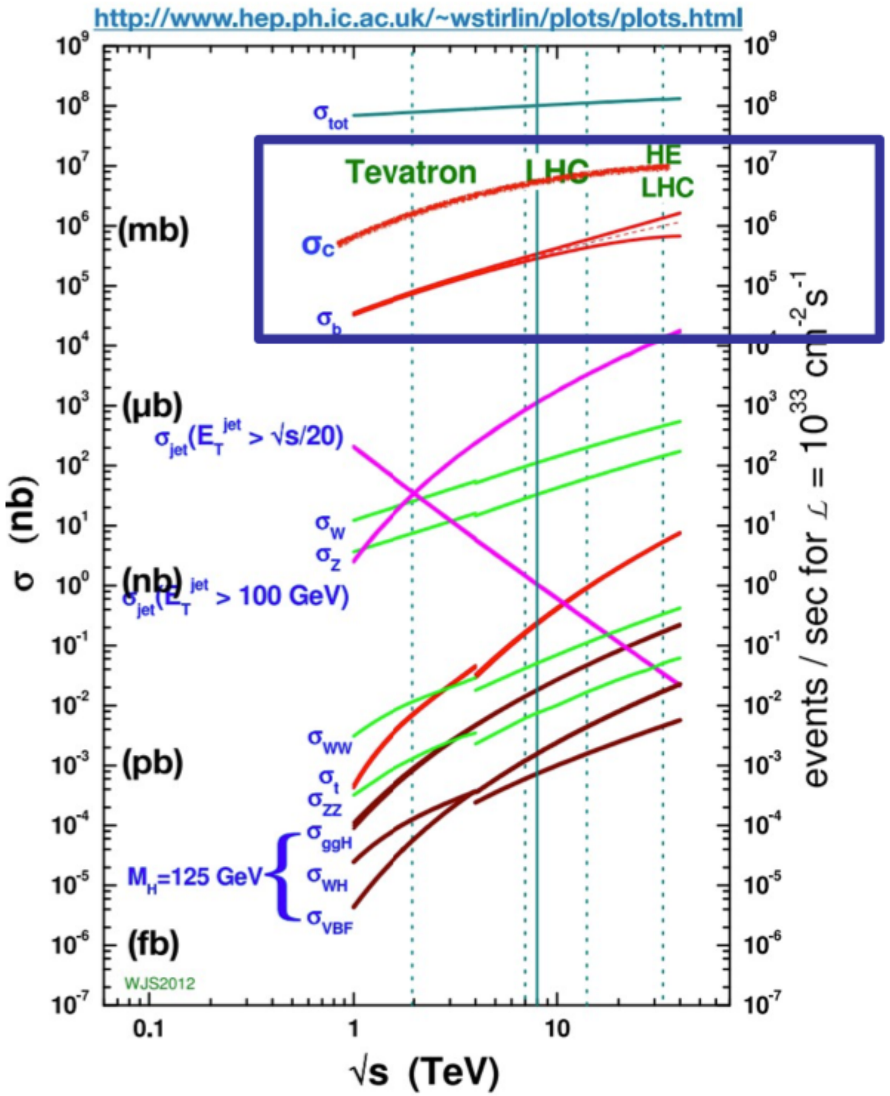
Run 2

- Data buffered in between two software trigger stages
- Allows for real-time alignment and calibration Offline-quality reconstruction within the trigger

Luminosity increase: x5

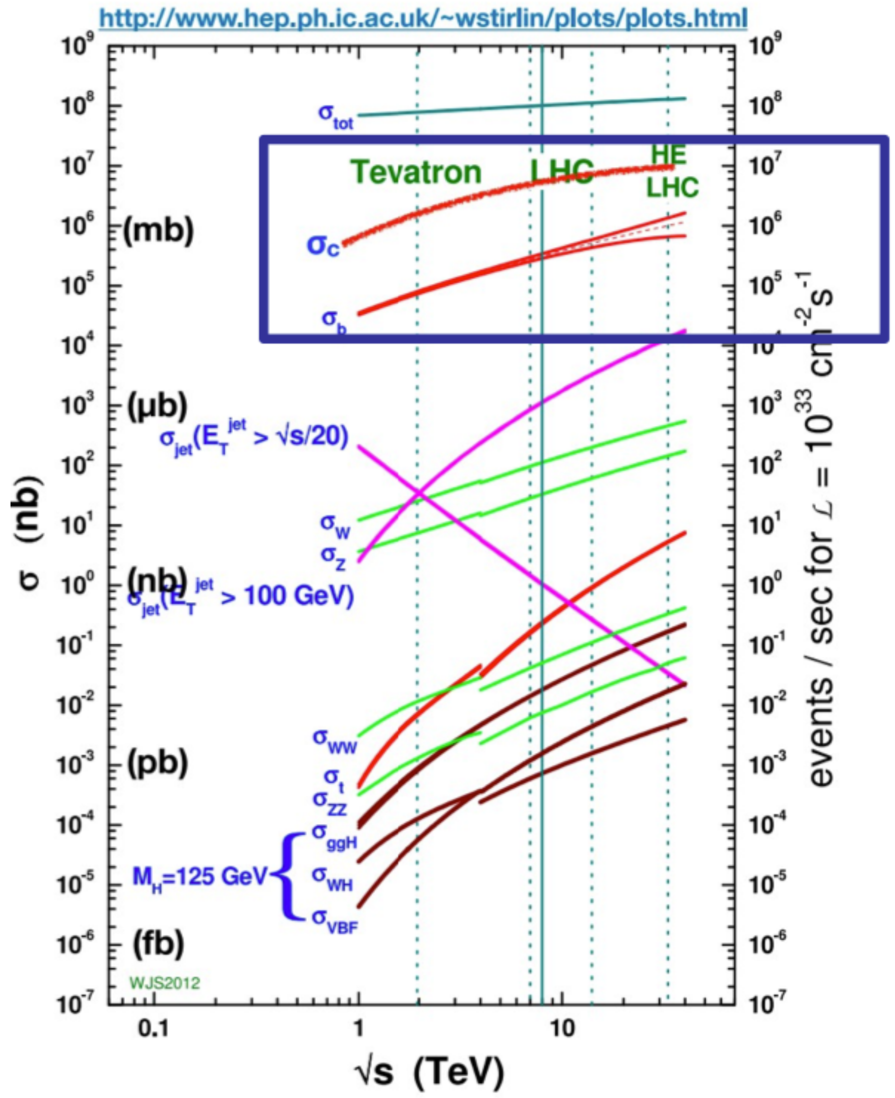
- More interaction vertices per collision of proton bunches, more tracks, more signal
- Beauty and charm signal rates: 1-10MHz
- Almost all events will have a *b* or *c* hadron in Run 3





The MHz signal era

	Signal/ background	Typical signal rates	kinematics	Trigger strategy	Trigger efficiency
GPD (ATLAS /CMS)	Rare events, background dominated	<100kHz	High pT	Local signatures, Reject background Select rare events	Cut at high pT Work at efficiency plateau
LHCb	High cross sections, signal dominated	1-10MHz	Low pT	No "simple" local criteria Classify decays Access as much information about the collision as early as possible Read full detector	Cut at low pT Work at efficiency onset edge

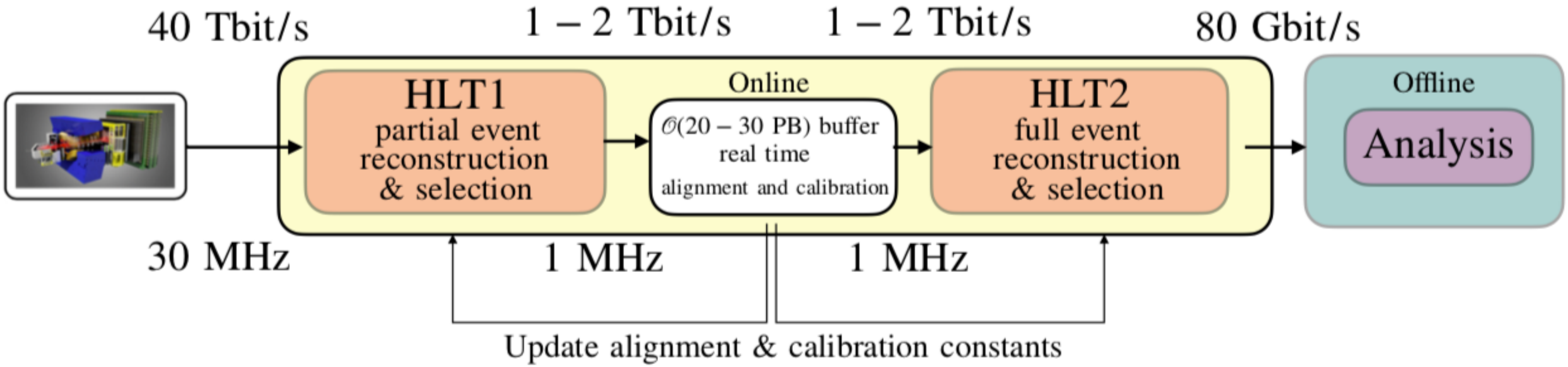


Bottom line: hardware trigger possible at GPDs, not an option for LHCb

The MHz signal era

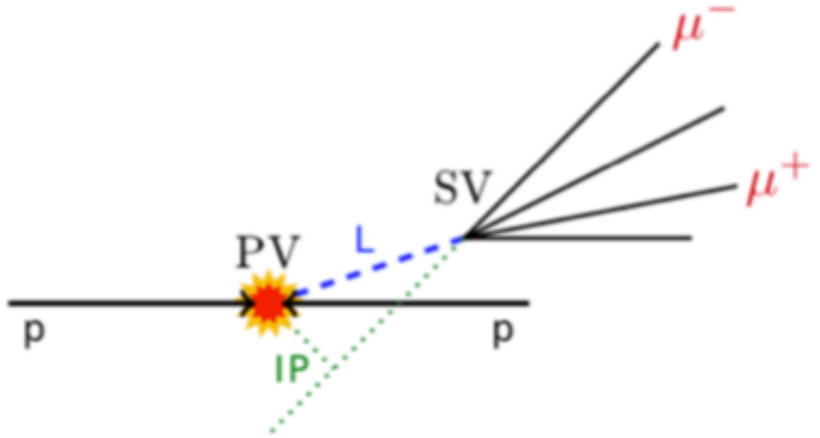
“From a needle in a haystack to an haystack of needles”

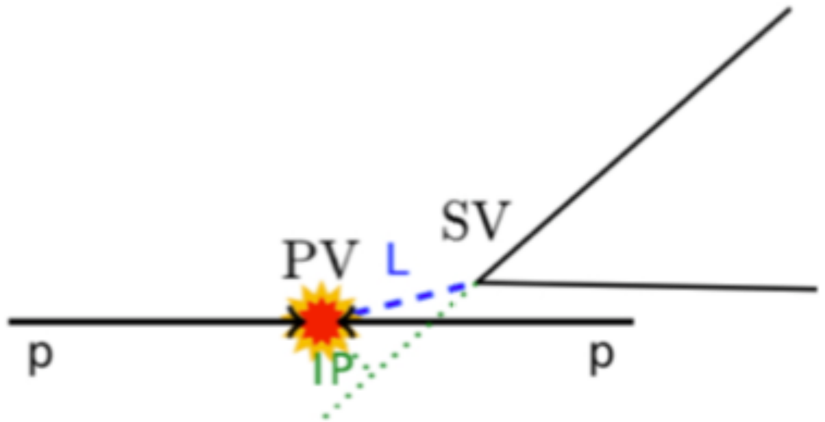
Run 3 trigger



- Remove Hardware trigger in favour of a fully software based one.
- Event reconstruction at collision rate
- Full detector read-out at 40 MHz (visible collision rate: 30MHz)

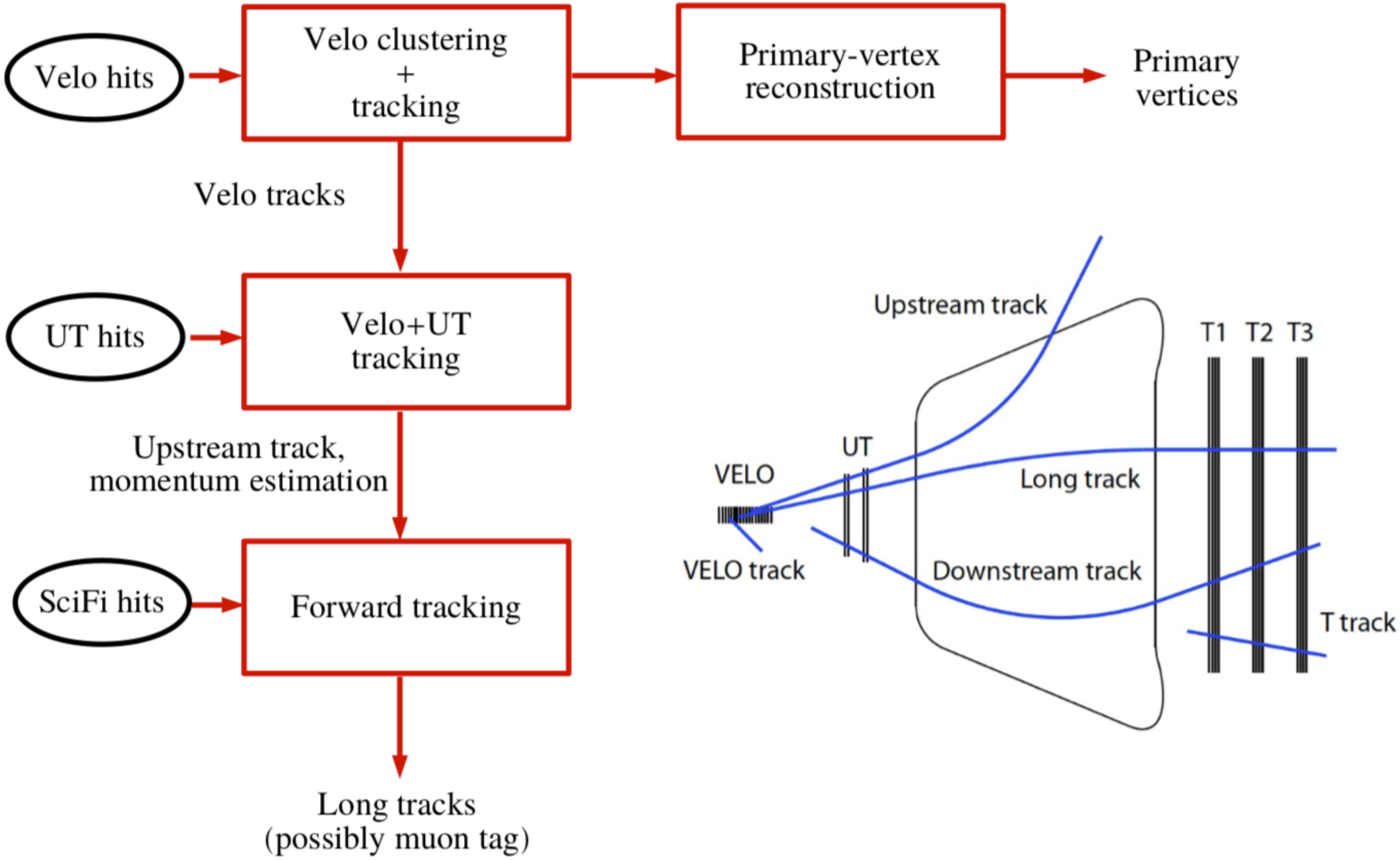
Run 3 conditions





- Key ingredients for efficient triggering and signal discrimination
  - Primary vertex finding, tracks reconstruction and optimal μ-Identification,
  - Inclusive triggers on signatures with 1&2 “displaced” tracks.
  - Challenge in Run3 is not only to have an efficient trigger, but also be able to identify the topology of events as early as possible in the triggering process: more information than single sub-detector read-out needed
  - Track reconstruction at collision rate required: huge computing challenge

The HLT1 reconstruction sequence



Software performance: early nightmares

- LHCb upgrade online TDR advocates for a trigger farm consisting of O(1000) nodes
- Running HLT1 at 30MHz means that a single node must process O(30k) events/second

LHCb-TDR-016

Software performance: early nightmares

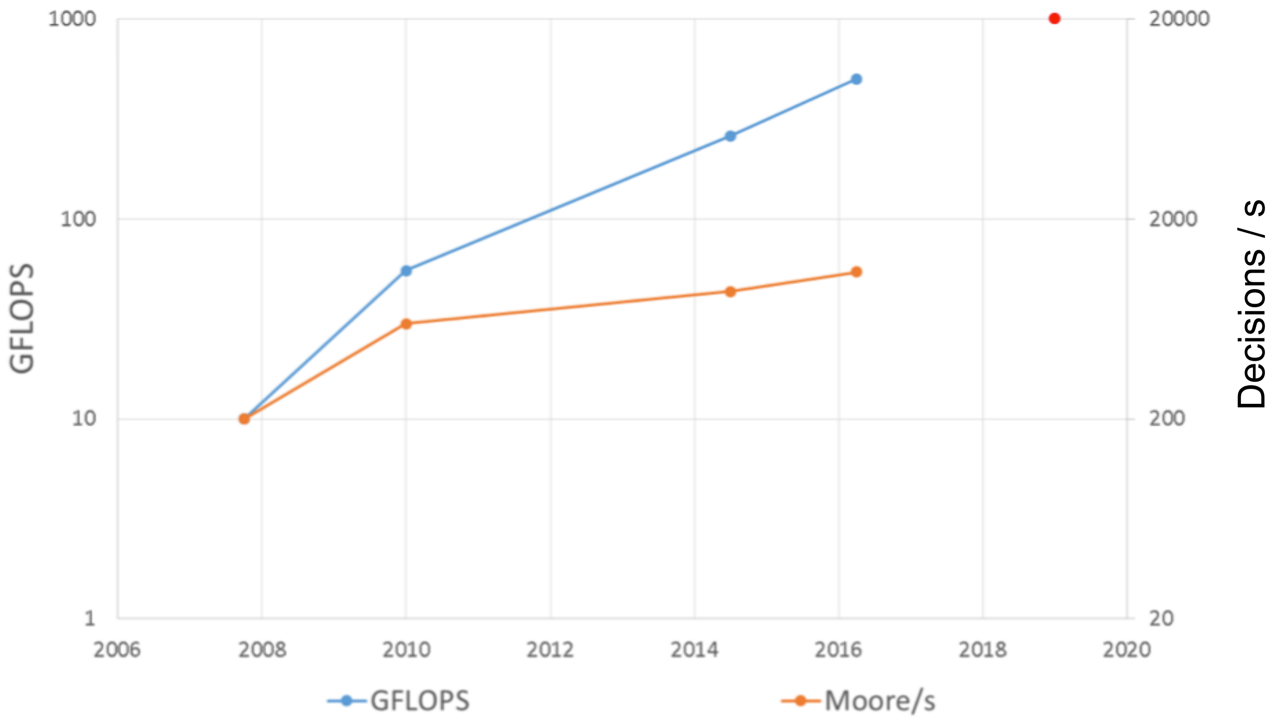
- LHCb upgrade online TDR advocates for a trigger farm consisting of O(1000) nodes
- Running HLT1 at 30MHz means that a single node must process O(30k) events/second
- First exercise (2016)
  - take upgrade MC simulation and run HLT1 on it by using the most powerful farm node (at that time: dual-Xenon E5-2630V4, 2*10 cores)
  - Resulting throughput: 6k evts/ s
  - ☹☹☹

LHCb-TDR-016

Software performance: early nightmares

- LHCb upgrade online TDR advocates for a trigger farm consisting of O(1000) nodes
- Running HLT1 at 30MHz means that a single node must process O(30k) events/second
- First exercise (2016)
  - take upgrade MC simulation and run HLT1 on it by using the most powerful farm node (at that time: dual-Xenon E5-2630V4, 2*10 cores)
  - Resulting throughput: 6k evts/ s
  - ☹☹☹
- Not unexpected though…

Trigger decisions vs. power of trigger farm

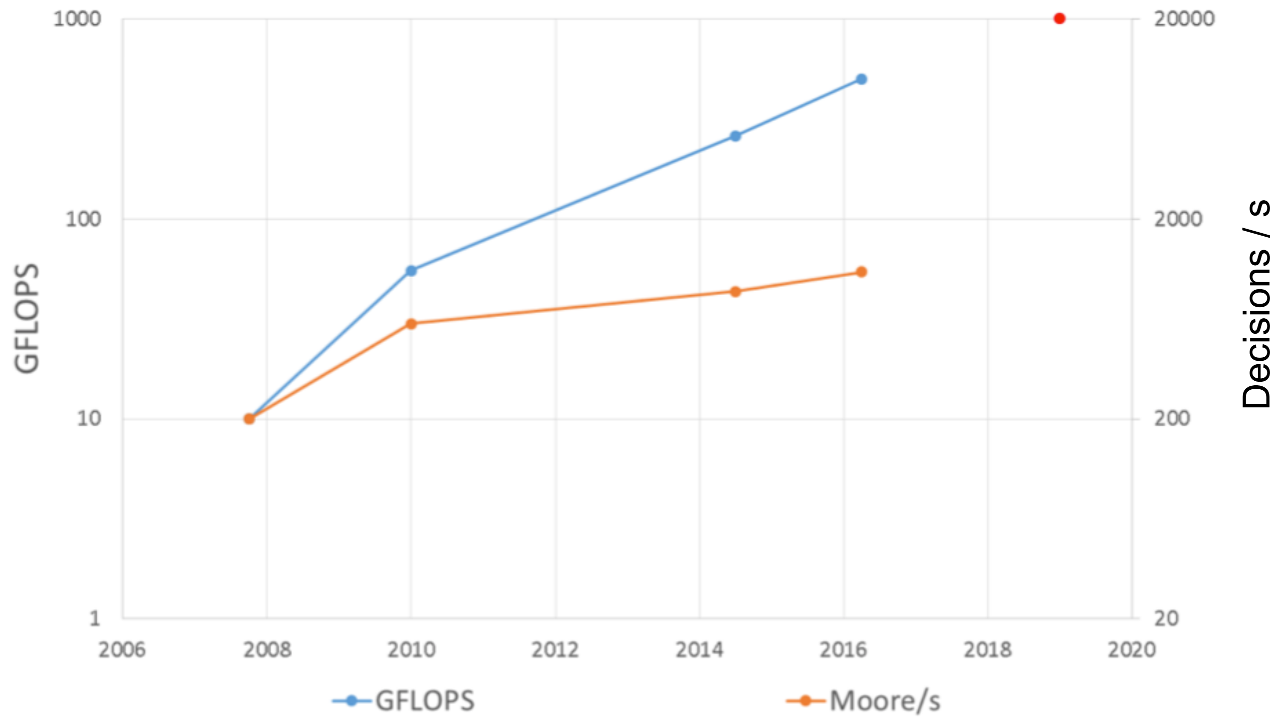


Software performance: somewhat expected

- Run1/2 trigger code single-threaded and scalar
- Evolution trend of faster single- threaded CPU performance broken several years ago.
  - Increase of CPU cores and more execution units.

- Gaudi core framework had been in production without major modifications for 17 years
- Its sequential event data processing model leads to
  - Weak scalability in RAM usage
  - Inefficient disk/network I/O

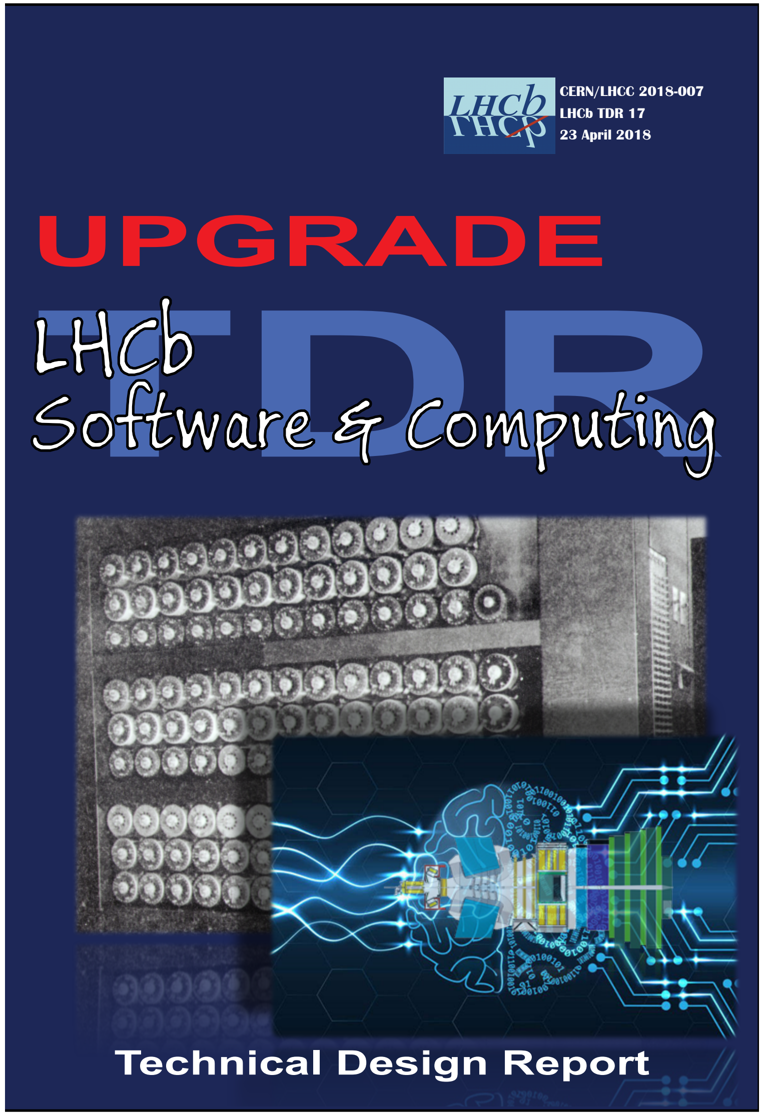
Trigger decisions vs. power of trigger farm



Software performance: much to gain!

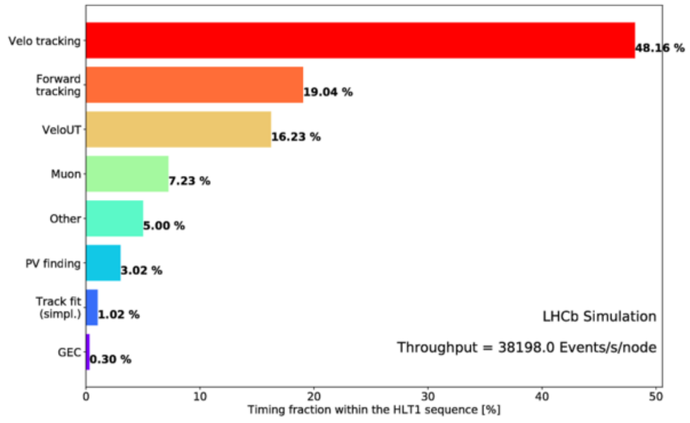
- Modernize Gaudi and make it fit for current and forthcoming challenges

- Several improvements:
  - Better utilization of current multi-processor CPU architectures
  - Enable code vectorization
  - Modernize data structures
  - Reduce memory usage
  - Optimize cache performance
  - Remove dead code
  - Replace outdated technologies
  - Enable algorithmic optimization

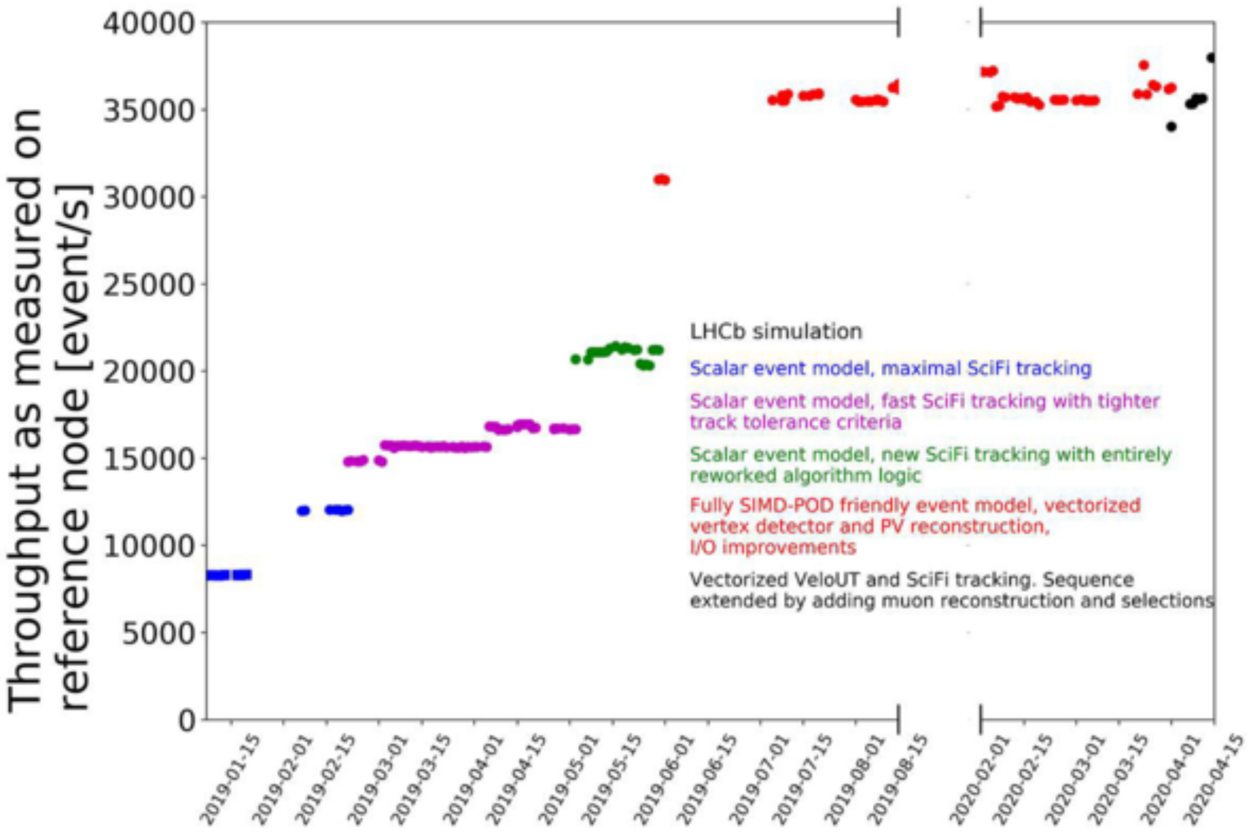


LHCb-TDR-017

HLT1 on CPUs: mission accomplished

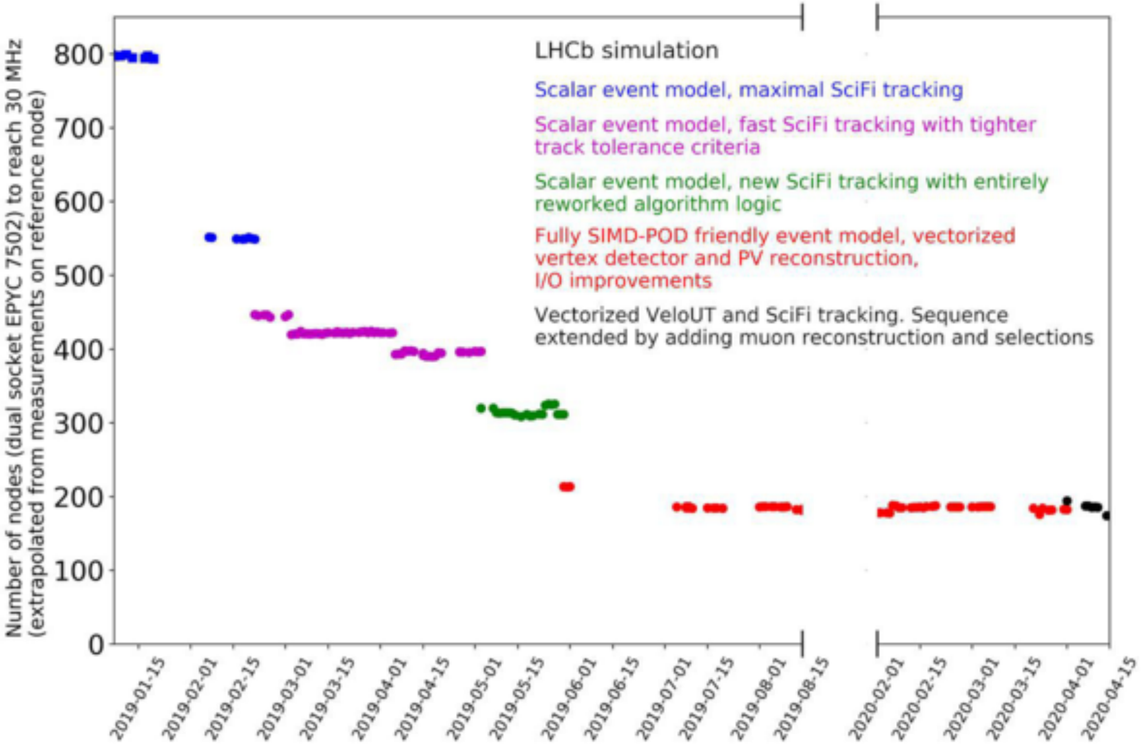


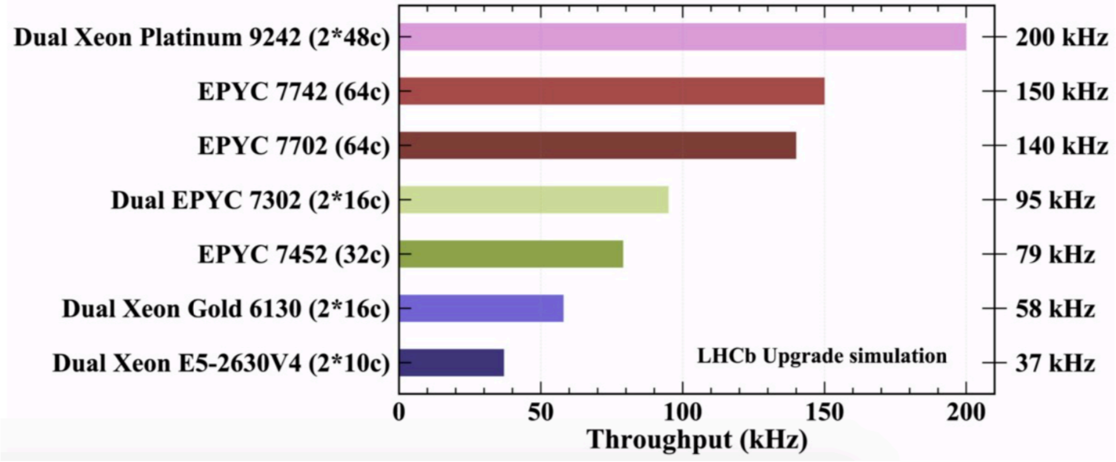
- HLT1 throughput on CPUs has been improved by nearly a factor 5 with no loss on physics performance, surpassing the initial requirement
- This has been made possible by:
  - Rewriting algorithms whose performance used not to be critical (e.g. decodings)
  - Improved use of architecture and intrinsic parallelism, through data model, coding and algorithm design (e.g. velo tracking)
  - Previous experience on operating the current detector, leading to trade-offs and revisited models (e.g. simplified Kalman fit, forward tracking)
- And for most algorithms, all of the above → no "one fits all" procedure

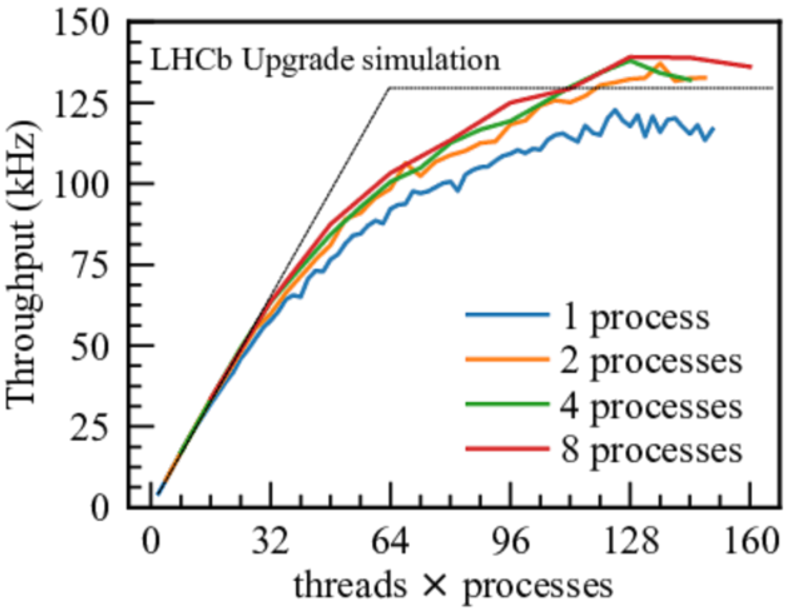


HLT1 on CPUs: mission accomplished

- HLT1 tested on more recent hardware show even better performance
- A full CPU HLT1 would need fewer than 200 EPYC 7502 servers (AMD CPUs)



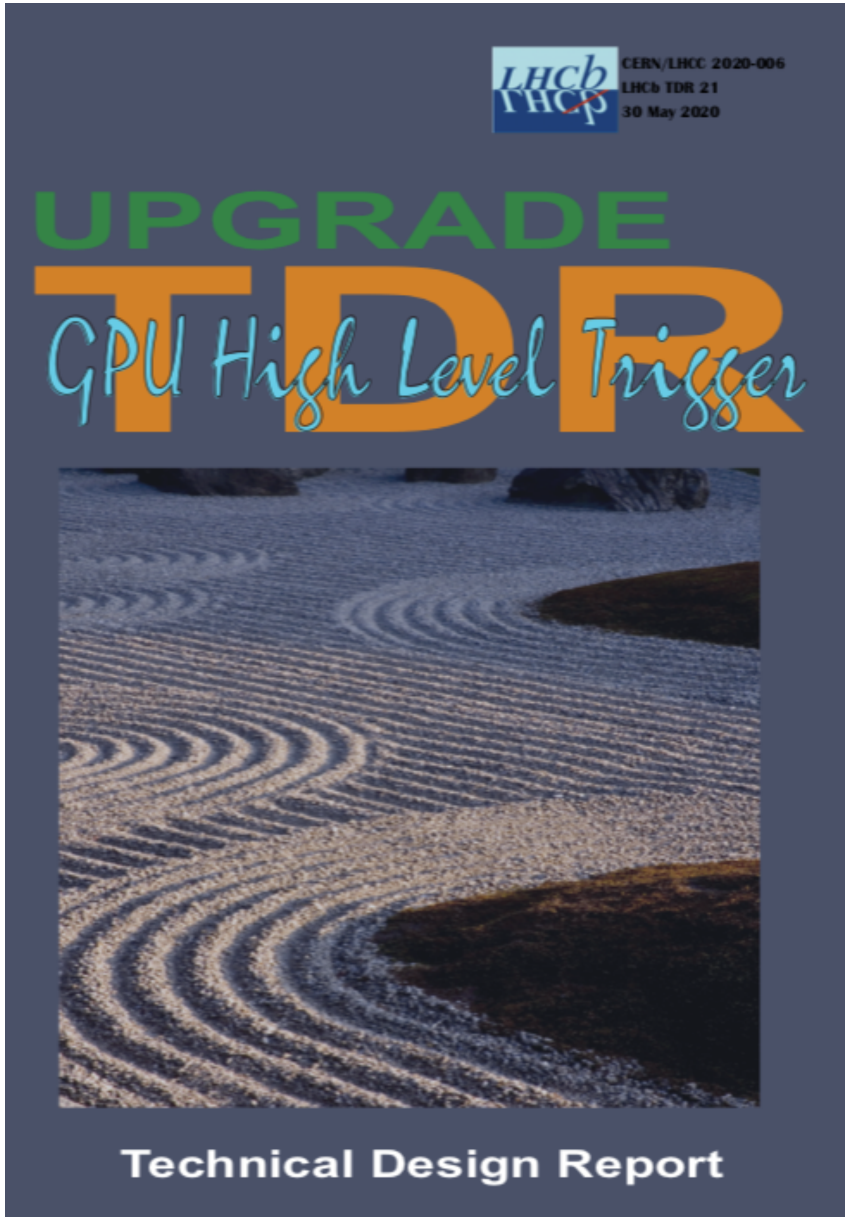




[EPYC 7702]

HLT1 on GPUs ?

- The Allen project began in February 2018 as an R&D project aimed at providing an HLT1 application running on GPUs
- GPUs offer more theoretical FLOPS in a compact package
- Lower cost than CPUs per theoretical FLOPS
- Many HLT1 tasks are inherently parallel



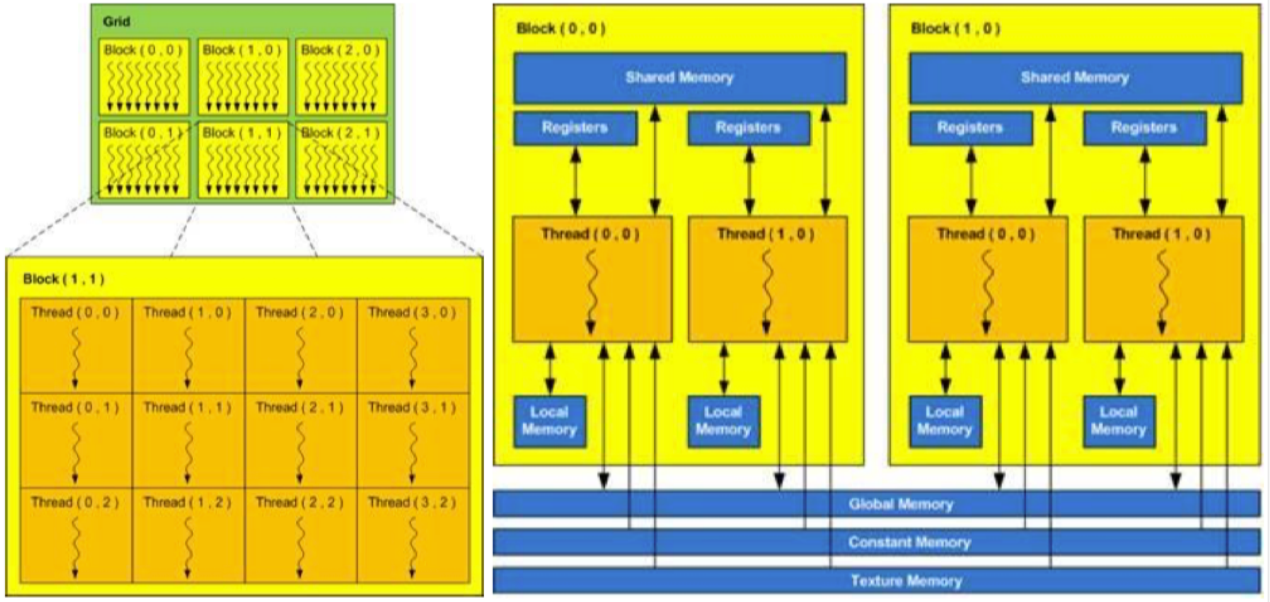
LHCb-TDR-021

Allen: salient features

- Implement parallelism on GPUs at the block and thread level
- One event per block along with sub-event parallelism

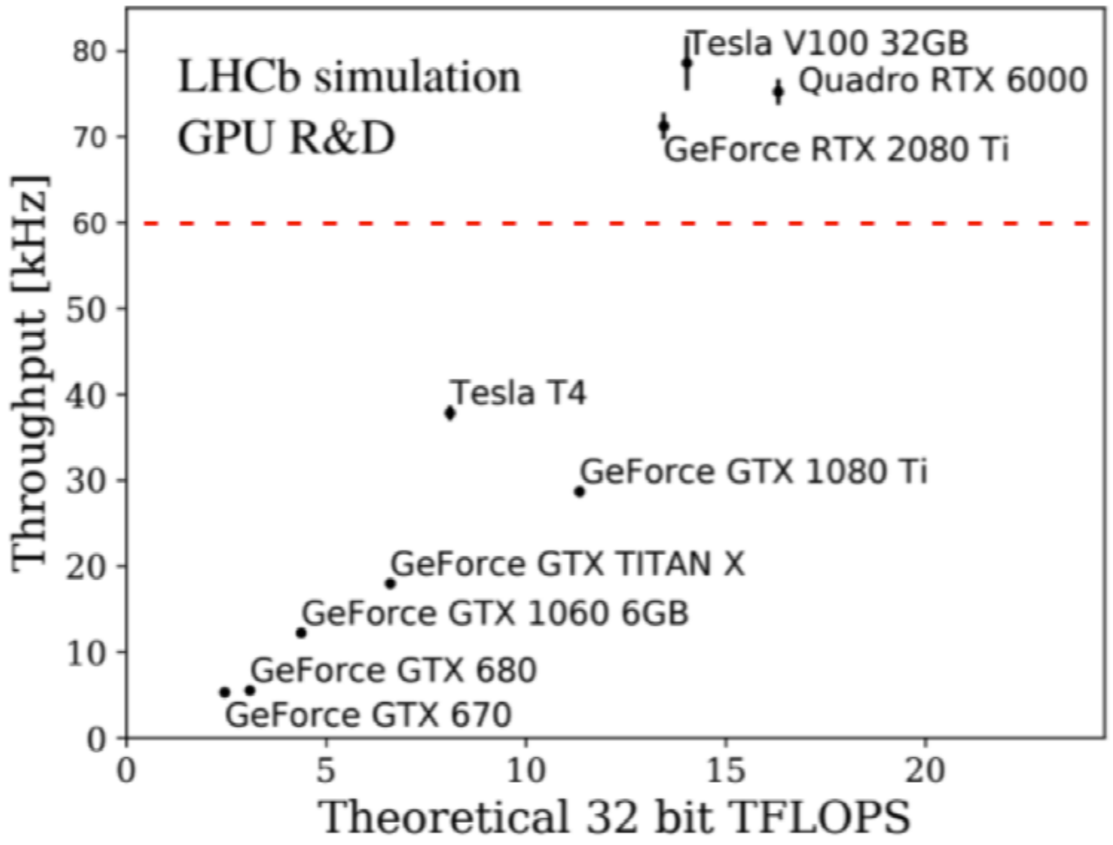
Memory management:

- Memory allocation is done at the start-up of application
- Custom memory manager for GPU memory
- Not dependent on dynamic libraries for memory allocation

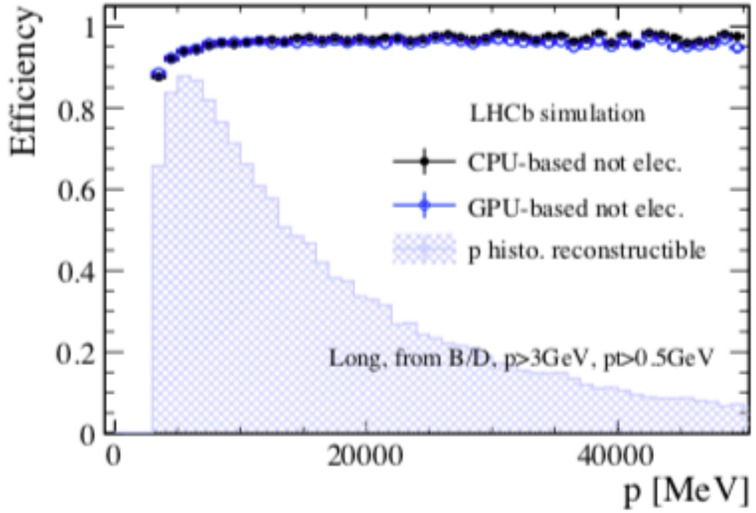


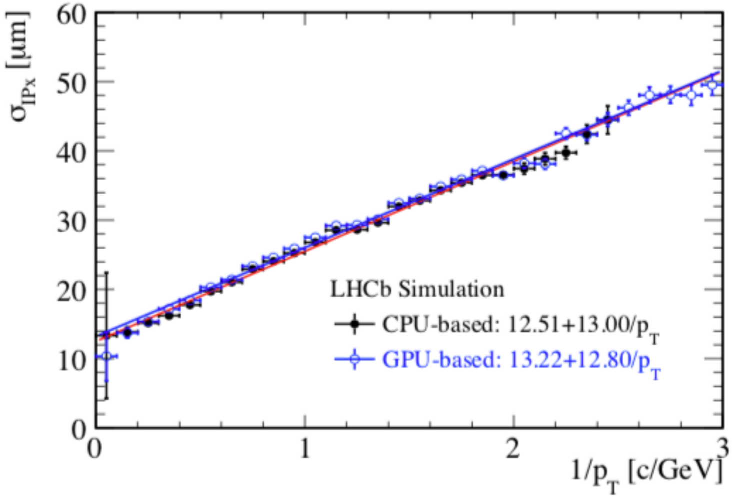
Allen performance (early 2020)

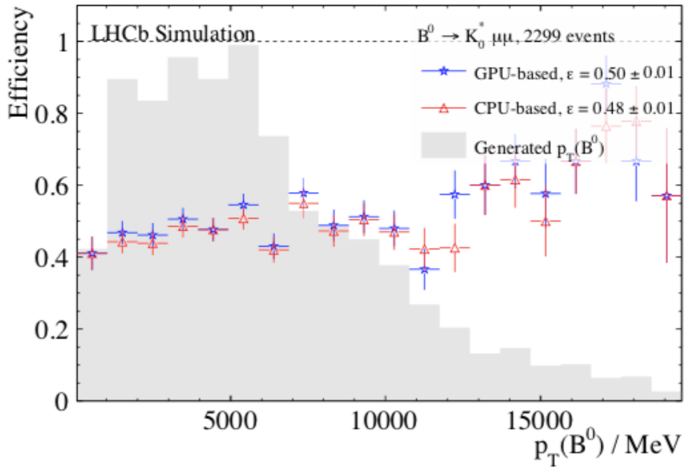
- 60 kHz is the miminum requirement for 30 MHz input rate and 500 GPU cards
- Therefore, Allen can handle the full 30 MHz collision rate with < 500 RTX 2080 Ti GPUs from 2018
- Throughput scales well with theoretical TFLOPs, so Allen will speed up as GPUs improve

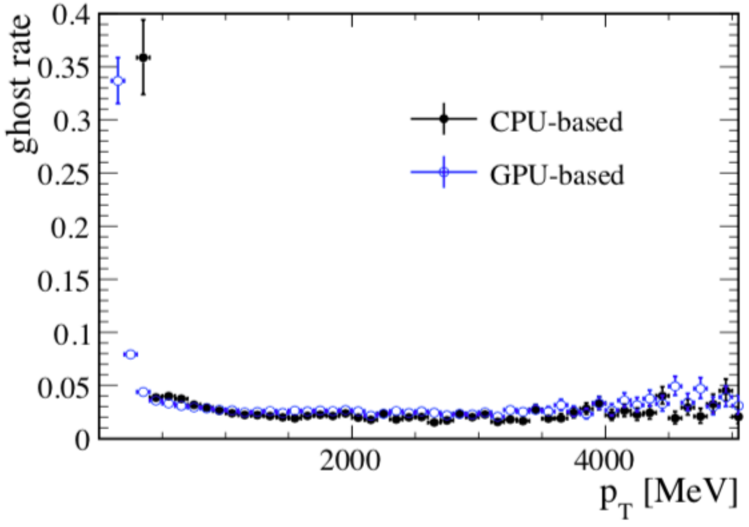


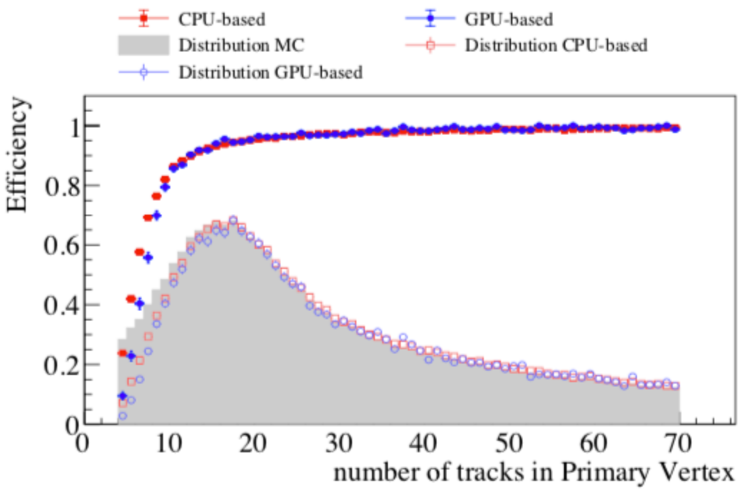
HLT1 on CPU and GPU: same physics performance

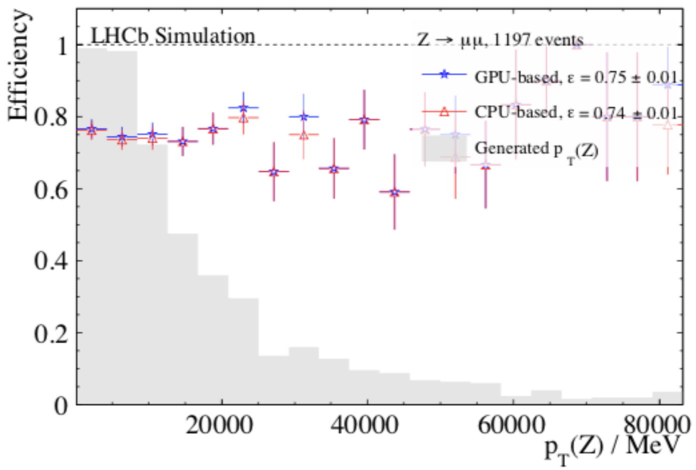




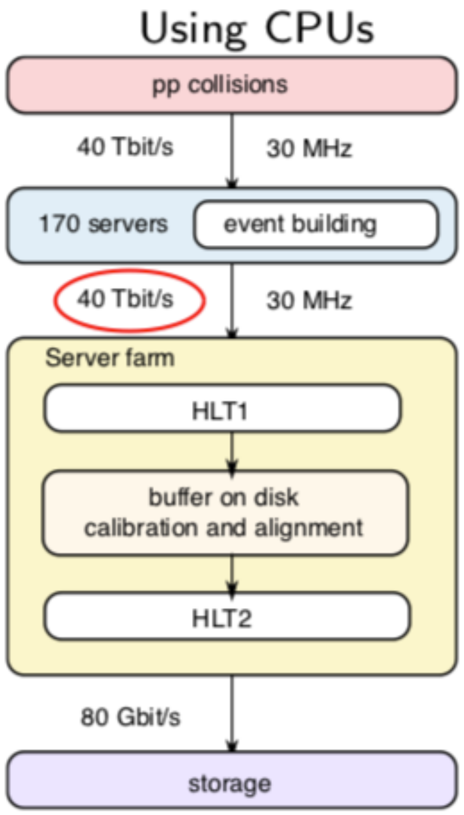


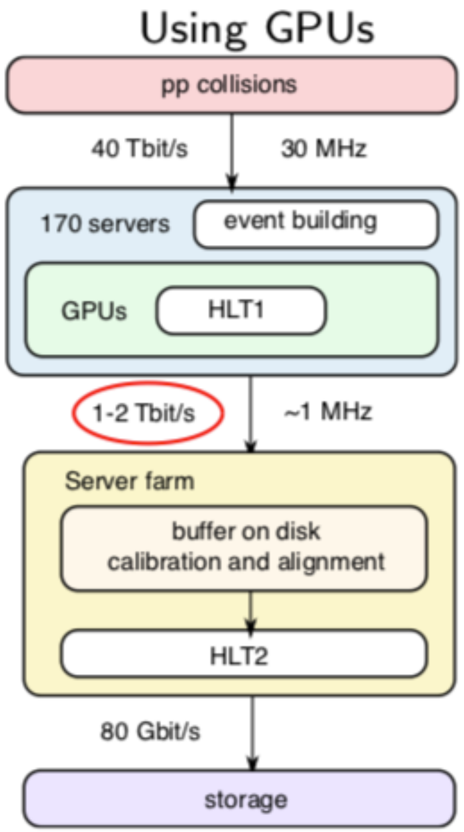






Keine leichte Entscheidung…

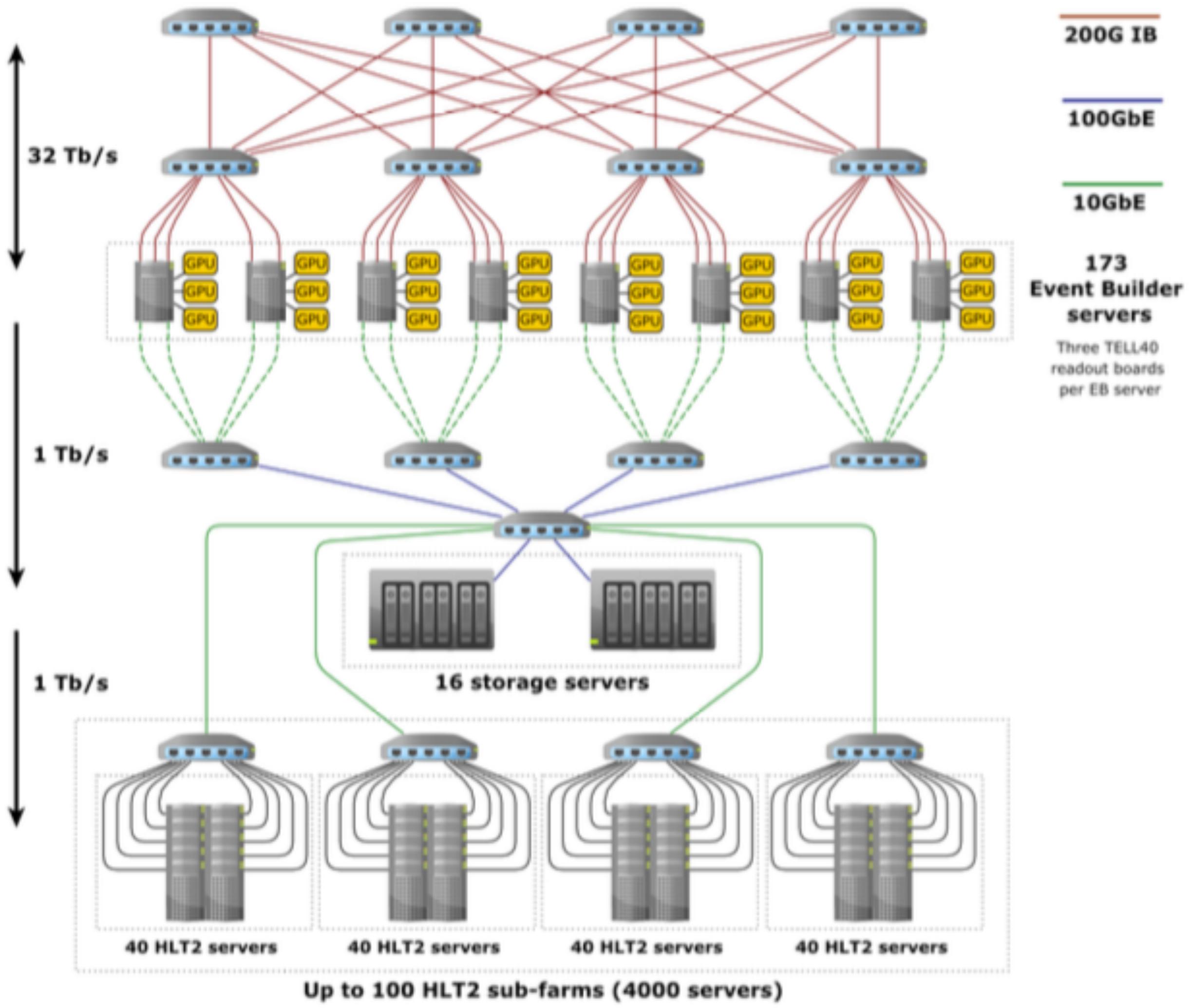


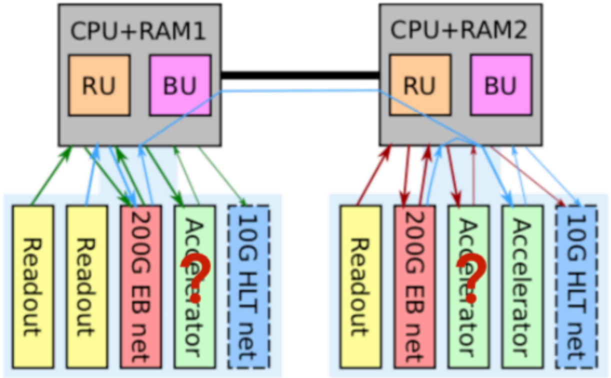


- Both CPU and GPU proposals carried out in the last years
- Extensive studies and developments on both architectures
- Brand new algorithms and ideas on pattern recognition developed on both architectures
- Benefits of running HLT1 on CPUs:
  1. Seamless integration with current infrastructure and operations minimal changes required
  2. Easy scalability
- Benefits of running HLT1 on GPUs:
  1. Reduce network bandwidth between EventBuilder and filter farms
  2. Free up filter farm CPUs for HLT2 only

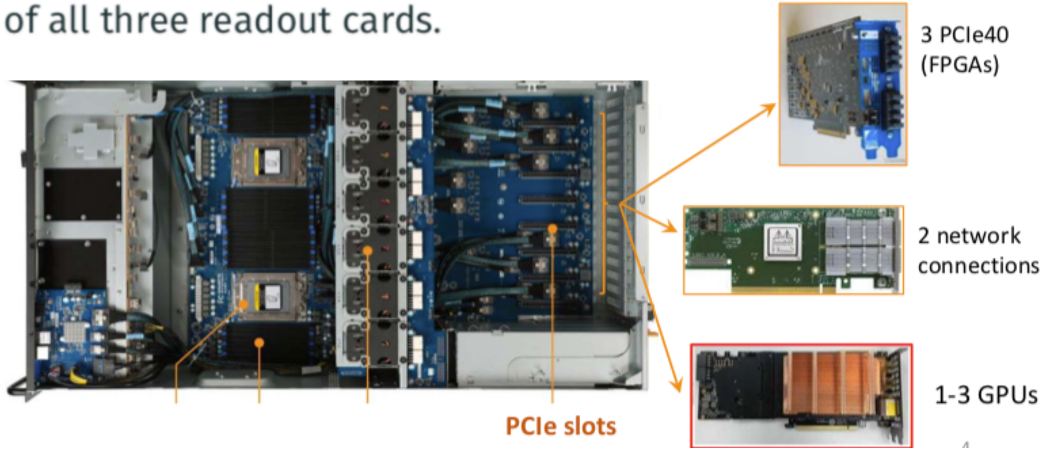
- **Final decision : use GPUs for HLT1**
- All the work and experience gained for HLT1 reconstruction using CPUs crucial to achieve large speed-up also for the HLT2 reconstruction.

Practical implementation

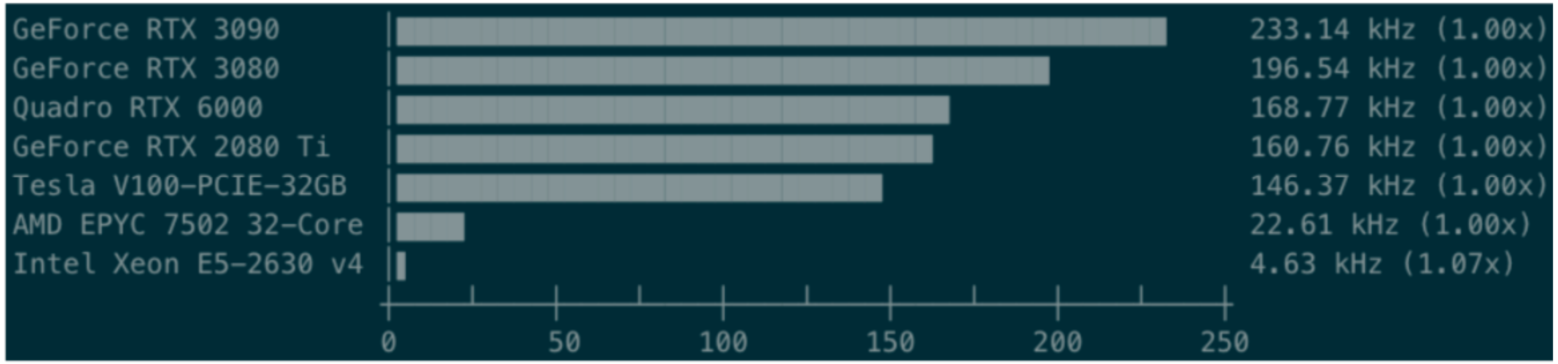


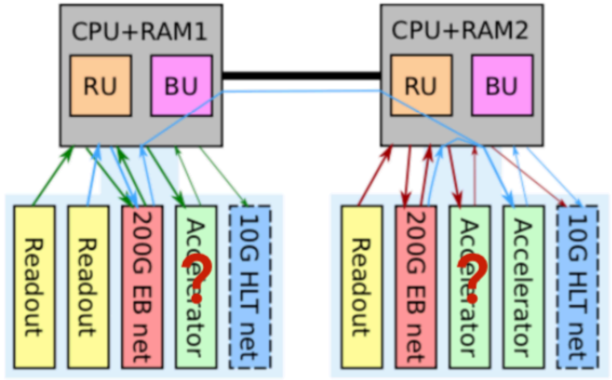


GPU-equipped event builder PC, with traffic of all three readout cards.



Allen performance (today)



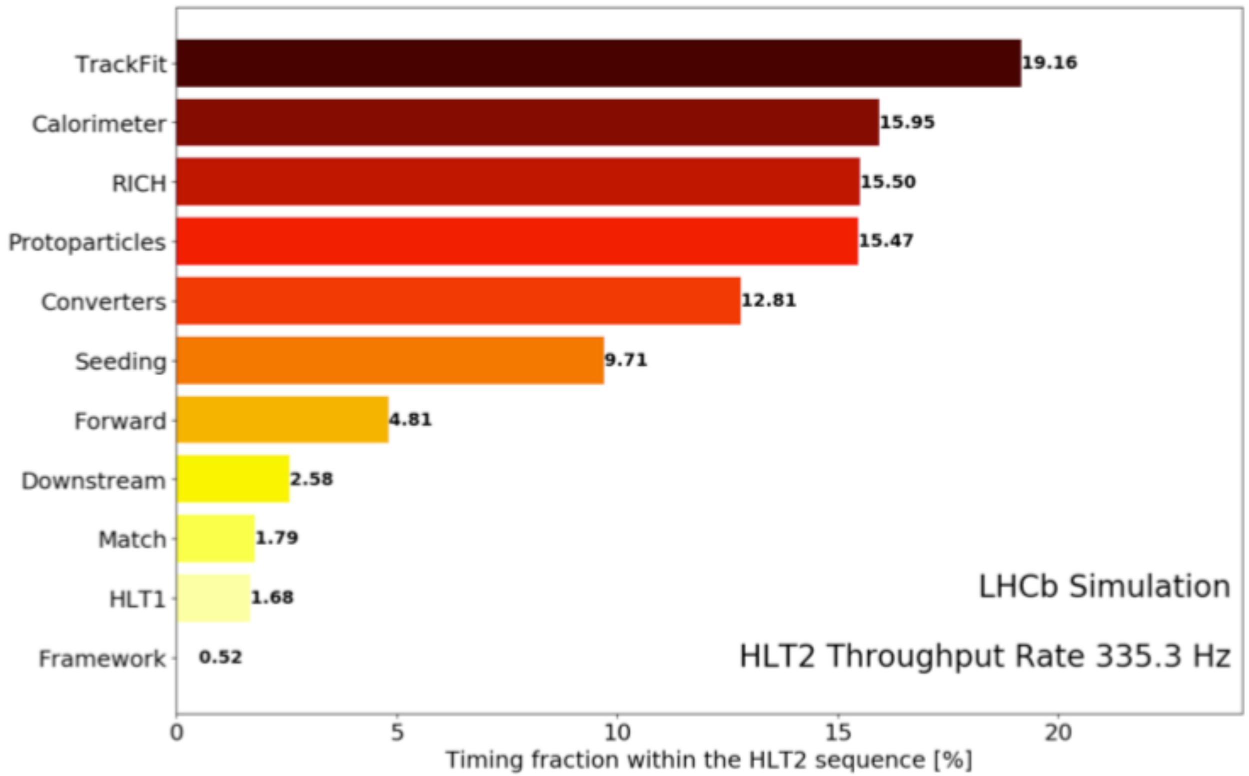


GPU-equipped event builder PC, with traffic of all three readout cards.

- Recent Allen optimizations, and usage of consumer NVIDIA cards allow us to deploy up to 4x the processing power foreseen just one year ago
- Using the 3090 results in one card per EB node, with about 10% headroom remaining. To be validated.

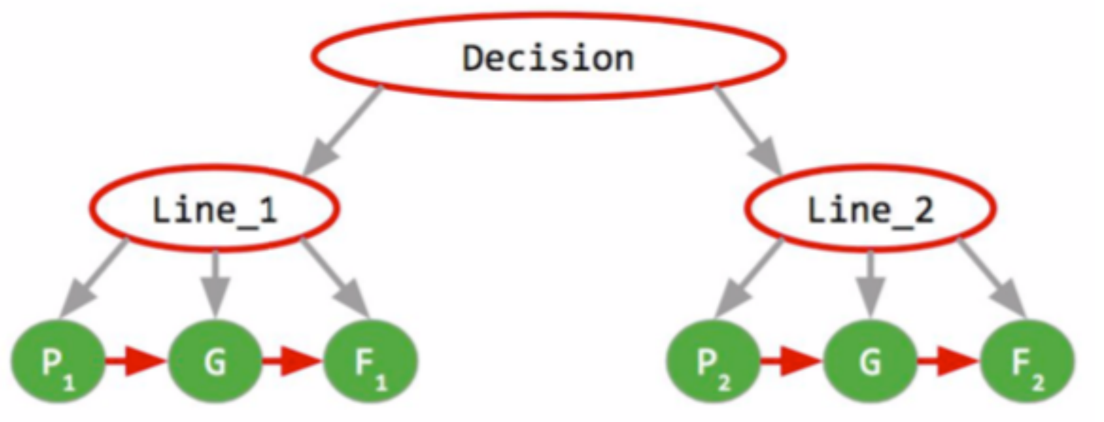
HLT2 status

- The same guiding principles used for optimizing the HLT1 application on CPU hold for HLT2
- However, in addition to track reconstruction, also calorimeters and RICHs must be included
- "Converters" also needed for now
  - They close the gap between CPU- and analysis-friendly event model
  - Their real need depends on evolution of analysis model
- More importantly, about 1000 trigger selection lines must also run and be optimised
- HLT2 throughput rate = HLT1 output rate
  - E.g. 3k CPU nodes would be currently needed by HLT2 to match 1MHz HLT1 total rate



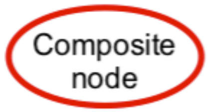
HLT2 selection framework

- *O*(1000) selections:
  - A very complex graph → execution must be optimised
- Data flow:
  - Configurable algorithms properties
  - User-defined inputs/outputs
- Control flow:
  - What should be run and when to stop
- For the execution:
  - Data dependency constructed by matching inputs/outputs
  - Basic nodes ordering respecting data constraints



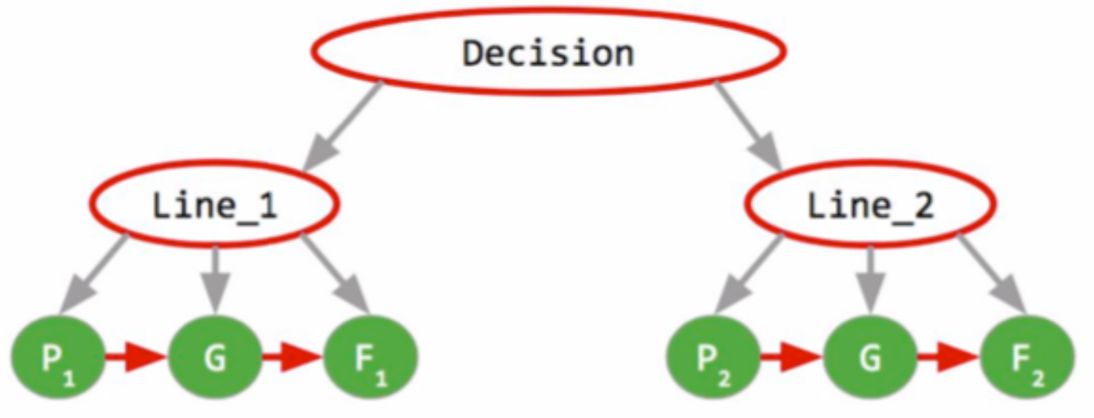
- Basic node:
  - One algorithm node
  - List of data dependencies

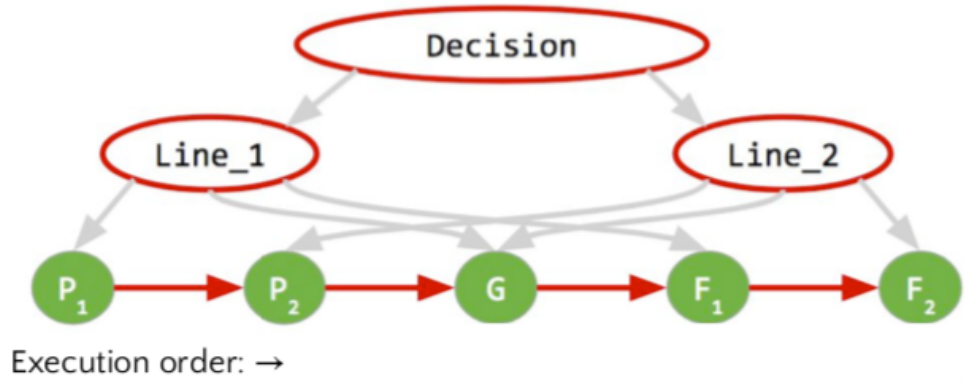
- Composite nodes:
  - Logic (AND | OR | NOT)
  - “Execute all children” or “allow short-circuiting”



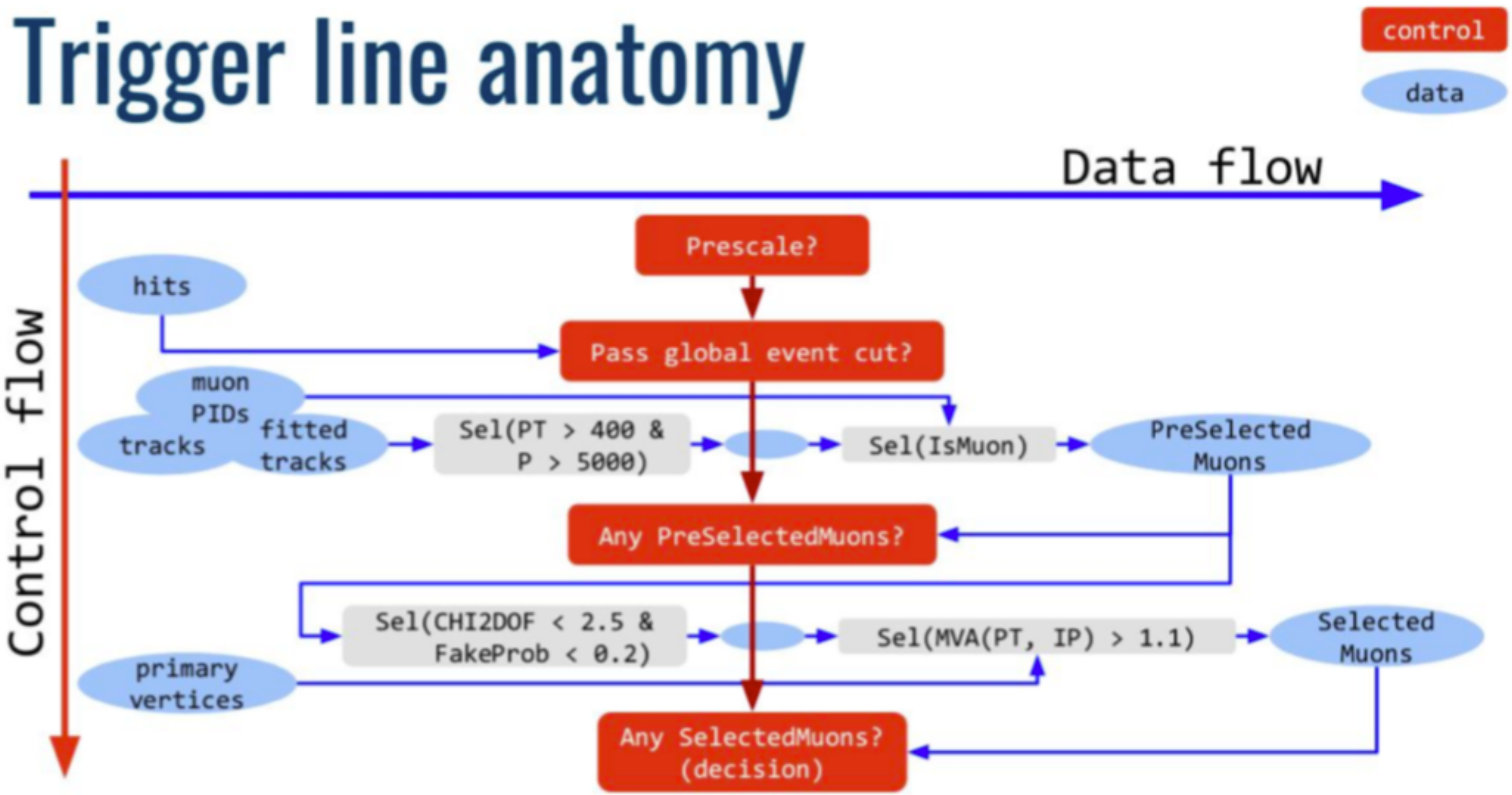
HLT2 selection framework

- *O*(1000) selections:
  - A very complex graph → execution must be optimised
- Data flow:
  - Configurable algorithms properties
  - User-defined inputs/outputs
- Control flow:
  - What should be run and when to stop
- For the execution:
  - Data dependency constructed by matching inputs/outputs
  - Basic nodes ordering respecting data constraints



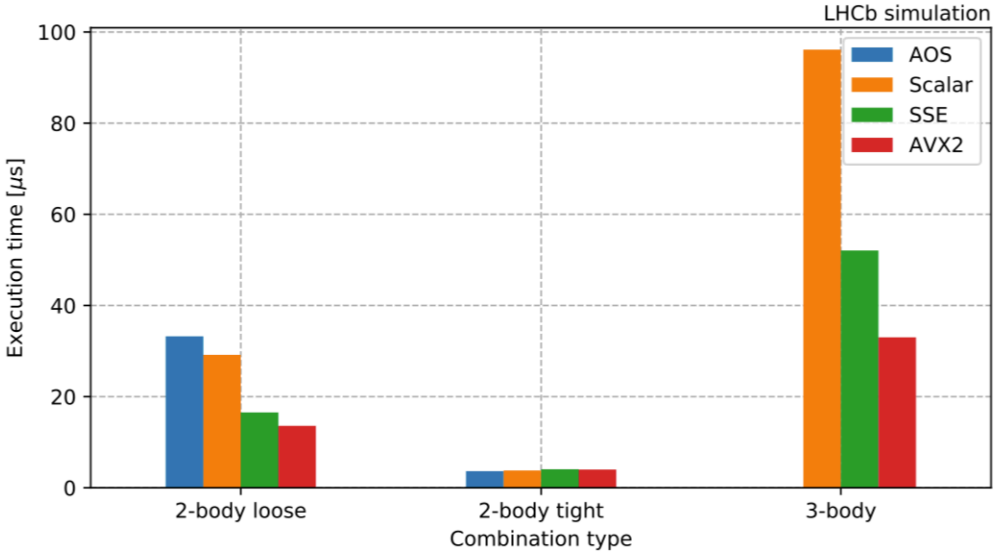


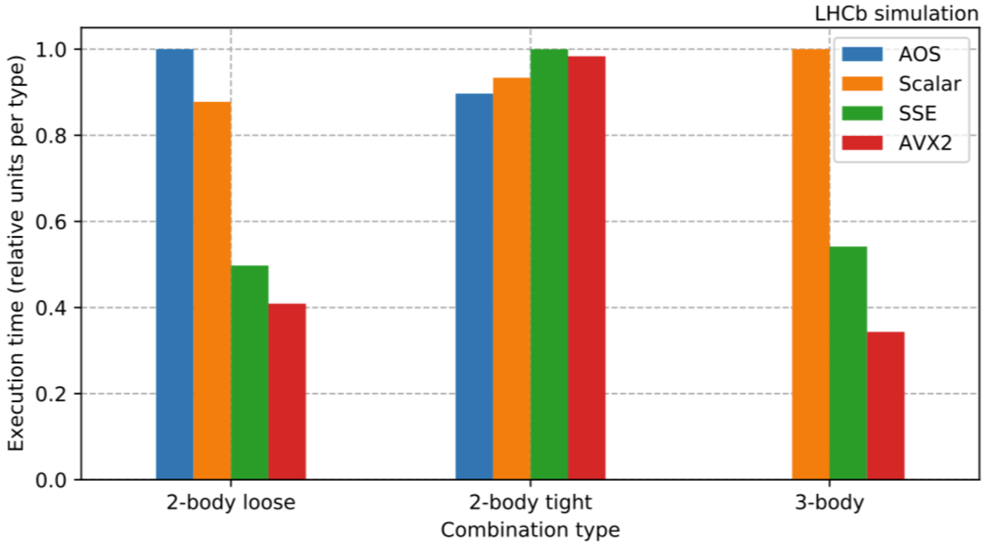
HLT2 selection framework



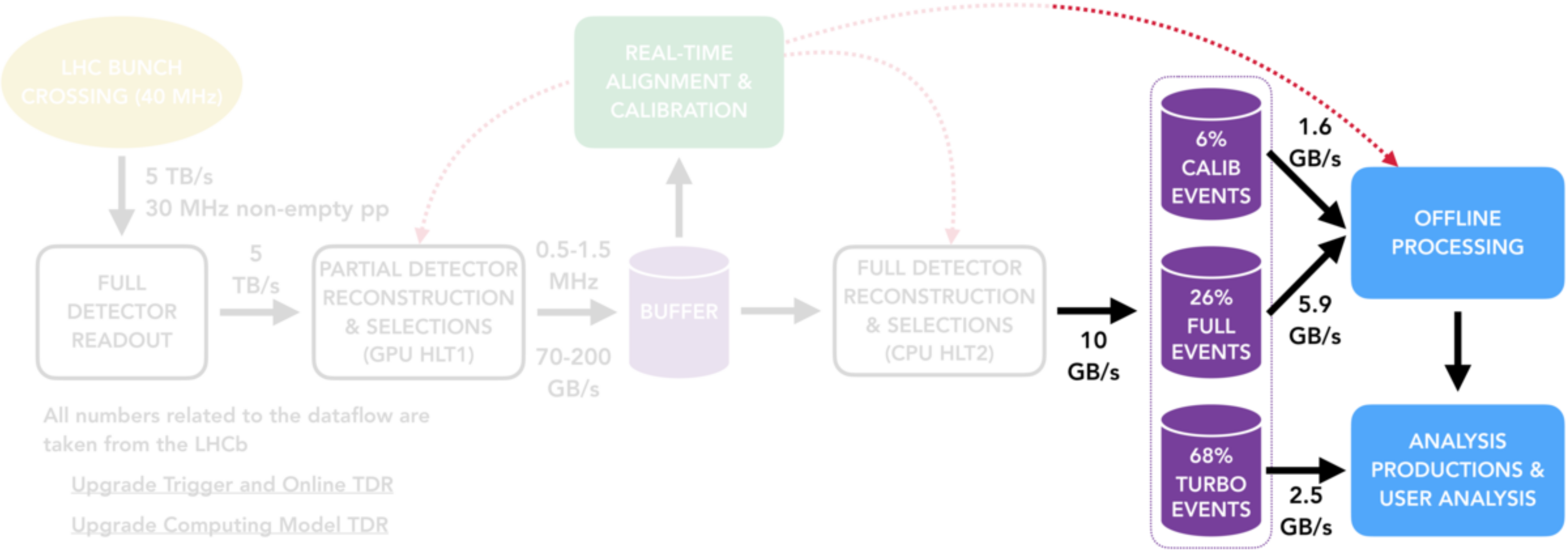
Vectorization of particle selection algorithms

- Two- and three-body particle combination algorithms are being optimised for speed
- Encouraging results when using vector registers on SoA inputs
- Registers must be well filled in order to benefit from vectorization





Outline

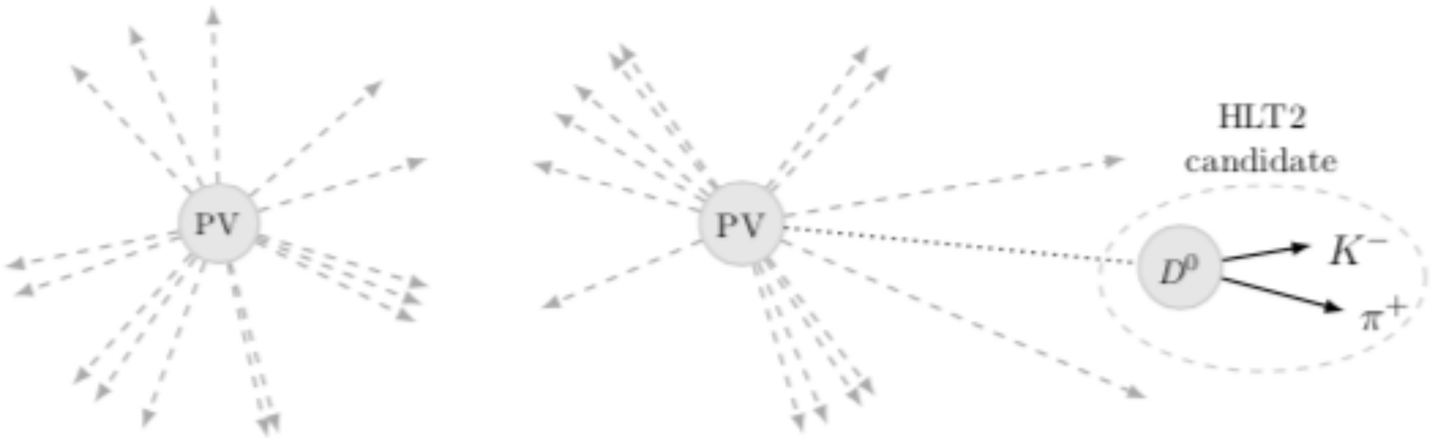

LHC BUNCH CROSSING (40 MHz)
5 TB/s
30 MHz non-empty pp
FULL DETECTOR READOUT
5 TB/s
PARTIAL DETECTOR RECONSTRUCTION & SELECTIONS (GPU HLT1)
0.5-1.5 MHz
70-200 GB/s
BUFFER
REAL-TIME ALIGNMENT & CALIBRATION
FULL DETECTOR RECONSTRUCTION & SELECTIONS (CPU HLT2)
10 GB/s
6% CALIB EVENTS
26% FULL EVENTS
68% TURBO EVENTS
1.6 GB/s
5.9 GB/s
2.5 GB/s
OFFLINE PROCESSING
ANALYSIS PRODUCTIONS & USER ANALYSIS
All numbers related to the dataflow are taken from the LHCb
Upgrade Trigger and Online TDR
Upgrade Computing Model TDR

From online to offline: Persistency model

- In Run2, LHCb explored another “dimension” of data handling
- In a typical HEP experiment, the trigger rate (kHz, MHz) is often quoted, then the bandwidth (MB/s, GB/s) is determined by assuming an average event size
  - RAW banks are typically streamed to offline for event reconstruction
- …but if the event reconstruction is done online by the HLT, then one can decide whether to send offline the entire event or only part of it
- At a fixed bandwidth = rate * event_size, one can then increase the rate, and therefore the physics sensitivity of the experiment, by saving only the “interesting” part of an event!

Persistency model

- Selective persistency: write out only the “interesting” part of the event.

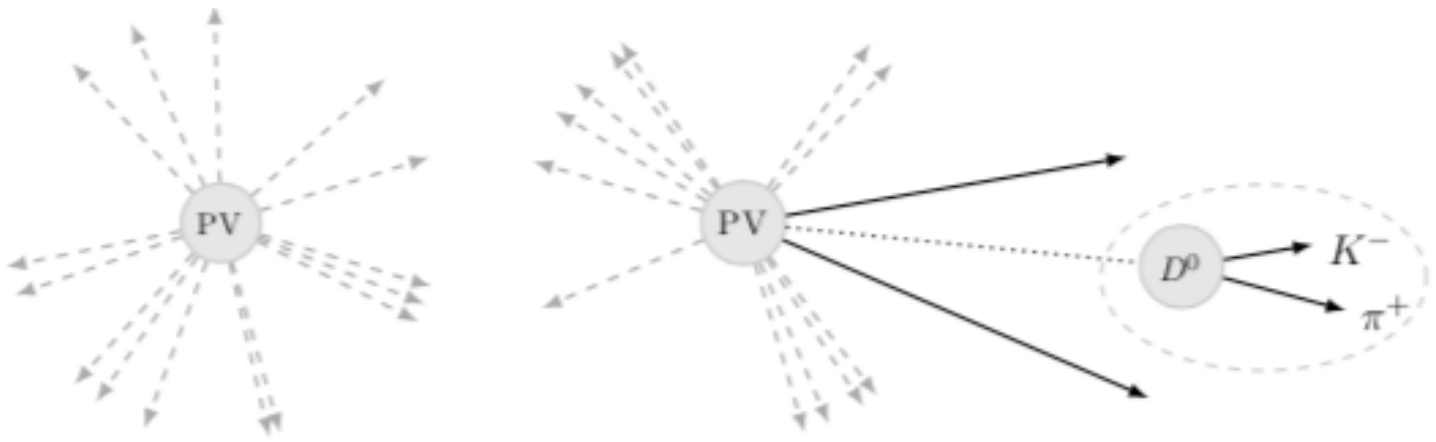


- Turbo stream:
  - Miminum output: only HLT2 signal candidates

Limitations: cannot refit tracks and PVs offline, rerun flavour tagging etc.
Advantage: Event size O(10) smaller than RAW

Persistency model

- Selective persistency: write out only the “interesting” part of the event.

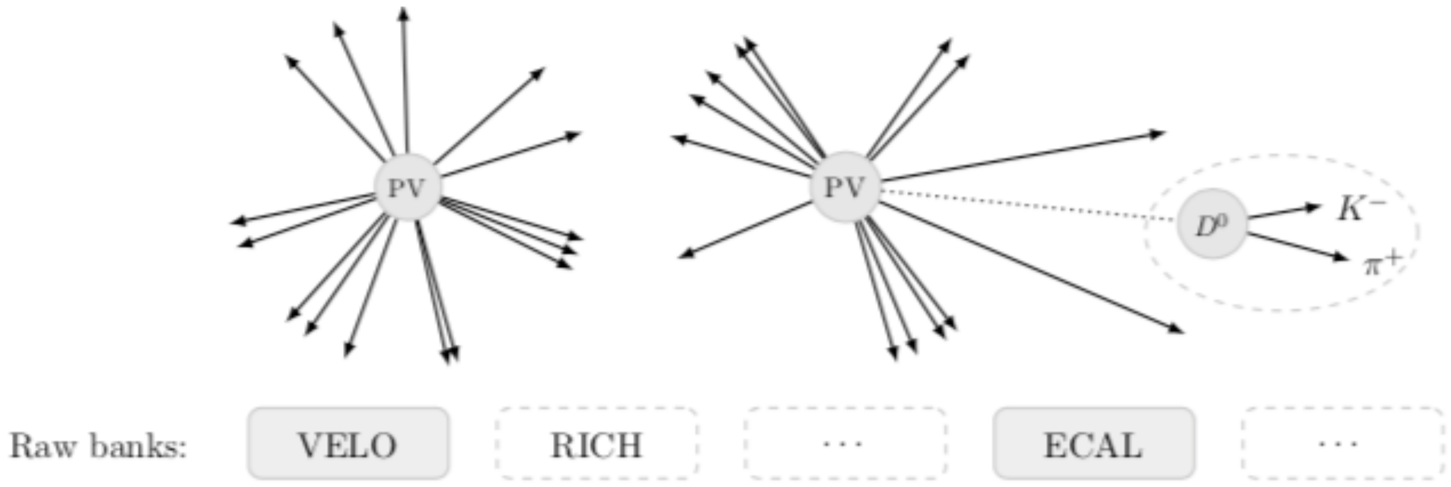


- Turbo stream:
  - Miminum output: only HLT2 signal candidates
  - Optionally: (parts of) pp vertex (e.g. ”cone” around candidate for spectroscopy searches)

Limitations: cannot refit tracks and PVs offline, rerun flavour tagging etc.
Advantage: Event size O(10) smaller than RAW

Persistency model

- Selective persistency: write out only the “interesting” part of the event.



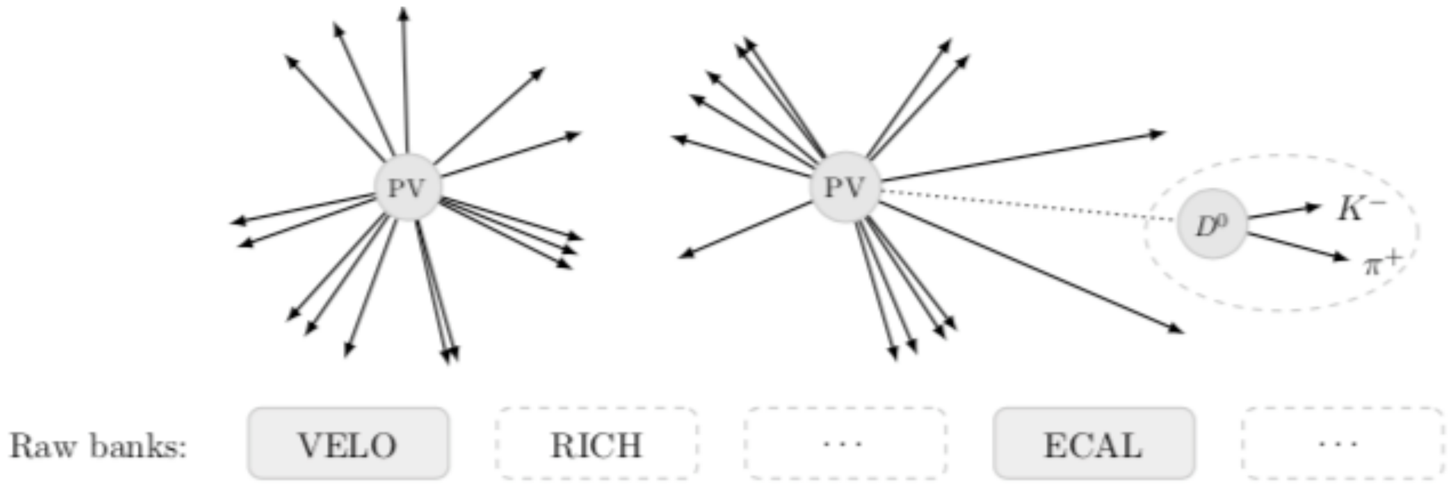
- Turbo stream:
  - Miminum output: only HLT2 signal candidates
  - Optionally: (parts of) pp vertex (e.g. ”cone” around candidate for spectroscopy searches)

Limitations: cannot refit tracks and PVs offline, rerun flavour tagging etc.
Advantage: Event size O(10) smaller than RAW

- FULL stream: all reconstructed objects in the event
  - Optionally adding selected RAW banks

Persistency model

- Selective persistency: write out only the "interesting" part of the event.



- Turbo stream:
  - Miminum output: only HLT2 signal candidates
  - Optionally: (parts of) pp vertex (e.g. "cone" around candidate for spectroscopy searches)

Limitations: cannot refit tracks and PVs offline, rerun flavour tagging etc.
Advantage: Event size O(10) smaller than RAW

- FULL stream: all reconstructed objects in the event
  - Optionally adding selected RAW banks
- TurCal stream: HLT2 candidates and RAW banks
  - Used for offline calibration and performance measurement

Streams and event sizes in Run 2

- Trigger output saved in 3 different streams using different file format

Stream	Content	File format
FULL	Full event information	RDST
Turbo	Selected event information	MDST
Calibration	Full event information + raw banks	RAW or RDST

Run 2 event sizes

stream	event size (kB)	event rate (kHz)	rate fraction	throughput (GB/s)	bandwidth fraction
FULL	70	7.0	65%	0.49	75%
Turbo	35	3.1	29%	0.11	17%
TurCal	85	0.6	6%	0.05	8%
total	61	10.8	100%	0.65	100%

Event size: Turbo/FULL ~0.5

N.B Turbo event size is an average. It ranges from a few kB (minimal persistence) to full event size

Extrapolation of Run2 rates to Run3 conditions

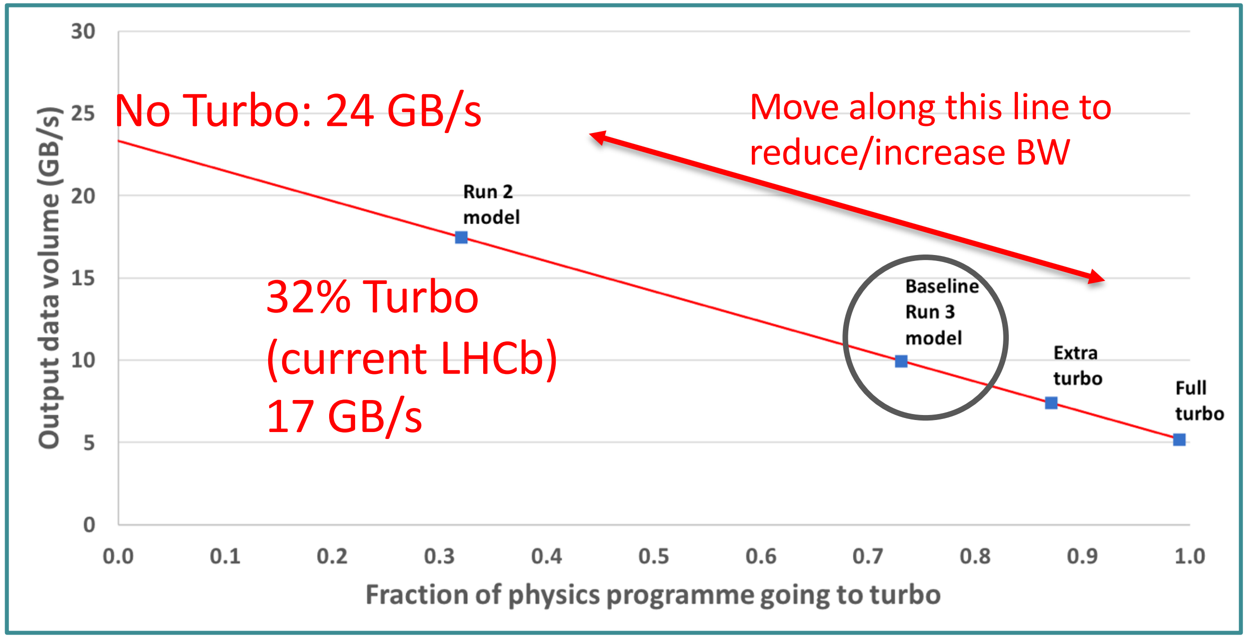
- With the upgrade conditions several factors need to be applied
  - Luminosity $4*10^{32}$ $cm^{-2}s^{-1}$ to $2x10^{33}$ $cm^{-2}s^{-1}$
  - HLT efficiency increase because of removal of L0 hardware trigger
  - Raw event size increase due to pileup, according to simulation
- Without any changes the HLT output rate would increase in Run 3 to 17.4 GB/s

	Run 2 (GB/s)	Lumi	No L0	Raw size	Run 3 (GB/s)
Full	0.49	x5	x2	x3	14.7
Turbo	0.11	x5	x2	x1	1.1
Calibration	0.05	x5	x2	x3	1.6
Total	0.66				17.4

Event size: Turbo/FULL ~0.167

Evolution of physics programme

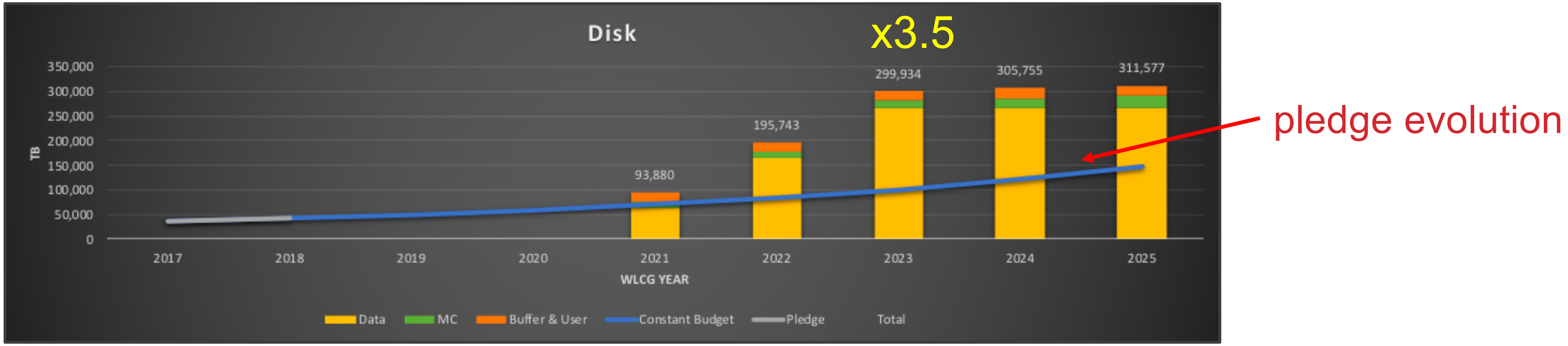
- Moving a larger fraction of the physics programme to Turbo decreases the output bandwidth
- Turbo events are considerably smaller (16 % of Full size)
- Some selections need to stay in Full
  - Keep some flexibility, recover from possible errors, develop new analysis ideas



- For the baseline model we assume 60% of the physics selections currently on FULL stream migrating to Turbo
- Massive migration, not trivial!
- Baseline model assumes 73% of the physics selections on Turbo
- Corresponds to a BW of 10 GB/s

Baseline bandwidth: evolution of the model

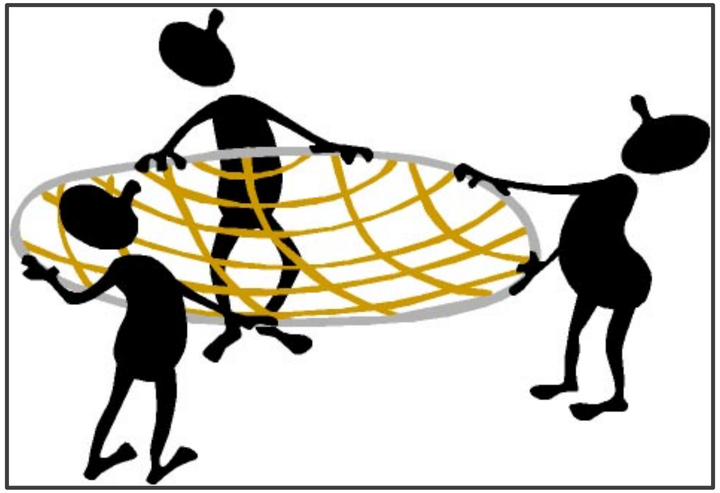
- Can we fit 10 GB/s in a reasonable amount of storage resources ?
- First attempt, presented in summer 2018 to LHCC and WLCG resulted in an amount of disk **3.5 times larger** than what expected in a “constant budget” evolution model !
- mitigation strategies clearly needed



First attempt to fit upgrade data on disk (summer 2018)

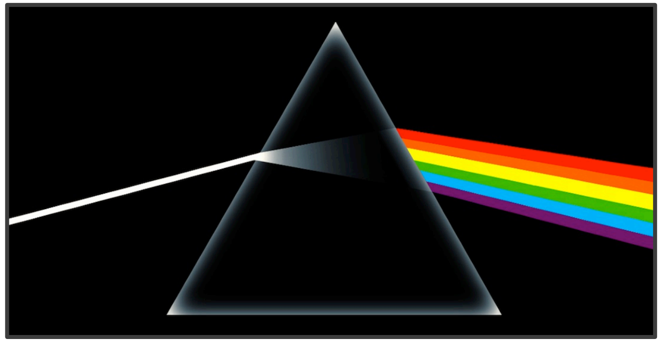
Baseline bandwidth: evolution of the model

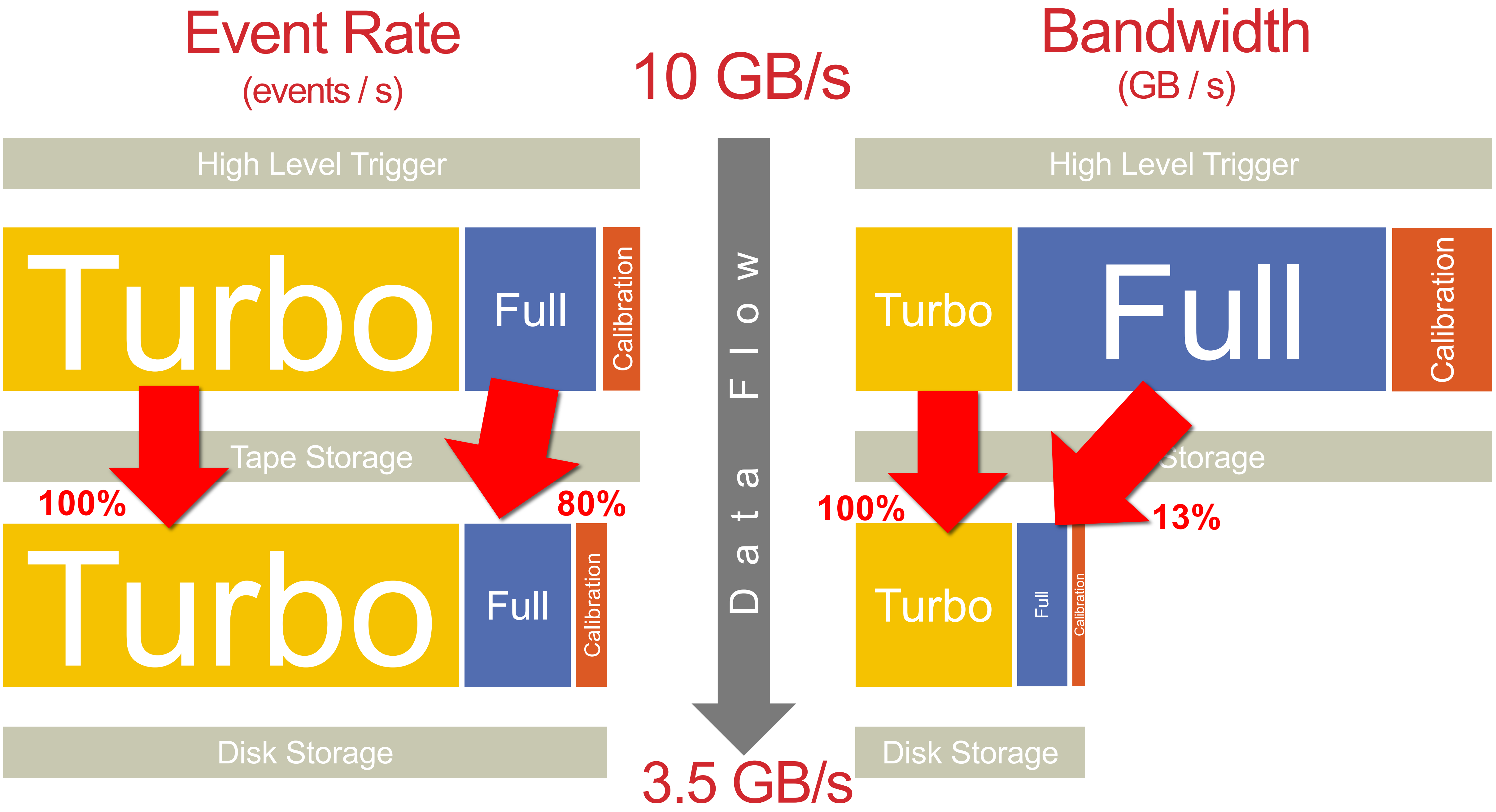
- Idea! Use cheap storage as a **safety net** :
  - save the desired BW on tape
  - Profit of *sprucing* to reduce data volume to disk.
  - …but with the possibility of reprocessing



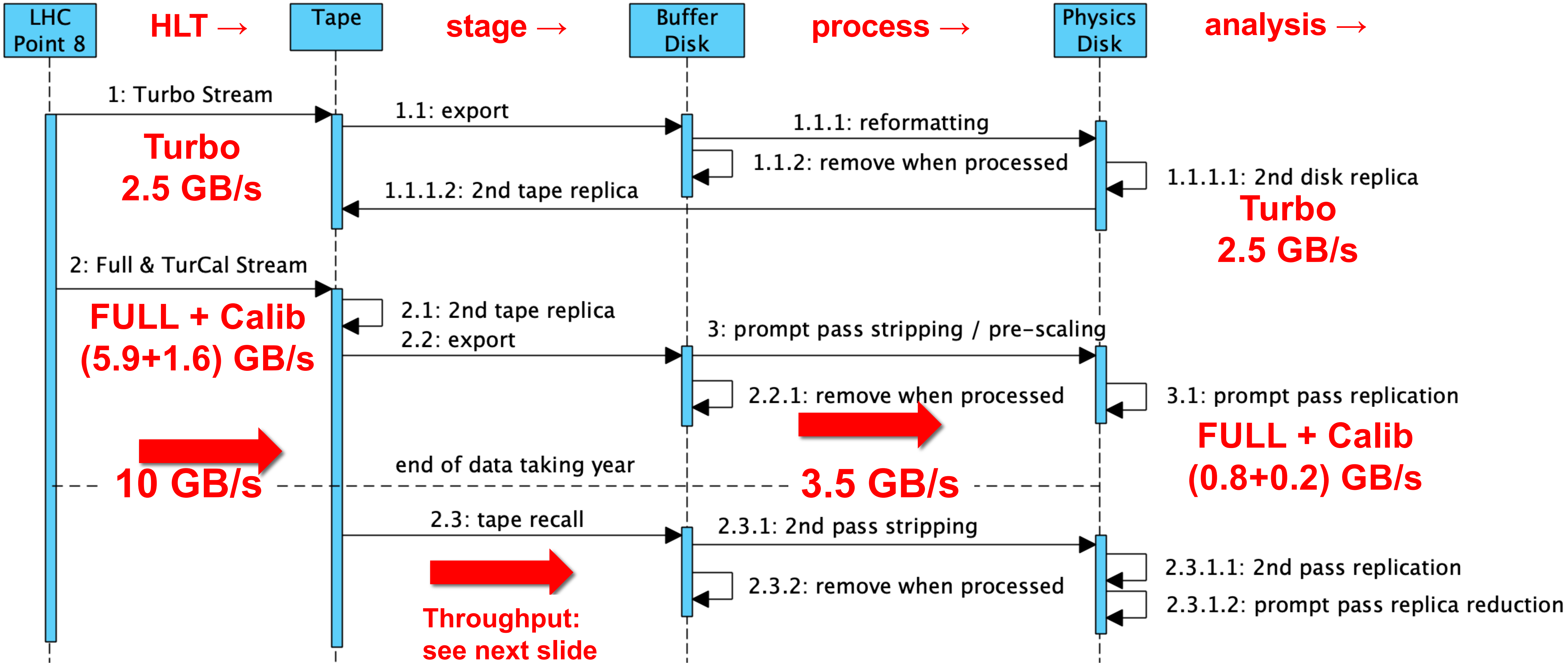
- Operationally more challenging
- Much safer from the physics point of view

- Sprucing == offline processing of data with a large set ($O(10^3)$) of specialised selections analysis oriented
  - Similar to Turbo trigger selections
  - High event retention (~80%)
  - Use selective persistence to substantially reduce data volume
  - Output format is MDST




Event Rate
(events / s)
10 GB/s
Bandwidth
(GB / s)
High Level Trigger
High Level Trigger
Turbo
Full
Calibration
Turbo
Full
Calibration
Tape Storage
Storage
100%
80%
100%
13%
Turbo
Full
Calibration
Turbo
Full
Calibration
Data Flow
Disk Storage
Disk Storage
3.5 GB/s

Data Processing Workflow per Data Taking Year



Bandwidth to and from tape

- CERN and Tier1 tape must keep up with the data throughput coming from online
- During (Extended-)Year-End-Technical-Stops, data will be recalled from tape
- Not a full re-reconstruction, only another filtering & slimming pass
- The staging throughput depends on the time required to fully stage
  - And on the dataset luminosity
- Expect ~4x increase with respect to Run2

Country	Site	Tape Read BW (GB/s)	Tape Write BW (GB/s)
CERN		4.24	5.50
Tier1 sites			
France	CC-IN2P3	0.49	0.63
Germany	GridKA	0.86	1.12
Italy	CNAF	0.86	1.12
Netherlands	SARA/NIKHEF	0.34	0.44
Russia	RRCKI	0.34	0.44
Spain	PIC	0.23	0.29
UK	RAL	1.13	1.46
TOTAL Tier1 sites		4.24	5.50

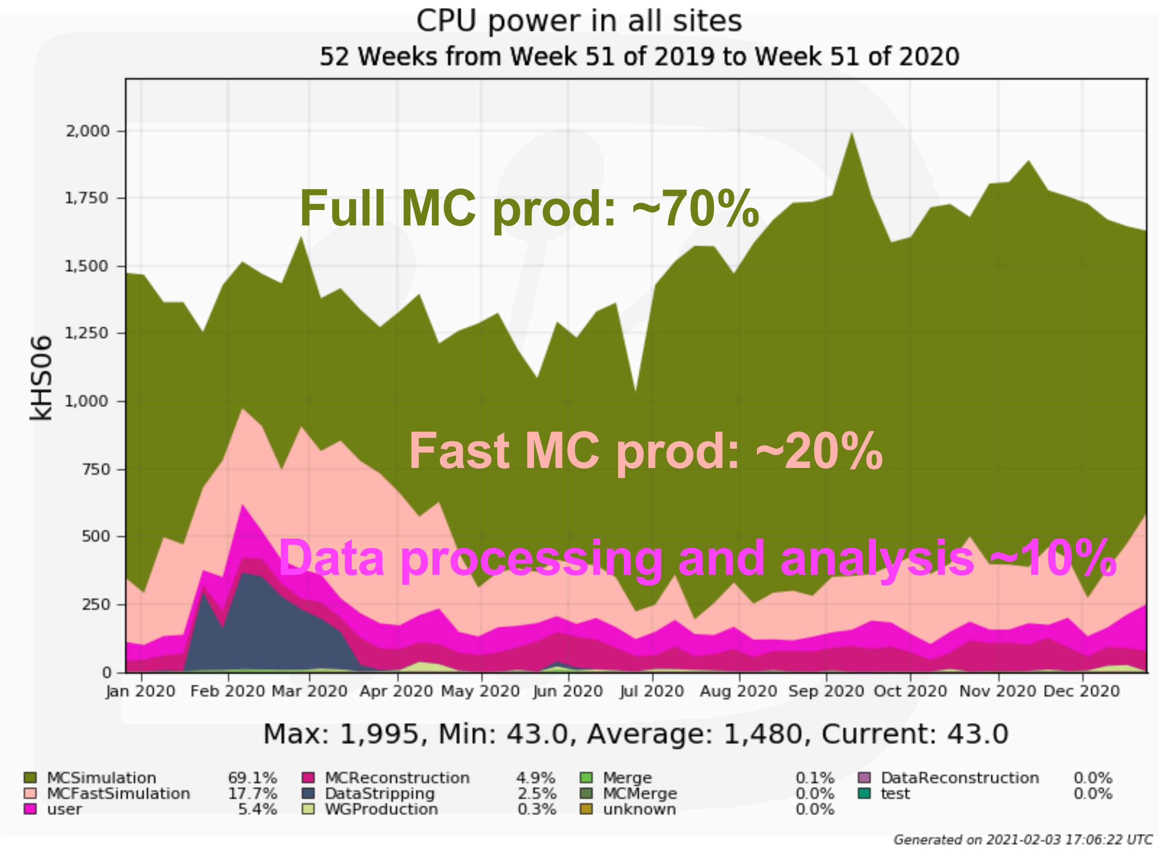
What about CPU ?

- CPU is dominated by MC production (~90% of CPU power)
- Expected to be the same at the Upgrade

- Scale current MC production to estimate the CPU needs

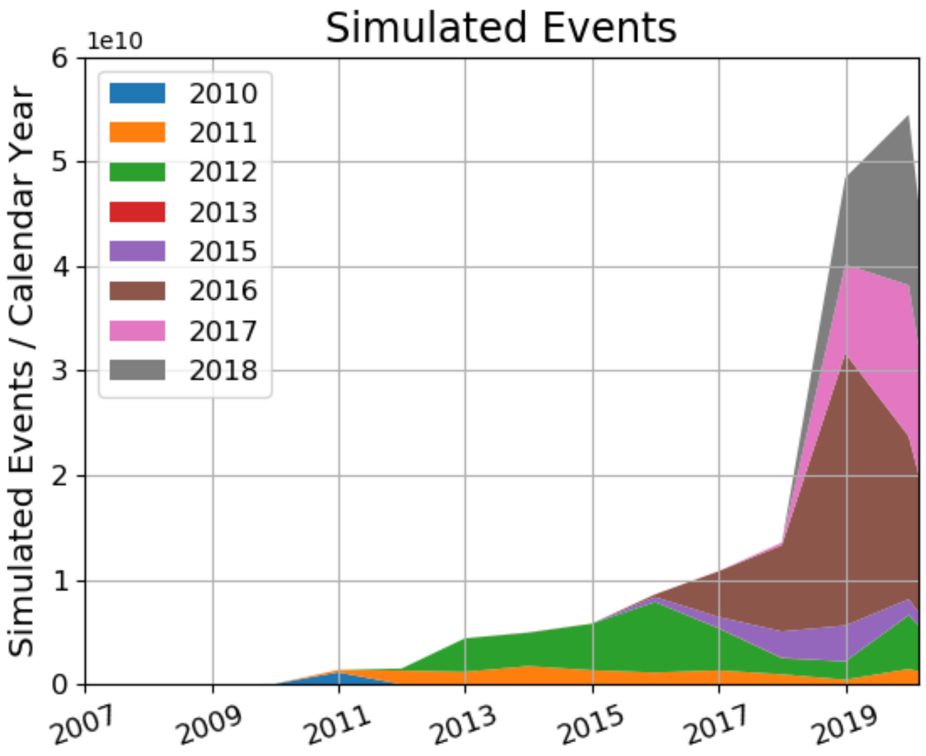
- Number of needed MC events scale with luminosity
- Seen “experimentally” in Run 2
- Well justified by physics
  - Events signal-dominated
  - Generally pure selections
  - $L_{int} \times \varepsilon_{trig}$ is a good proxy for yield

- Assume the same scaling for Upgrade



Fast(er) simulation

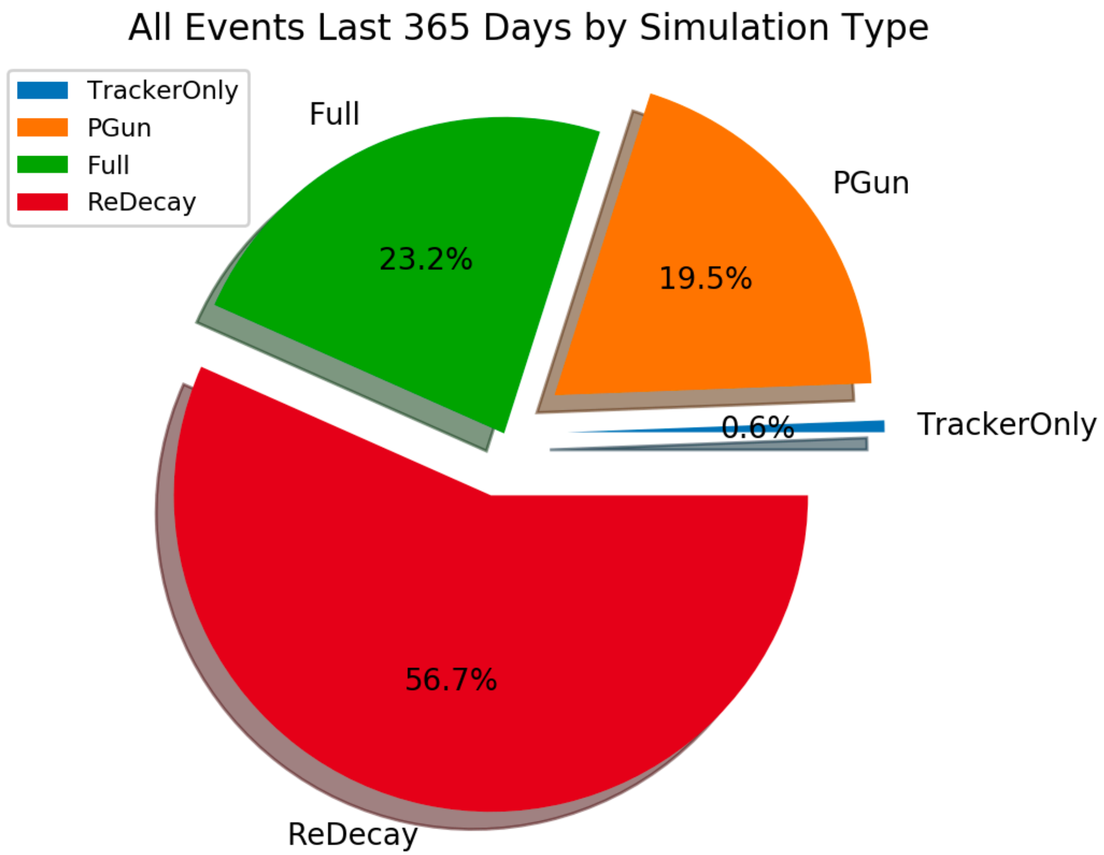
- Assumptions on simulated event volume
  - N. of MC events scales with L_{int}
  - MC production for a data taking years extends over the following 6 years
  - MC events saved in MDST format (x40 size reduction!)
- Implementation of fast simulation techniques already resulted in a leap in the number of simulated events
  - 2018→2019: 4x and only 30% more CPU



Year	Simulated events (10^9)	Stored events (10^9)	Ratio	CPU work kHS06.y	CPU per event kHS06.s	LFS TB
2017	10.3	4.2	40.3%	817	2.50	640
2018	12.0	3.0	25.3%	1009	2.65	550
2019	45.0	6.9	15.2%	1290	0.90	1110
2020	53.0	16.8	31.7%	1357	0.81	2010

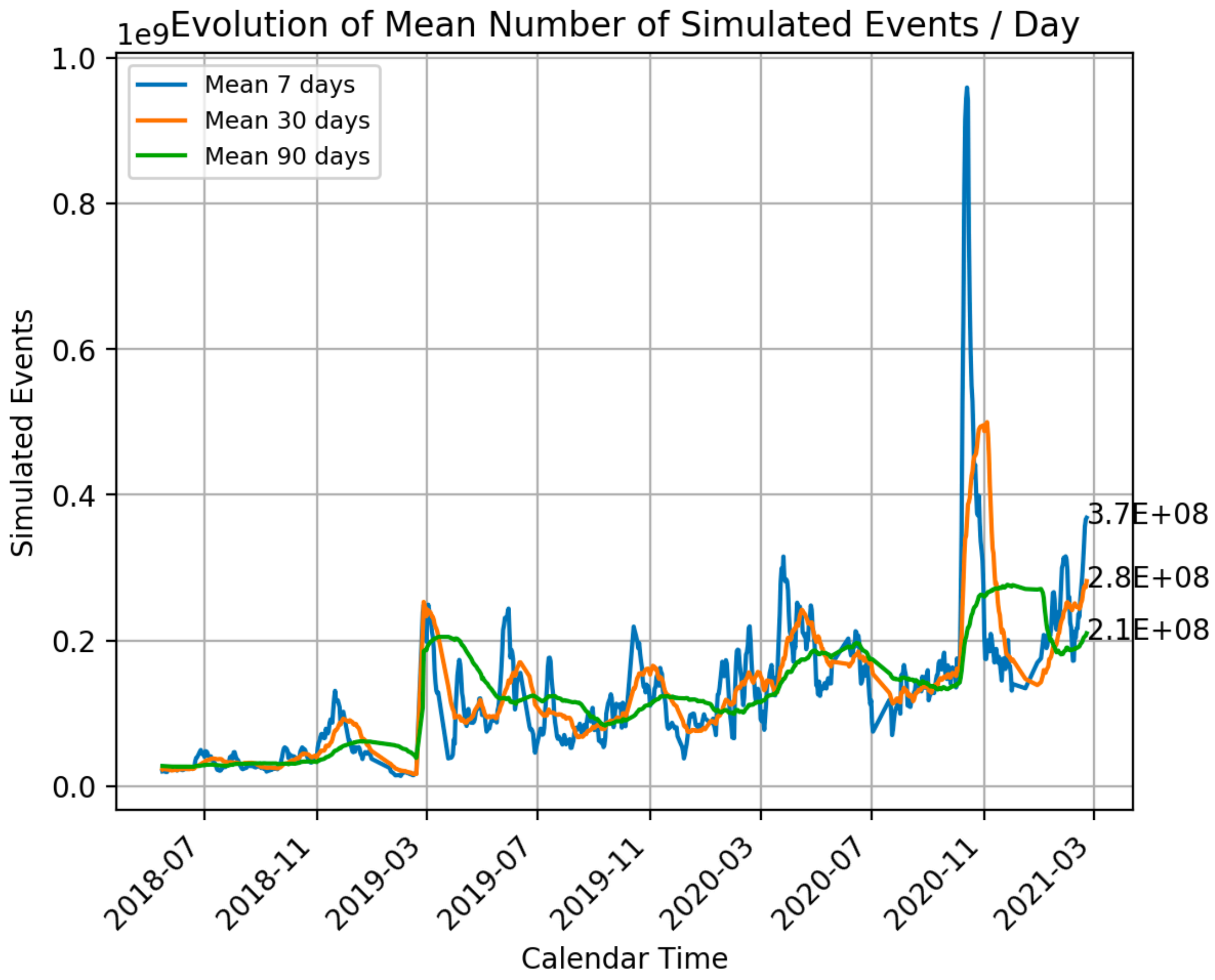
Successful adoption of fast simulations

- **Full –** full Geant4 detector simulation
- **PGun –** single signal particle spawned with kinematics configured to follow distribution (no full pythia event) Factor 50 speed increase
- **ReDecay –** re-use the underlying event but generate and simulate new signal decays every time Eur. Phys. J C78 (2018) 1009 Factor 10-20 speed increase
- **TrackerOnly** simulation – Factor 10 speed increase
- **SplitSim –** only simulate full event if required condition is passed e.g. if a photon converts to e+e- Speed up depends on condition

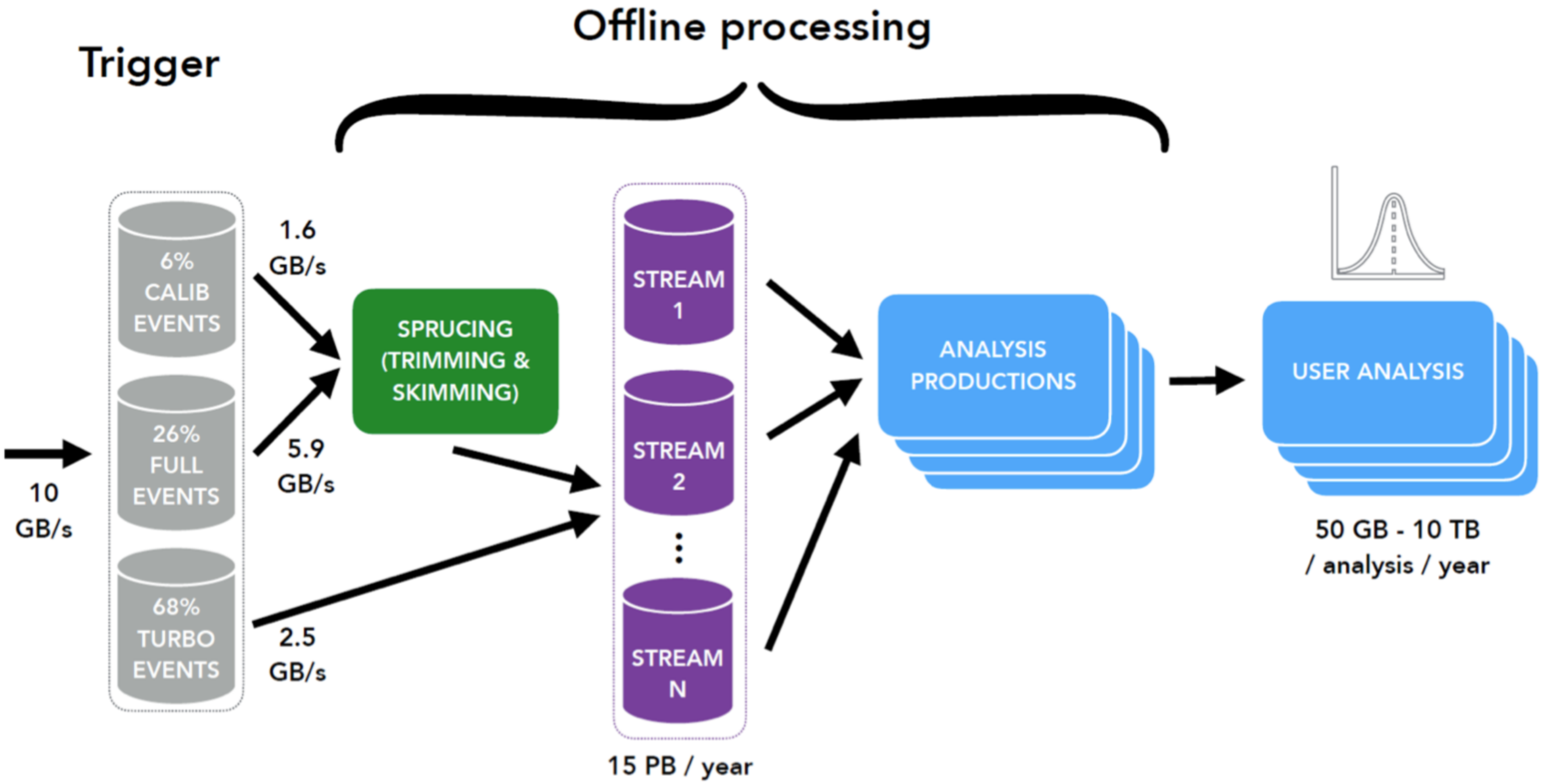


Successful adoption of fast simulations

- **Full –** full Geant4 detector simulation
- **PGun –** single signal particle spawned with kinematics configured to follow distribution (no full pythia event) Factor 50 speed increase
- **ReDecay –** re-use the underlying event but generate and simulate new signal decays every time Eur. Phys. J C78 (2018) 1009 Factor 10-20 speed increase
- **TrackerOnly** simulation – Factor 10 speed increase
- **SplitSim –** only simulate full event if required condition is passed e.g. if a photon converts to e+e- Speed up depends on condition



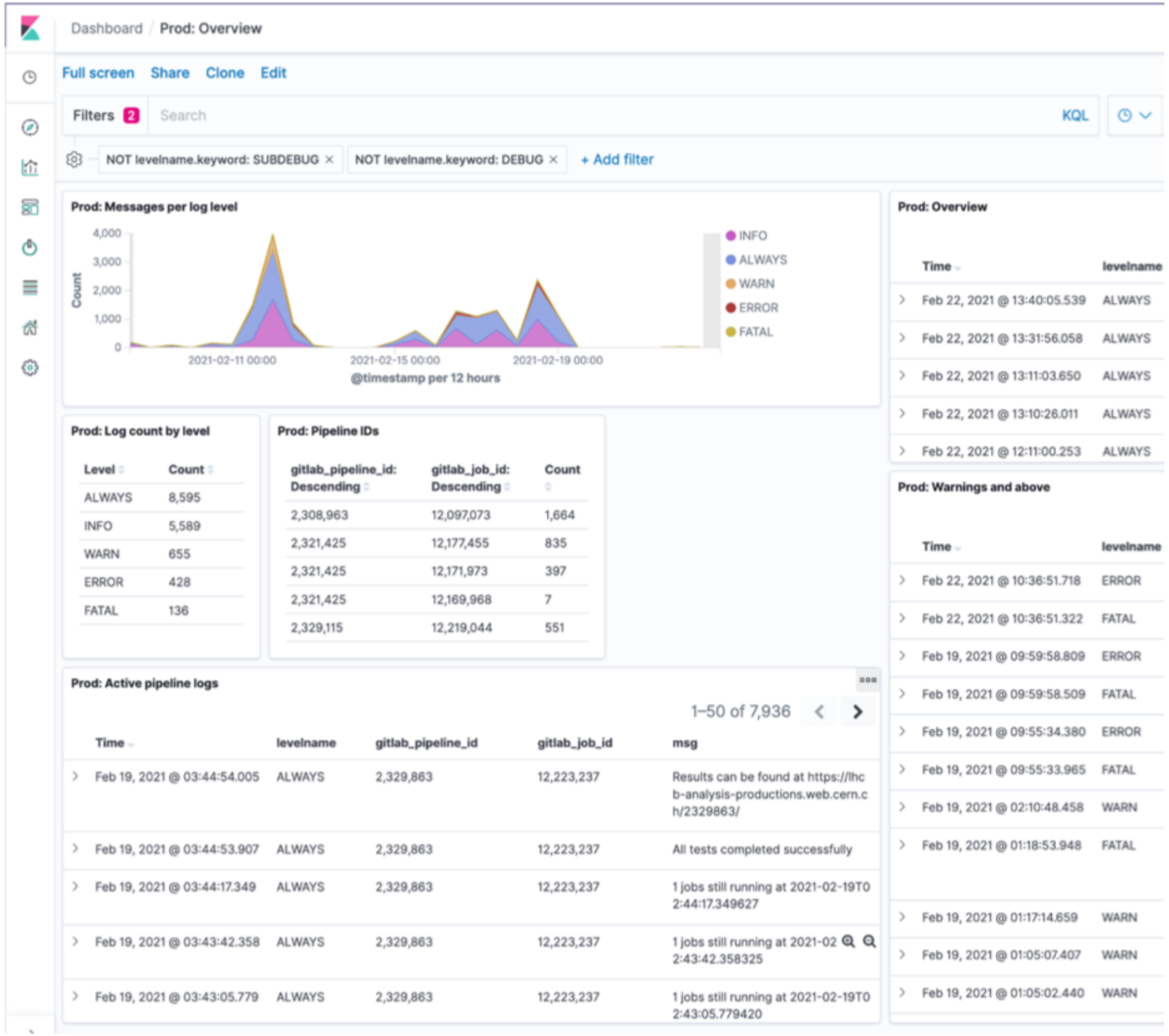
Outline


Offline processing
Trigger
6% CALIB EVENTS
26% FULL EVENTS
68% TURBO EVENTS
10 GB/s
1.6 GB/s
5.9 GB/s
2.5 GB/s
SPRUCING (TRIMMING & SKIMMING)
STREAM 1
STREAM 2
STREAM N
15 PB / year
ANALYSIS PRODUCTIONS
USER ANALYSIS
50 GB - 10 TB / analysis / year

Data Analysis

- In Run 1 + 2 analysts create **nTuples individually …**does not scale well for Run 3
  - 1000s of **faulty jobs** can be submitted instantly
  - Time consuming - O(weeks) for Run 1 + 2 tuples - failed jobs re-submitted manually by user
  - **No analysis preservation** infrastructure

- In Run 3 submit jobs centrally using **DIRAC transformation System** (Analysis Productions
  - MC data is already produced this way
  - Does not require analyst to babysit jobs
  - Jobs can be **tested automatically** with GitLab CI
  - Job details/configuration/logs **automaticall preserved** in LHCb bookkeeping/EOS
  - Automated error interpretation/advice
  - Results displayed on webpage

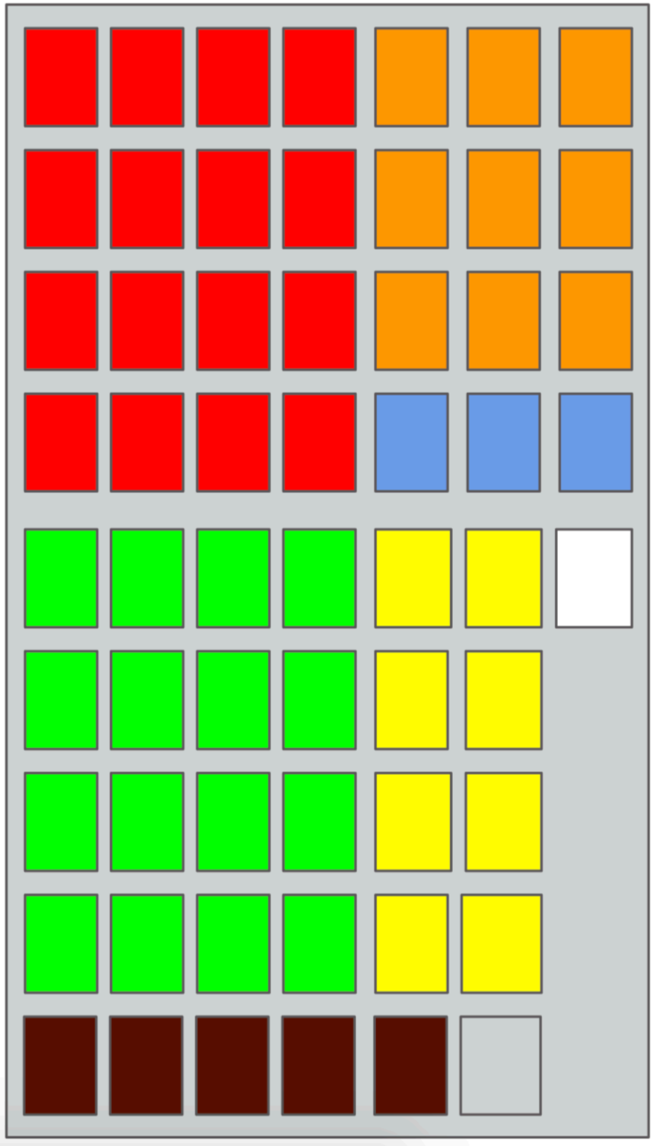


Offline analysis tools

- Tuples produced using **TupleTools** - creation and saving of variable branches for typical use cases eg. TupleToolTrackInfo
  - Very **easy to implement** but adds lots of **redundant branches** - can easily save 500+ variables
  - 500GB - 10TB of data for a single Run 1+2 analysis - nTuples tend to be only used for one analysis
  - **Redesign** of tools such that this redundancy is minimised
- LHCb collaboration uses a wide range of tools C++ /Python/ ROOT/ uproot/ numpy/ pandas/..
- Custom user environments (for use on distributed computing) limited by CVMFS distributions
  - Experimenting with providing analysts the ability to install **Conda environments** on CVMFS
  - **Singularity containers** (CERNVM) are used for running legacy applications on grid - looking to expand

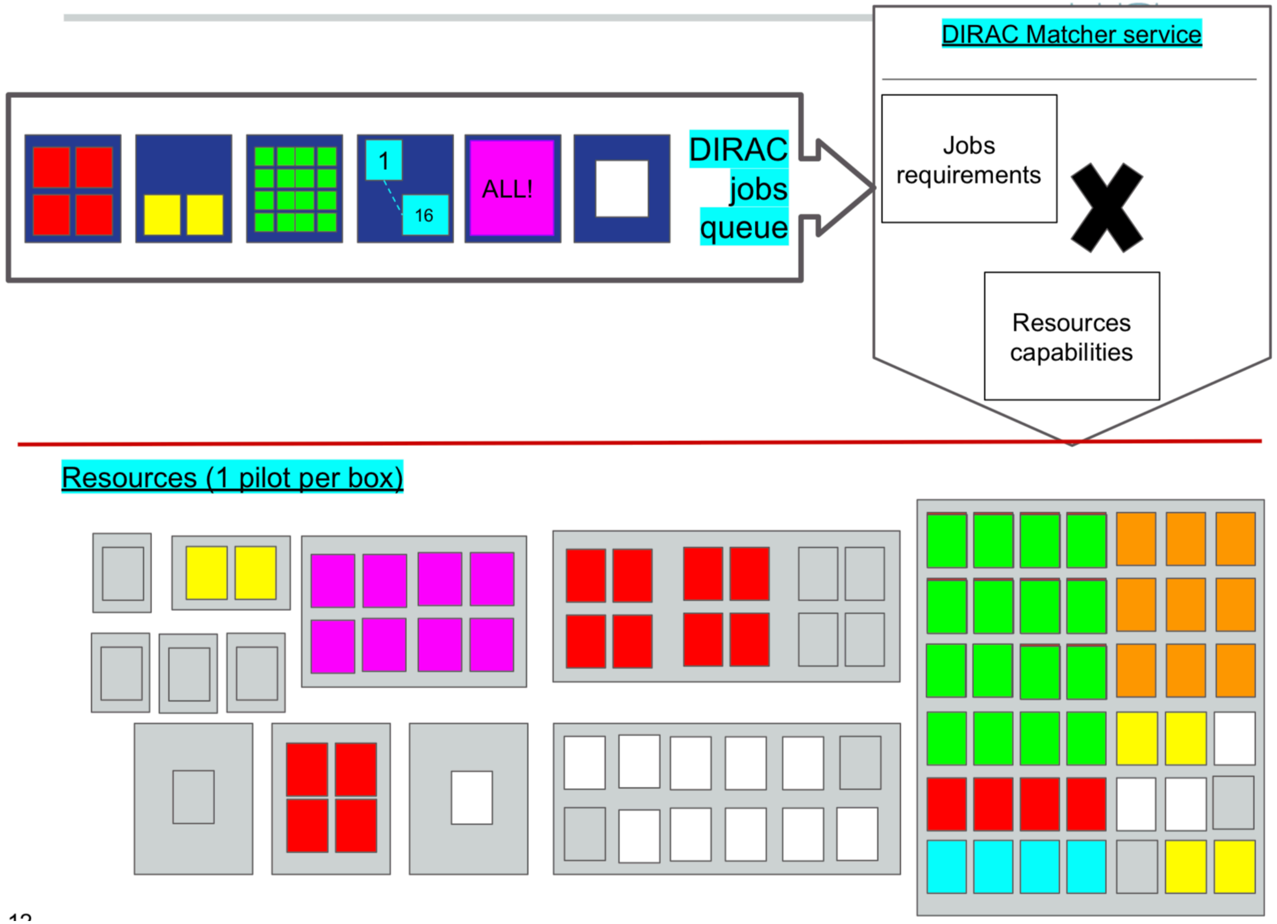
Distributed computing

- DIRAC is and remains the LHCb standard for workload and data management
- Current DIRAC design is expected to scale with Run3 workloads and data volumes
- Recent deployments to exploit many-core architectures
  - Use case: Marconi-A2 partition at CINECA, 68x4HT = 272 logical processors
  - DIRAC is able to "partition" the node for optimal memory and throughput
    - Using DIRAC "pool", an inner computing element
    - Parallel jobs matching



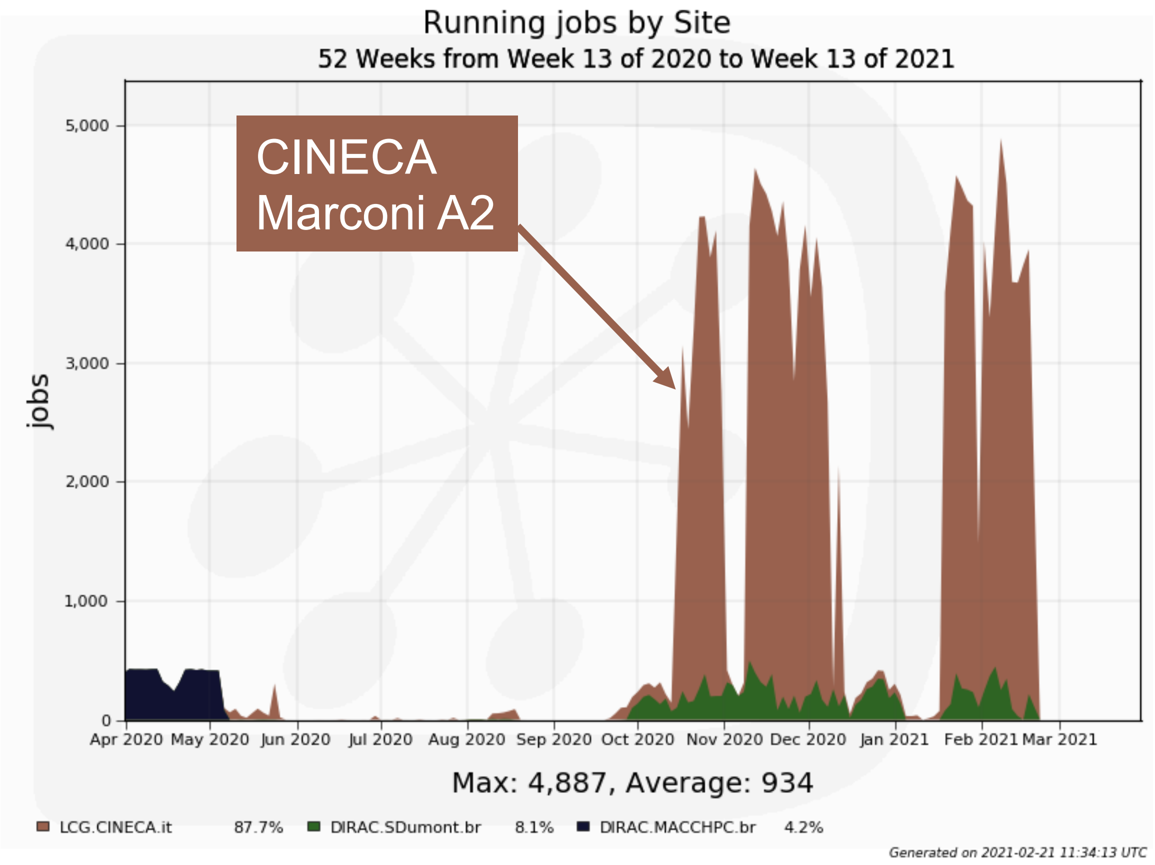
Distributed computing

- DIRAC is and remains the LHCb standard for workload and data management
- Current DIRAC design is expected to scale with Run3 workloads and data volumes
- Recent deployments to exploit many-core architectures
  - Use case: Marconi-A2 partition at CINECA, 68x4HT = 272 logical processors
  - DIRAC is able to “partition” the node for optimal memory and throughput
    - Using DIRAC “pool”, an inner computing element
    - Parallel jobs matching



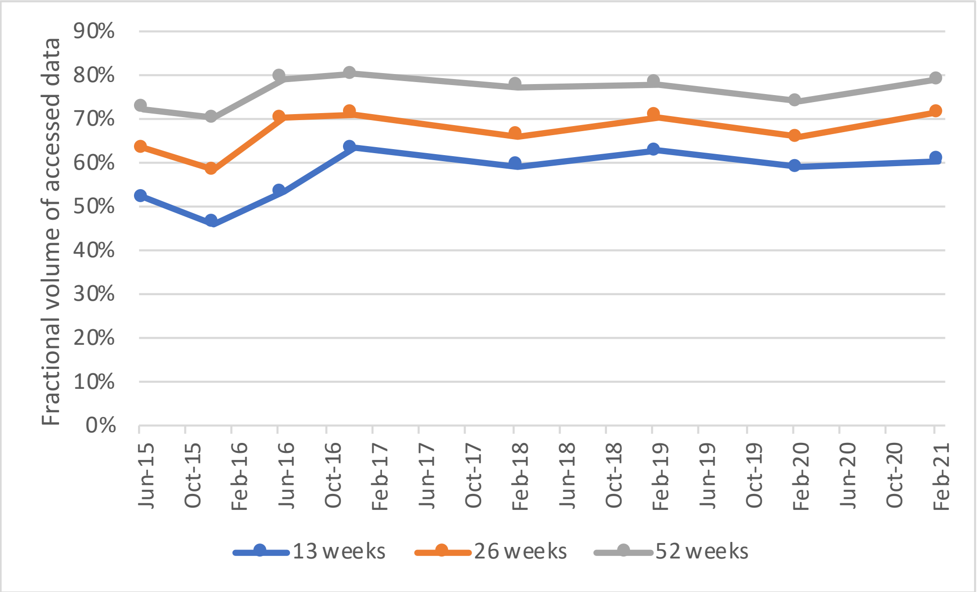
Distributed computing

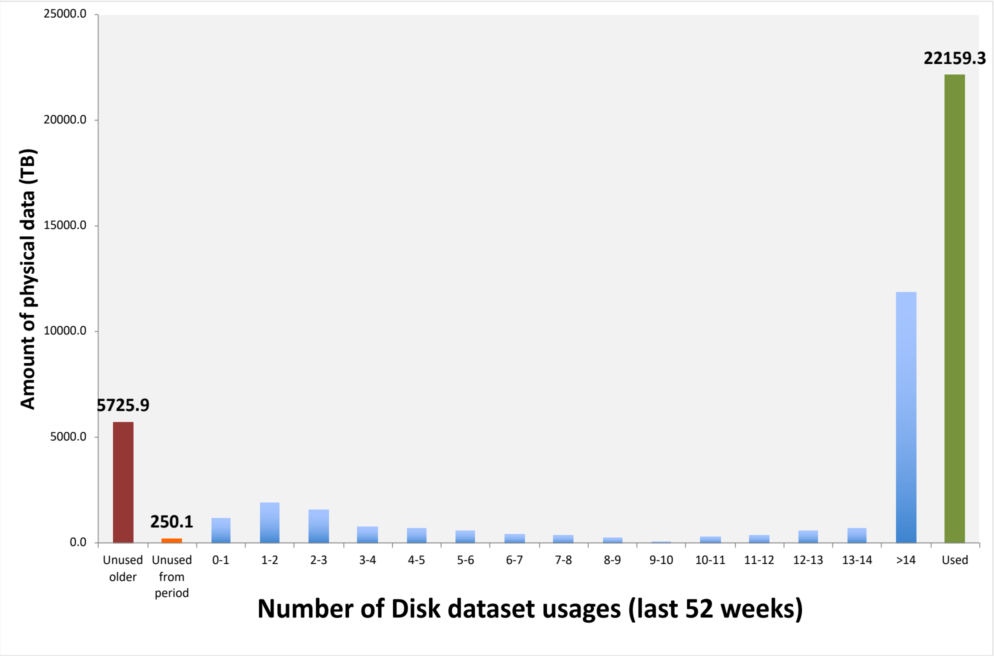
- DIRAC is and remains the LHCb standard for workload and data management
- Current DIRAC design is expected to scale with Run3 workloads and data volumes
- Recent deployments to exploit many-core architectures
  - Use case: Marconi-A2 partition at CINECA, 68x4HT = 272 logical processors
  - DIRAC is able to "partition" the node for optimal memory and throughput
    - Using DIRAC "pool", an inner computing element
    - Parallel jobs matching



Data management

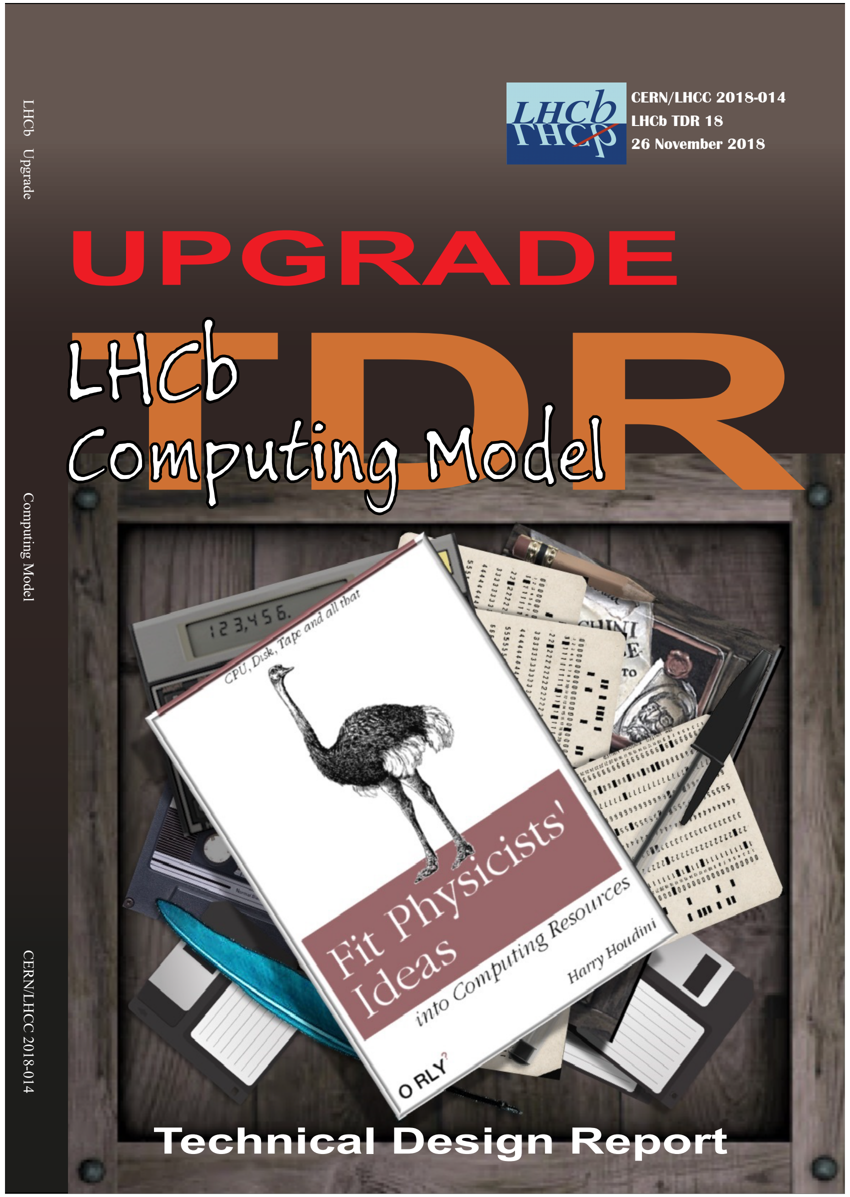
- Keep it simple
  - Reasonably small number of sites with storage
    - CERN + 7 Tier1 + ~15 Tier2 with disk
- Job matching based on where data is located, no remote access (except in case of failure), high efficiency
- No caches, no underlying data movements
- Static number of replicas
- Data popularity studies give reasonable utilization
- Following WLCG standards and their evolution for transfers:
  - FTS, TPC, …
- Not directly involved, but following DOMA activities





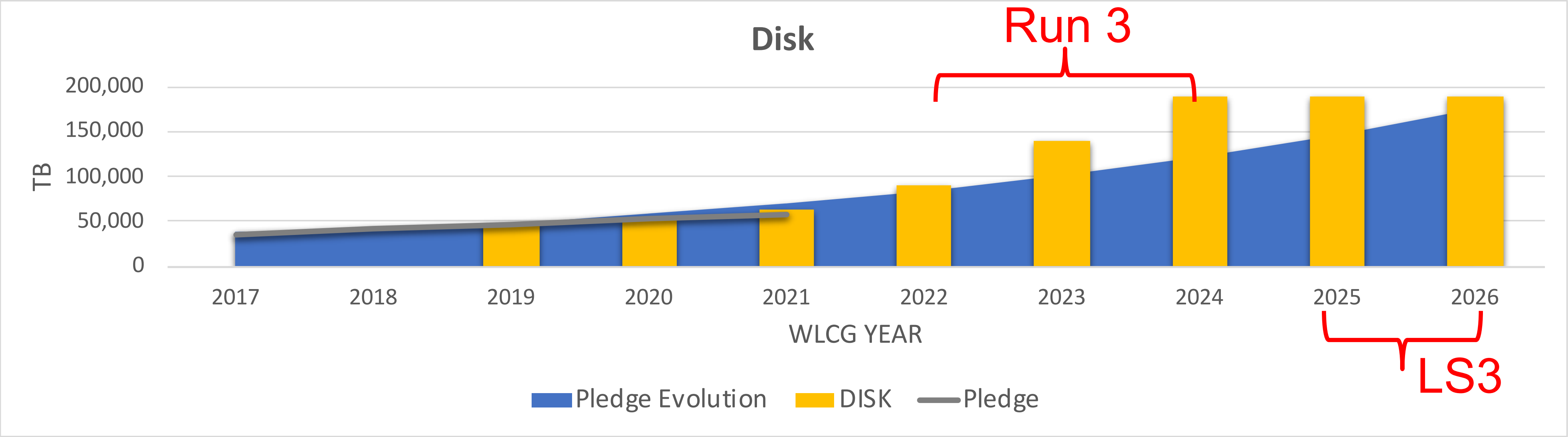
Run 3 Computing Model

- Concepts developed and implemented during Run 2 to become predominant
  - Split HLT → real-time alignment and calibration
  - TURBO stream for majority of physics program → RAW events discarded
  - FULL and CALIBRATION streams to insure flexibility → filter & slim offline
- Offline CPU computing needs dominated by simulation
  - Number of events to be simulated scales with luminosity
  - Simulation time per event scales with pileup
  - →CPU simulation explodes → need for faster simulations
- Offline storage driven by trigger output bandwidth
  - MC saved in μDST, so little impact on storage



LHCb-TDR-018

Storage requirements - disk

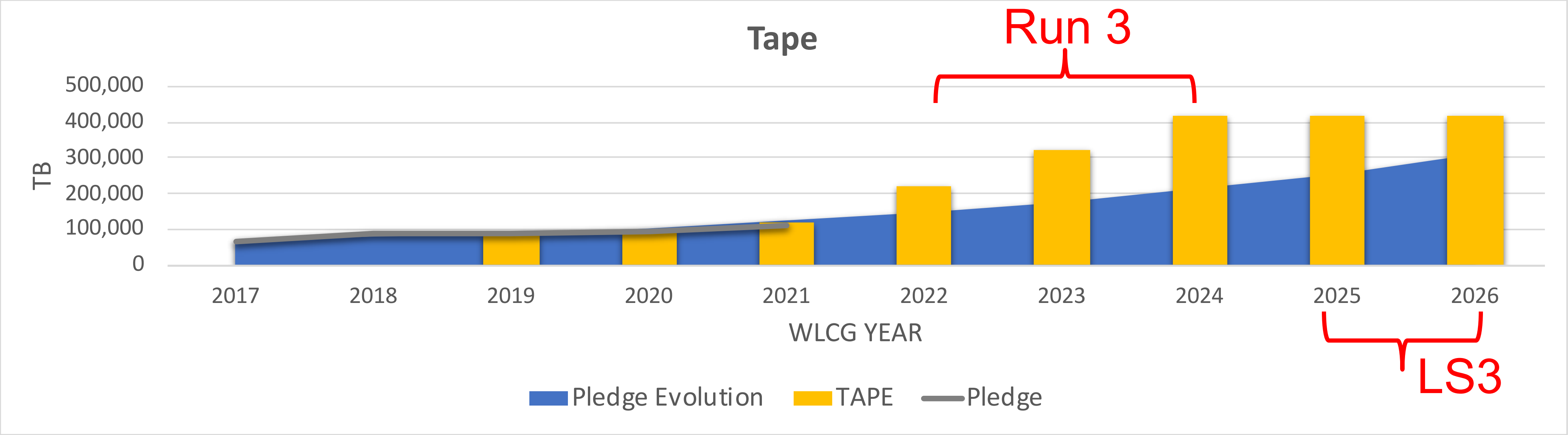


LHCb		2020	2021	2022	2023	2024	2025	2026	Average
DISK	PB	53	64	90	140	190	190	190	
	Increase		20%	41%	56%	36%	0%	0%	24%

- Pledge evolution assumes a "constant budget" model (+20% more every year)
- Given as a gauging term

- Max deviation from this model: x1.6
- In line with the model by the end of LS3

Storage requirements - tape



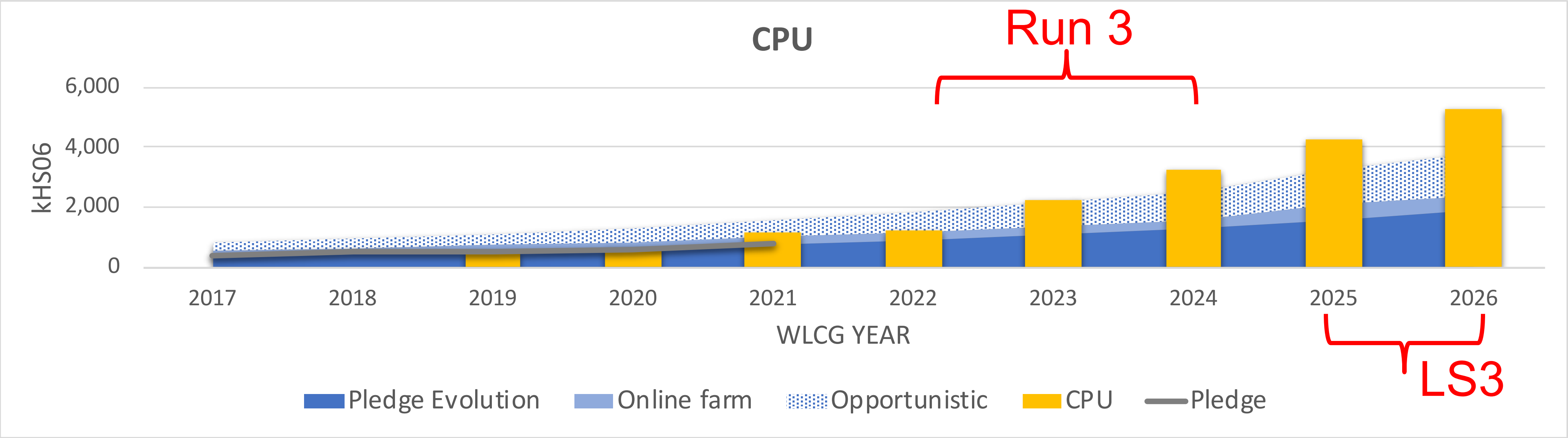
LHCb		2020	2021	2022	2023	2024	2025	2026	Average
TAPE	PB	92	120	220	320	420	420	420	
	Increase		30%	84%	45%	31%	0%	0%	29%

- Pledge evolution assumes a “constant budget” model (+20% more every year)
- Given as a gauging term

- Max deviation from this model: x1.8
- ~ in line with the model by the end of LS3

- N.B. tape is considered “cheap”

CPU requirements

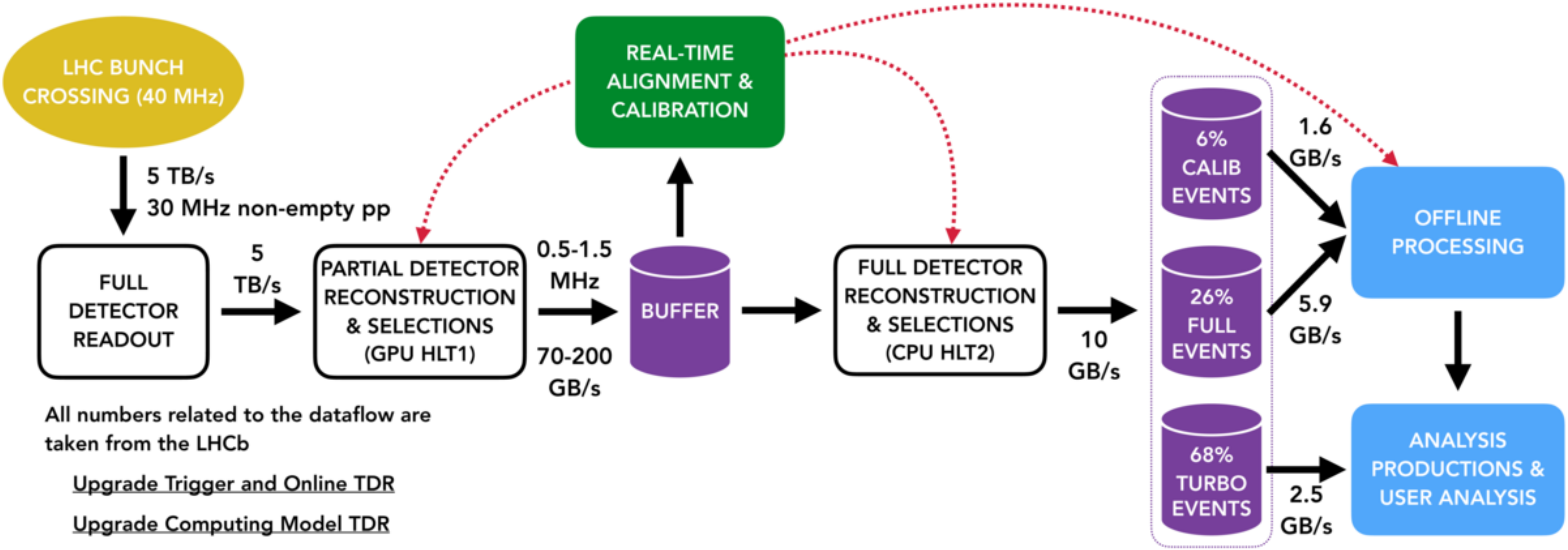


LHCb		2020	2021	2022	2023	2024	2025	2026	Average
CPU	kHS06	607	1170	1256	2256	3256	4256	5256	
	Increase		93%	7%	80%	44%	31%	23%	35%

- Pledge evolution assumes a “constant budget” model (+20% more every year)
- Given as a gauging term

- Max deviation from this model: x1.8
- Plan to use opportunistic resources, which are however not granted
- Online farm can be used opportunistically when idle (as we do now)

Outline



All numbers related to the dataflow are taken from the LHCb

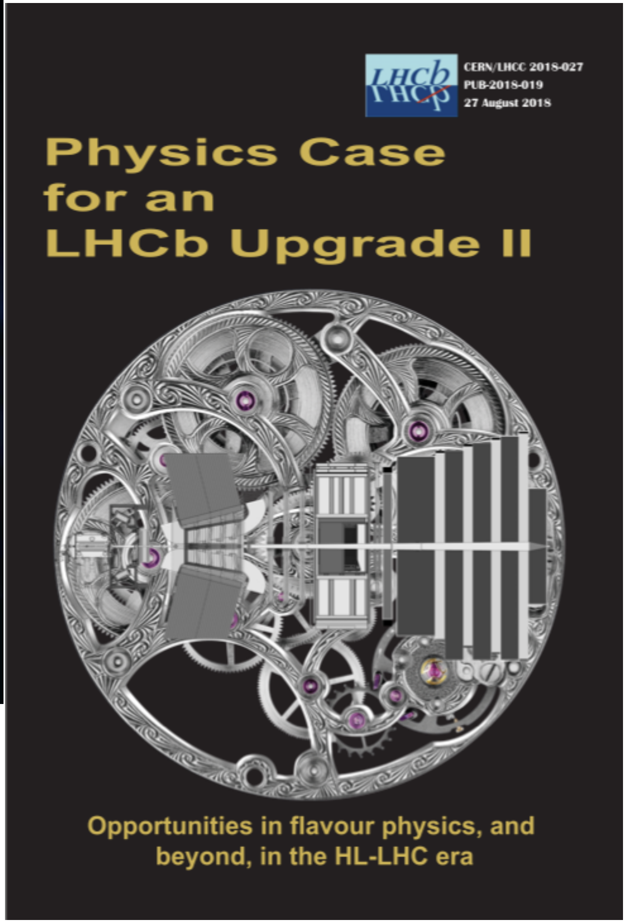
Upgrade Trigger and Online TDR

Upgrade Computing Model TDR

Towards a phase-II upgrade?

- The recent European Strategy on Particle Physics calls for full exploitation of the high-luminosity LHC
  - Unique opportunity to reach the ultimate precision in flavour physics observables
- R&D in the past couple of years towards a phase-II upgrade of LHCb with yet another factor 5 increase in luminosity (→10^{34} cm^{-2} s^{-1})
- Technologically challenging for detector technologies…
  - increased pileup, occupancies, radiation
  - Timing information (~10ps) needed to isolate signals

CERN-LHCC-2017-003
CERN-LHCC-2018-027



Towards a phase-II upgrade?

- …and software & computing
  - Aggressive data reduction by moving processing even closer to the real detector: e.g. real-time tracking with FPGAs
  - A simple extrapolation of Run3 computing model does not scale: resource requirements explode, R&D is needed to exploit new dimensions of computing