Computing & Software – Status and Future

David Rohr for the ALICE, ATLAS, CMS and LHCb Collaborations

ICHEP 2024, Prague

22.7.2024

drohr@cern.ch

Thanks a lot for everyone contributing and sending material!

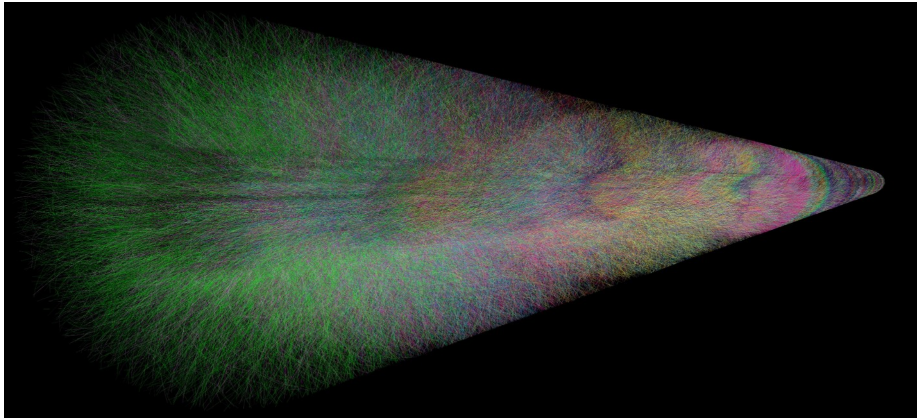
LHC Computing Challenges

See also **LHCP software plenary talk**:
more focus on simulation / analysis

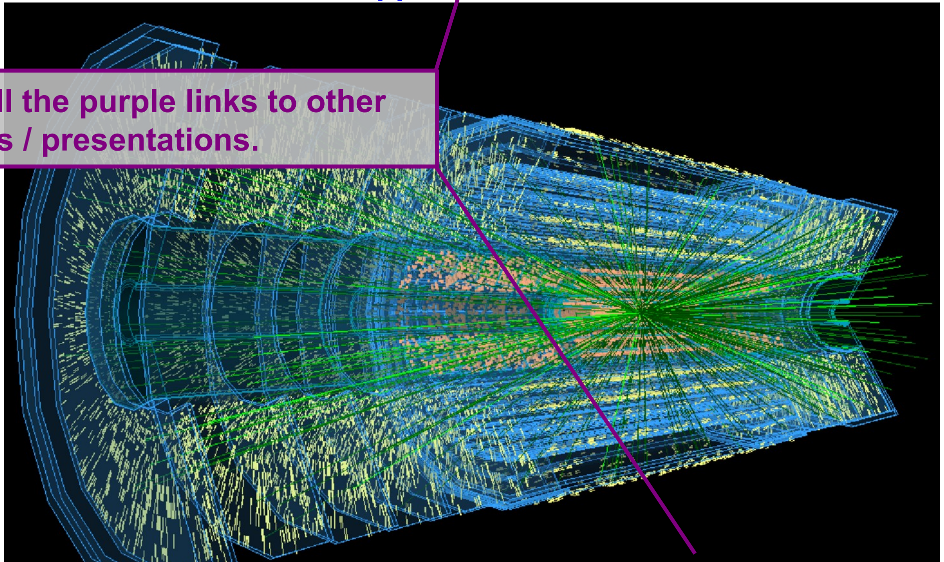
- **Event types and detectors might be different, but all LHC experiments are coping with high multiplicity event reconstruction at very high rates:**
  - **ALICE** has **time frames:** several high-multiplicity **Pb-Pb** collisions **overlapped**, shifted in time / z-axis.
    - Cannot disentangle collisions, track to vertex association only probabilistic.
    - **2.5** – **20 ms** of **continuous data**.
  - **ATLAS**, **CMS**, **LHCb** have mostly **pp events** with high **in-bunch pile up**.

Check out all the purple links to other talks / presentations.

ALICE Pb-Pb time frame 2ms of data



ATLAS pp collision illustration



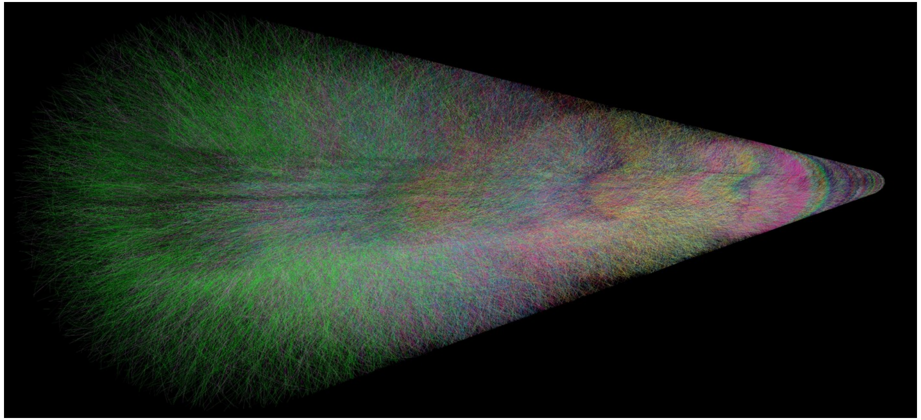
Dedicated **ATLAS Computing talk**

LHC Computing Challenges

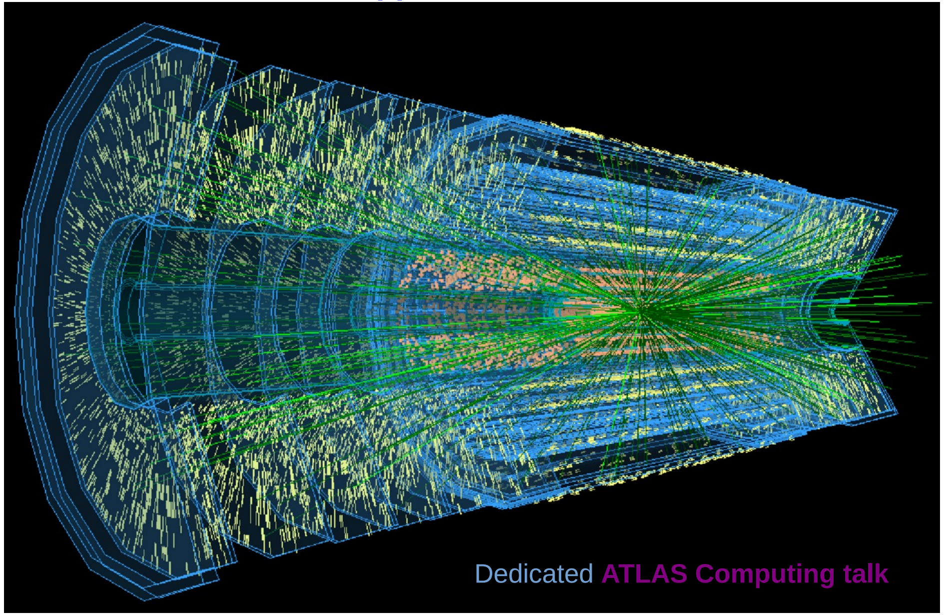
See also **LHCP software plenary talk**:
more focus on simulation / analysis

- **Event types and detectors might be different, but all LHC experiments are coping with high multiplicity event reconstruction at very high rates:**
  - **ALICE** has **time frames:** several high-multiplicity **Pb-Pb** collisions **overlapped**, shifted in time / z-axis.
    - Cannot disentangle collisions, track to vertex association only probabilistic.
    - **2.5** – **20 ms** of **continuous data**.
  - **ATLAS**, **CMS**, **LHCb** have mostly **pp events** with high **in-bunch pile up**.

ALICE Pb-Pb time frame 2ms of data



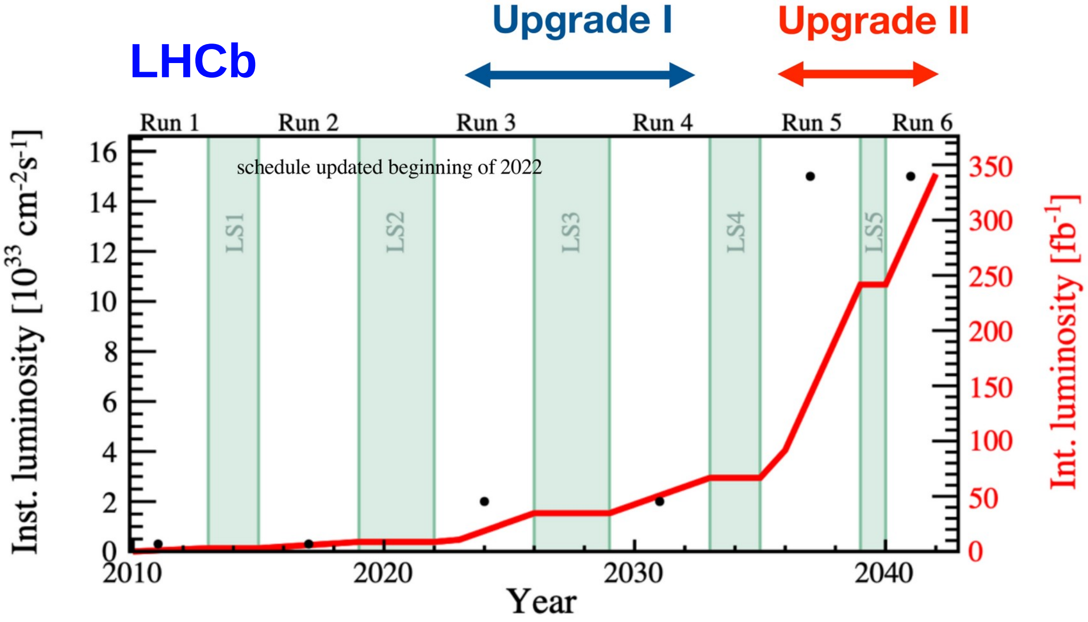
ATLAS pp collision illustration

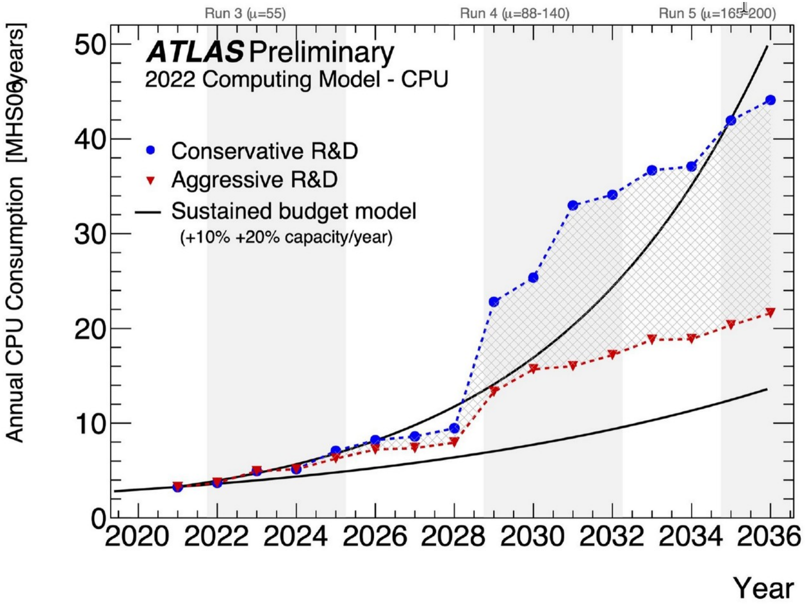


Dedicated **ATLAS Computing talk**

Recent and upcoming upgrades

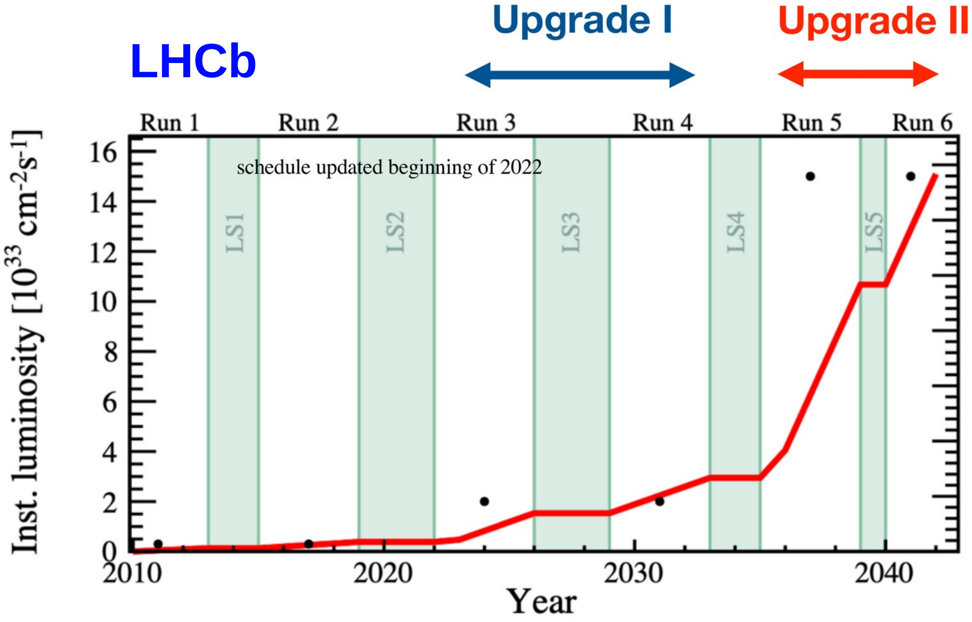
- **ALICE** and **LHCb** just went through their **Run 3 updates**, order of magnitude higher rate than Runs 1 / 2.
- **ATLAS** and **CMS** will see major upgrades in **LS3** for **HL-LHC**:
  - Luminosity $2\times10^{34}s^{-1}cm^{-2}$ to $\mathbf{7.5\times10^{34}s^{-1}cm^{-2}}$ **(μ = 200).**
- **LHCb Upgrade 2** and **ALICE 3** in the planning for **LS4**.
- **Consequentially, processing requirements will increase**.

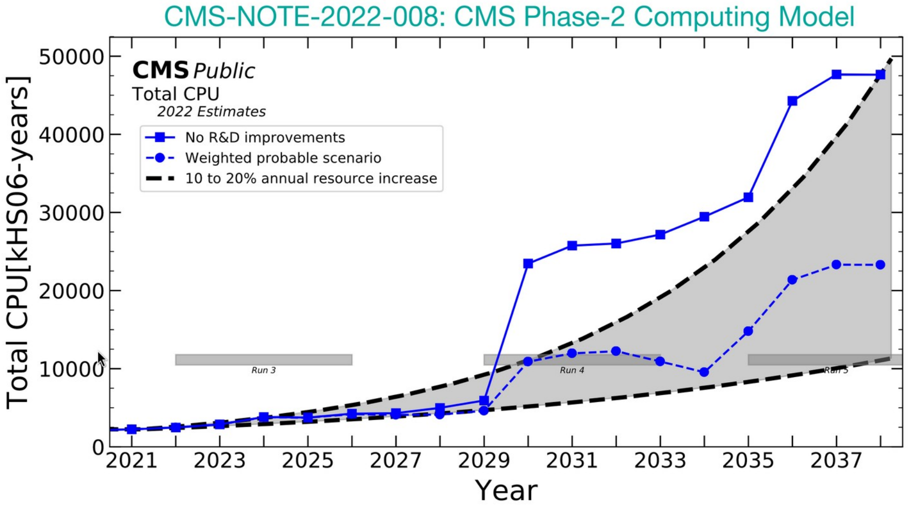




Recent and upcoming upgrades

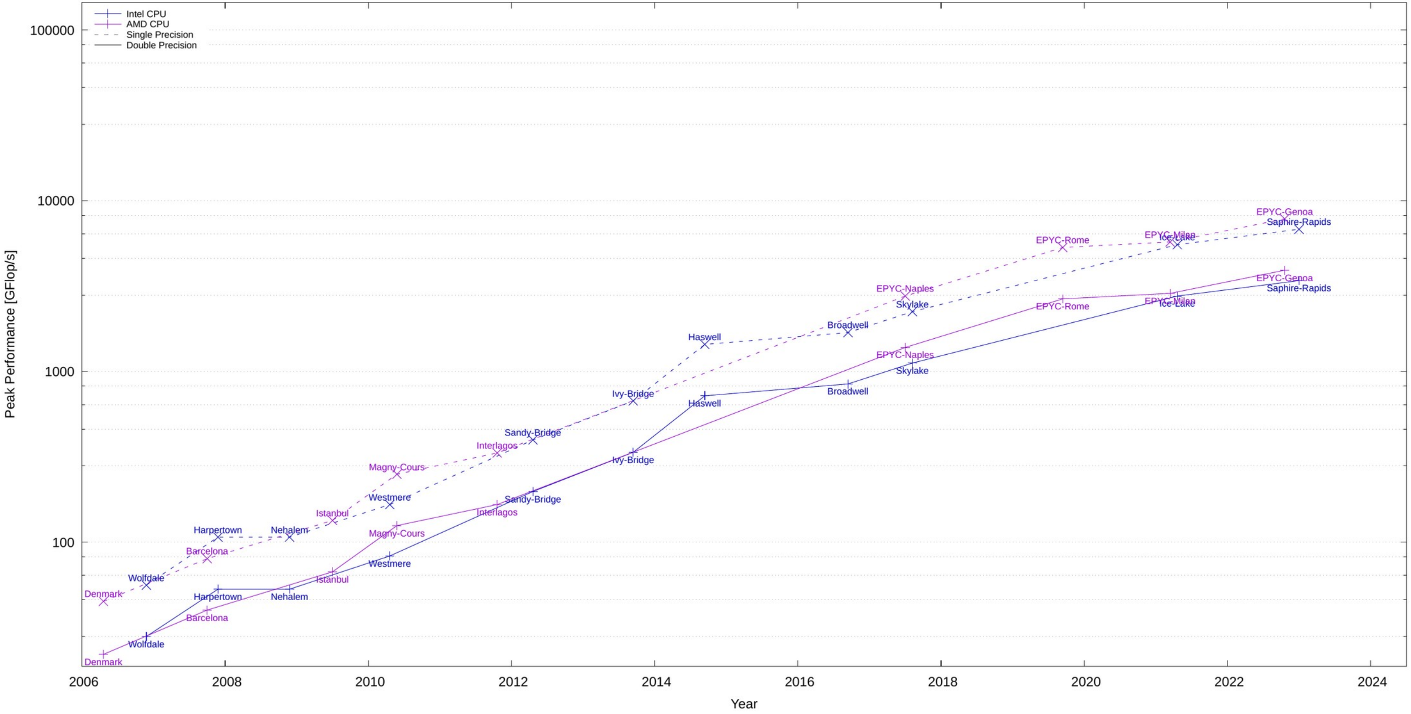
- **ALICE** and **LHCb** just went through their **Run 3 updates**, order of magnitude higher rate than Runs 1 / 2.
- **ATLAS** and **CMS** will see major upgrades in **LS3** for **HL-LHC**:
  - Luminosity $2 \times 10^{34} s^{-1} cm^{-2}$ to $\mathbf{7.5 \times 10^{34} s^{-1} cm^{-2}}$ **(μ = 200).**
- **LHCb Upgrade 2** and **ALICE 3** in the planning for **LS4**.
- **Consequentially, processing requirements will increase**.





CPU & GPU Performance Evolution

- **Performance of CPUs still almost exponentially increasing (FP32 & FP64), though a bit slower than some years ago.**



CPU & GPU Performance Evolution

- Performance of CPUs still almost exponentially increasing (FP32 & FP64), though a bit slower than some years ago.
- GPUs increasing with the same rate (FP32 shown only), almost constant offset.

CPU & GPU Performance Evolution

- **Need to use the potential of modern hardware and parallelization to cope with our ever increasing data rates.**
  - **High performance code** must consider **more aspects, hardware layers**, **features**, **particularities**...
  - ... **but**, **more and more complicated code** bases developed by **more and more people**.
  - → **We should simplify development**!

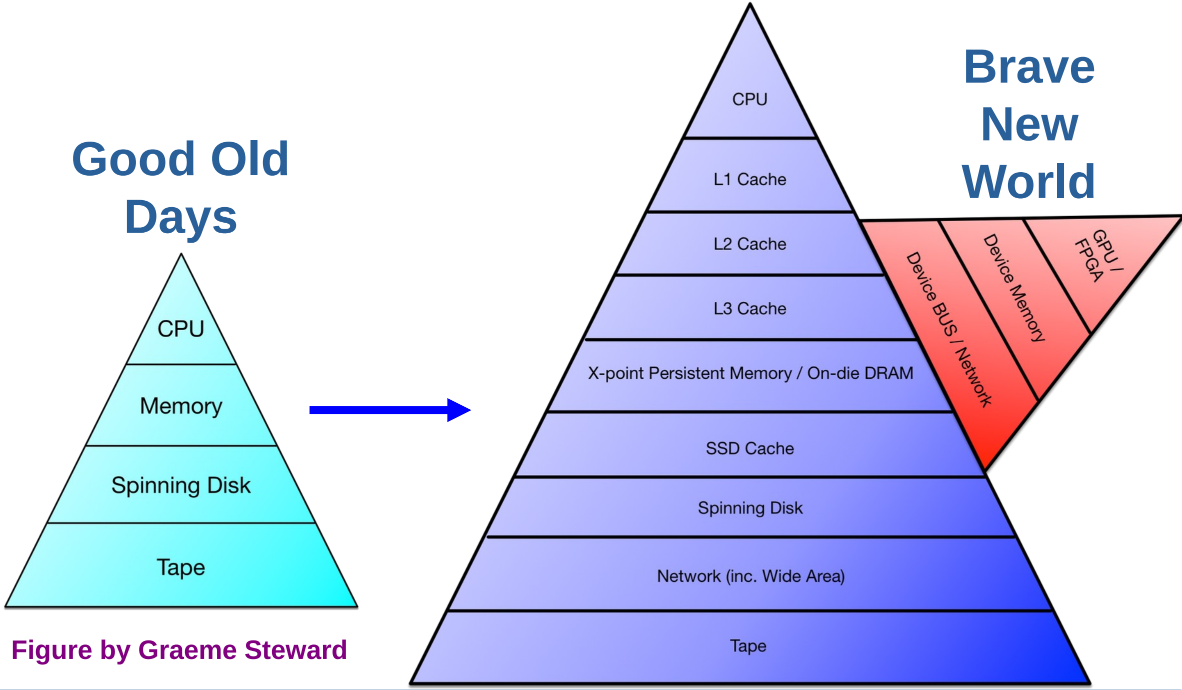
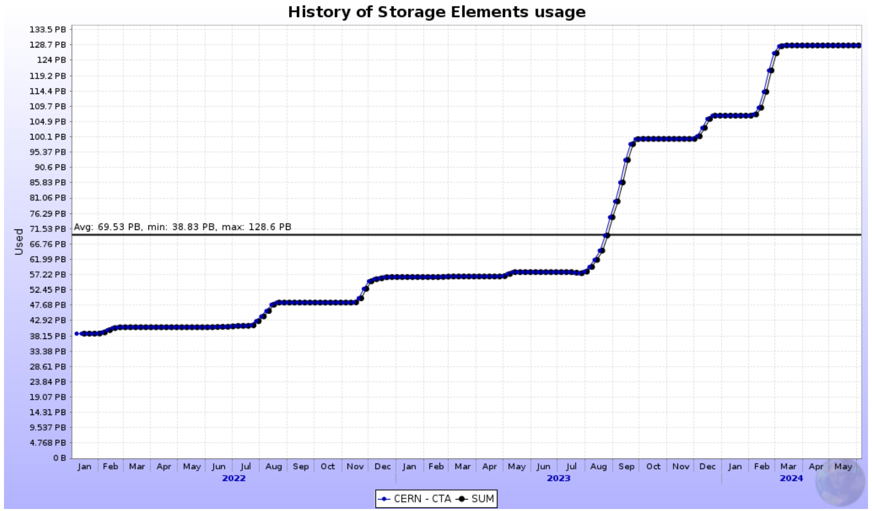


Figure by Graeme Steward

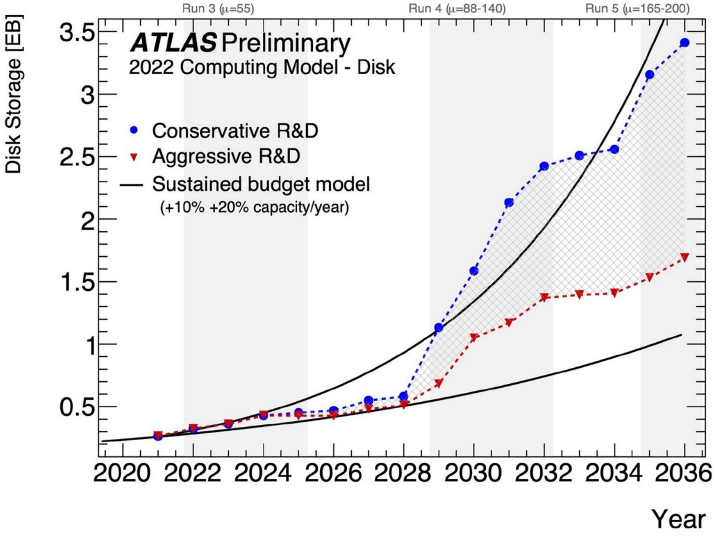
Storage Challenges for High Luminosity

- **Compute throughput is a hard problem, but reachable with GPUs, HPC (high performance computing) centers, etc.**
- **Storage / bandwidth is not scaling as fast, we can process more data than we can store....**
  - E.g. **ALICE** records **100 times more Pb-Pb collisions** than before.
  - → Reducing the data size online is paramount, needs triggering and compression.
  - **ATLAS**: expects **10-15%** gain with move to **RNTuple**, **CMS** does technical evolution to move to **RNTuple** as well.
  - **ATLAS data carousel** (warm tape storage by automatic disk – tape movement).

ALICE Aggregate Data Storage in Run 3



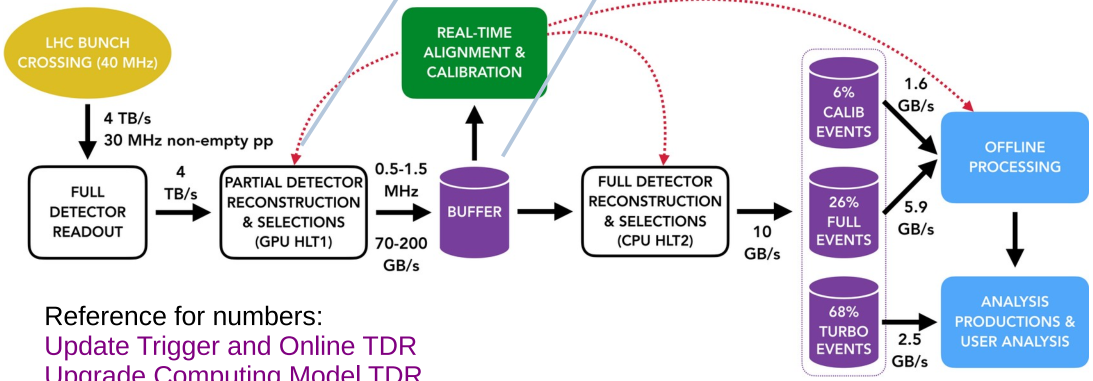
ATLAS Projected disk requirements



Online Computing & GPU / FPGA Usage

- Experiments going in different directions:
  - **ALICE** & **LHCb removed** the **hardware trigger**, doing **full online processing** with **GPUs**:
    - **ALICE** does **not trigger at all** (in **Pb-Pb**), runs **online calibration and compression.**
      - **~99%** of the reconstruction on **GPU** (more does not make sense).
      - **Delayed analysis trigger** for **pp**, sims **which data** from disk-buffergoes to **permanent storage**.
    - **LHCb** runs the **full HLT1 trigger** on **GPUs** with the **Allen** framework.
    - **ALICE** & **LHCb** use a local **disk buffer**, with the **online farm** processing data when **no beam** in LHC. (ALICE: asynchronous (offline) reprocessing, LHCb: final processing and triggering)

LHCb



Note: corresponding plot of ALICE in backup.

Reference for numbers:
Update Trigger and Online TDR
Upgrade Computing Model TDR

N.B. All experiments are / were using the HLT farms for GRID jobs when idle since Run 2.

Online Computing & GPU / FPGA Usage

- Experiments going in different directions:
  - **ALICE** & **LHCb removed** the **hardware trigger**, doing **full online processing** with **GPUs**:
    - **ALICE** does **not trigger at all** (in **Pb-Pb**), runs **online calibration and compression**.
      - **~99%** of the reconstruction on **GPU** (more does not make sense).
      - **Delayed analysis trigger** for **pp**, sims **which data** from disk-buffergoes to **permanent storage**.
    - **LHCb** runs the **full HLT1 trigger** on **GPUs** with the **Allen** framework.
    - **ALICE** & **LHCb** use a local **disk buffer**, with the **online farm** processing data when **no beam** in LHC. (ALICE: asynchronous (offline) reprocessing, LHCb: final processing and triggering)

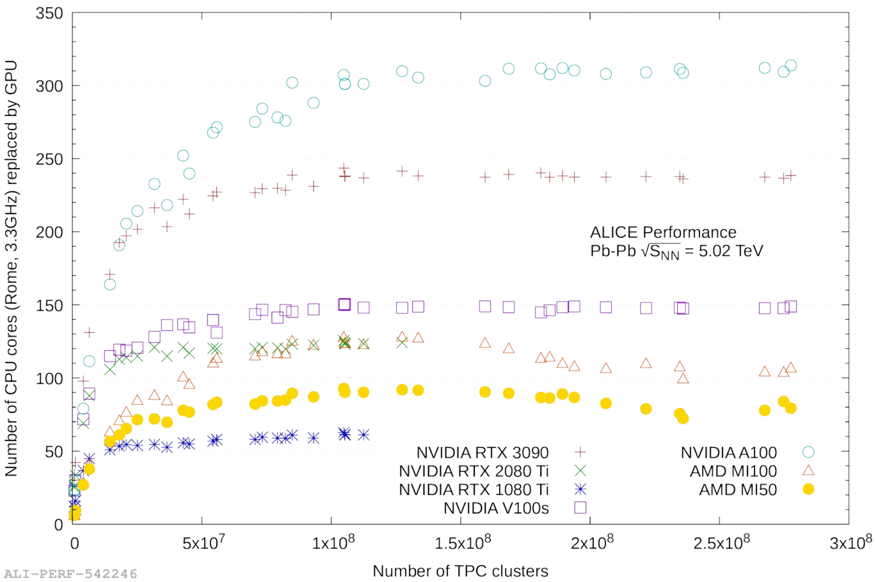
  - **ATLAS** & **CMS** still need a **hardware trigger** for even **higher HL-LHC rates**. Talk on CMS L1 Trigger
    - Classical usage of FPGAs in low-level hardware triggers inevitable.
    - **CMS** is using **CPU+GPU** in their **HLT** farm today.
    - **ATLAS** will decide on the hardware for the **HL-LHC HLT** in **2025**.
      - **GPU support** with memory management and infrastructure available, **build system** now supports CUDA, HIP, Sycl, Alpaka, Kokkos

- **CMS Scouting** stream / **LHCb Turbo** stream: save **HLT objects** at **higher rate** on top of raw data, stores many more events.

- **For GPU usage in offline / simulation see later.**

ALICE GPU Online processing performance

- **Performance of Alice O2 software on different GPU models and compared to CPU.**



ALICE used GPUs in the HLT during Runs 1 and 2
In Run 3: full online processing on GPU

Without GPUs, more than 3000 64-core servers would be needed for online processing!
Would be prohibitively expensive!

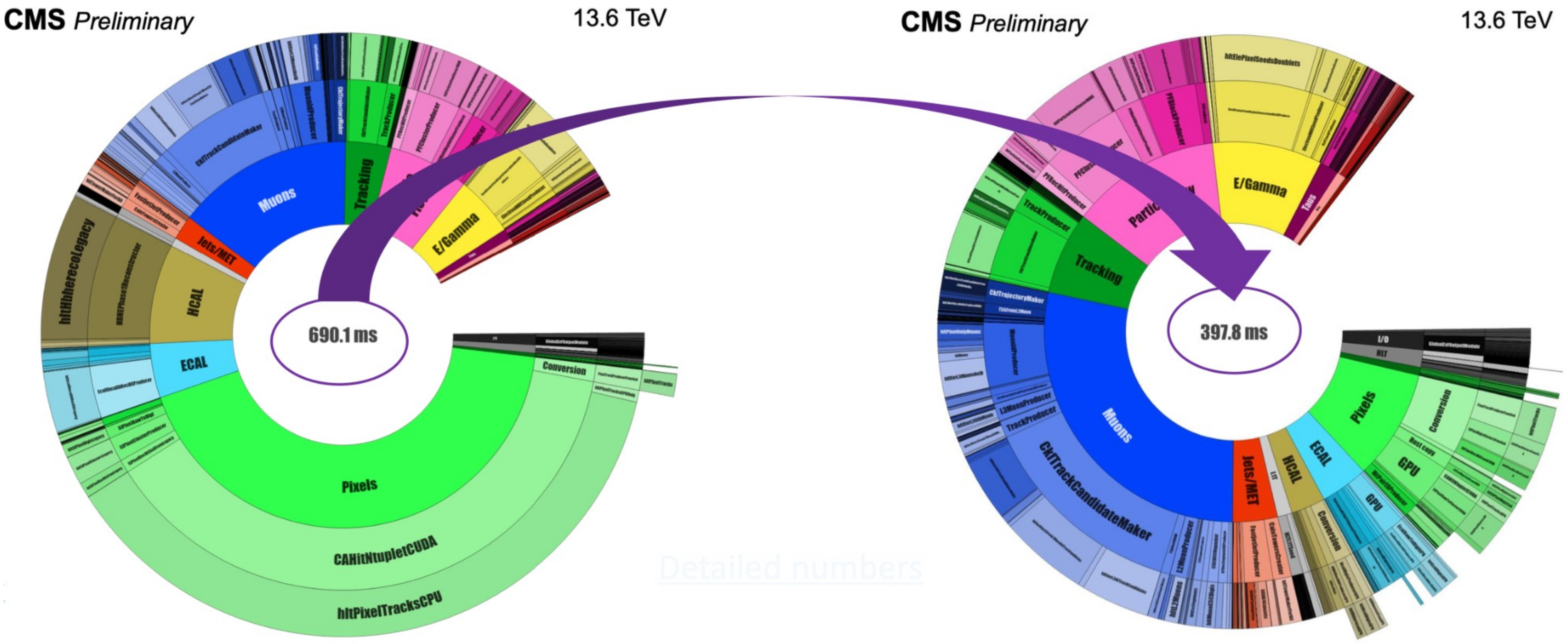
- ALICE runs the **TPC tracking** (**99%** of online reco, **95%** of online processing) on GPU.
- CPUs can easily handle the rest (QC, etc.), no need to offload it.
  - But makes sense for offline, see later.
- Same source code for CPU and GPU.
  - → Same tracking efficiency, resolution, etc.

- **ALICE uses 2240 MI50 and 560 MI100 GPUs in the online farm.**
- **MI50 GPU replaces ~80 AMD Rome CPU cores in online reconstruction.**
  - **~55 CPU cores** in **offline** reconstruction (different mix of algorithms).

Detailed talk on ALICE GPU Reconstruction

CMS GPU Performance Improvements for Online

- **CMS** is using **GPU** enabled reconstruction in the **High Level Trigger** since the start of Run 3.
- The execution time per event was reduced by **~40%**.

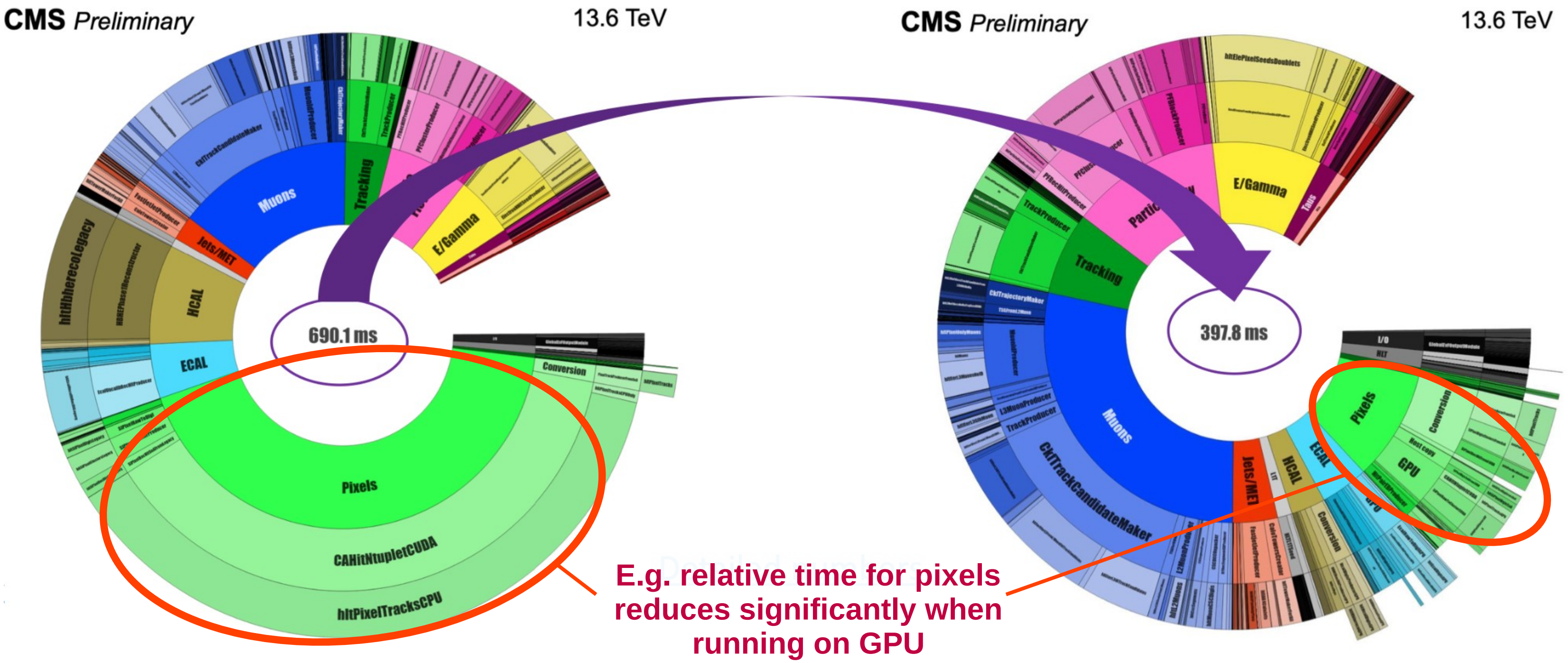


Average time per event for CPU Only Configuration

Average time per event for CPU + GPU Configuration

CMS GPU Performance Improvements for Online

- **CMS** is using **GPU** enabled reconstruction in the **High Level Trigger** since the start of Run 3.
- The execution time per event was reduced by **~40%**.

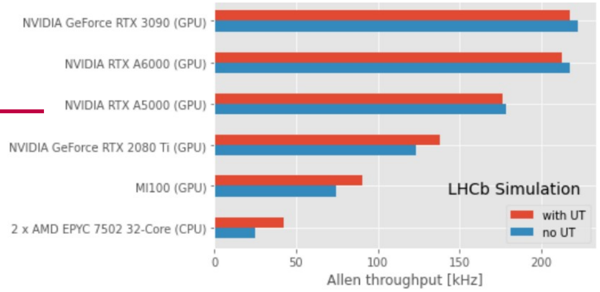


Average time per event for CPU Only Configuration

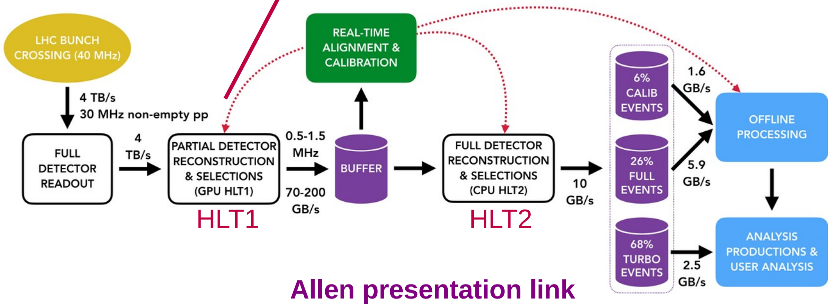
Average time per event for CPU + GPU Configuration

LHCb GPU processing in the Allen framework

- **Full HLT1 trigger on NVIDIA RTX A5000 GPUs with the Allen framework.**
  - Supports multiple GPU models / vendors.
    Decision to use GPUs driven by **global cost benefit analysis**:
    - Otherwise a fast expensive event builder network was needed.
    - Additional bonus: GPUs are available when not running HLT1.
      (**CPU / GPU Comparison document link**)



GPU speedup in the HLT1

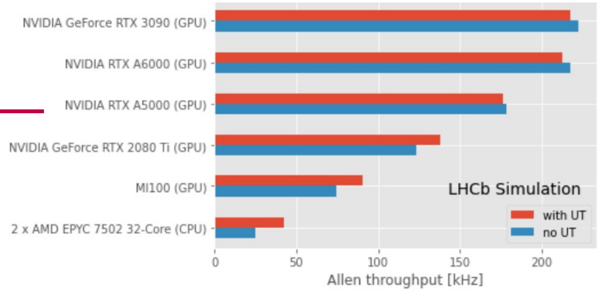


Allen presentation link

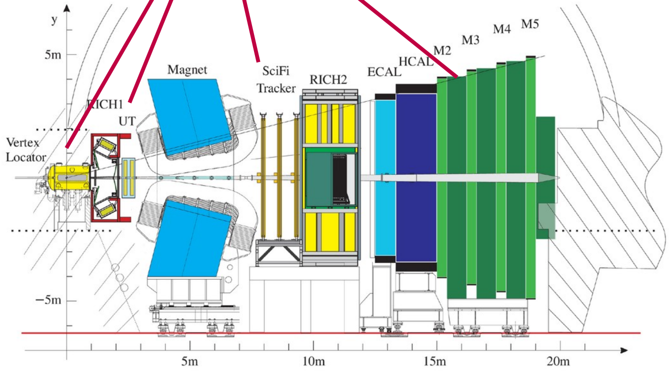
LHCb GPU processing in the Allen framework

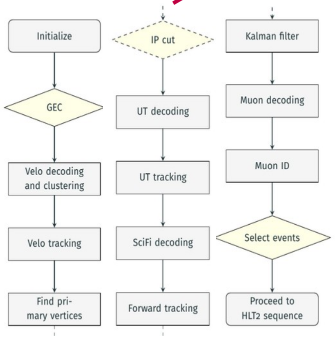
Talk on
LHCb trigger

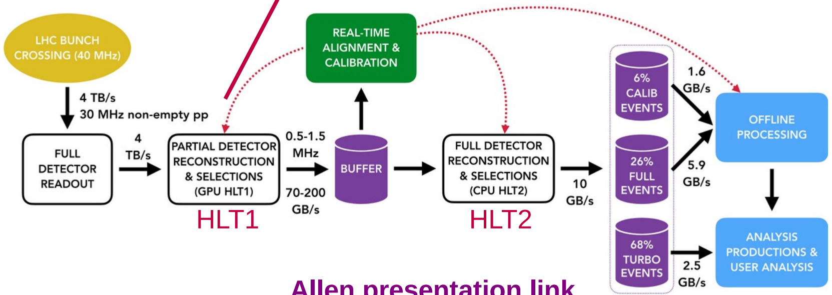
- **Full HLT1 trigger on NVIDIA RTX A5000 GPUs with the Allen framework.**
  - Supports multiple GPU models / vendors.
    Decision to use GPUs driven by **global cost benefit analysis**:
    - Otherwise a fast expensive event builder network was needed.
    - Additional bonus: GPUs are available when not running HLT1.
  - HLT1 involves decoding, clustering and track reconstruction of all tracking detectors, all on GPU (Velo, Upstream Tracker, SciFi Tracker, Muons, Electrons, Calorimeters) and then PV finder and full trigger algorithms.



GPU speedup in the HLT1





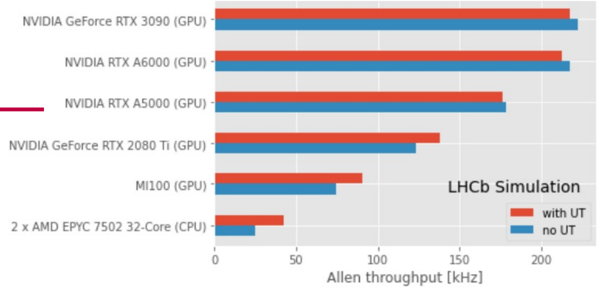


Allen presentation link

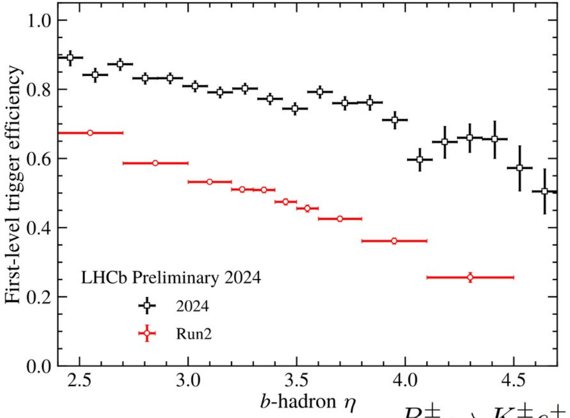
LHCb GPU processing in the Allen framework

Talk on
LHCb trigger

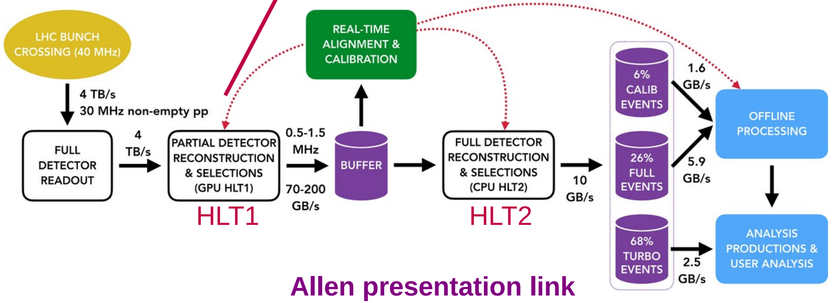
- **Full HLT1 trigger on NVIDIA RTX A5000 GPUs with the Allen framework.**
  - Supports multiple GPU models / vendors.
    Decision to use GPUs driven by **global cost benefit analysis**:
    - Otherwise a fast expensive event builder network was needed.
    - Additional bonus: GPUs are available when not running HLT1.
  - HLT1 involves decoding, clustering and track reconstruction of all tracking detectors, all on GPU (Velo, Upstream Tracker, SciFi Tracker, Muons, Electrons, Calorimeters) and then PV finder and full trigger algorithms.
  - Significantly better trigger efficiency than in Run 2, e.g. (**reference**):



GPU speedup in the HLT1



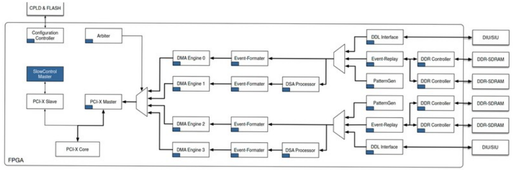
$B^{\pm} \to K^{\pm} e^{+} e^{-}$ channel

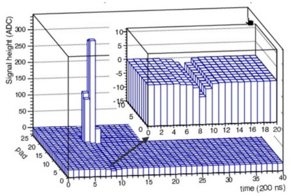


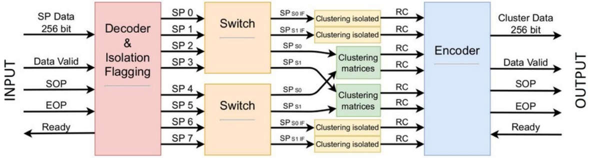
Allen presentation link

Hardware accelerator usage in online computing

- Several efforts to use **FPGAs**, etc. for more elaborate triggering / processing online.
  - **ALICE** used it for **TPC clustering** in Runs 1 & 2 (link).
  - In **Run 3**, **ALICE** runs some **corrections** (**ion tail and common mode**) for the **TPC** in the **FPGA** user logic of the readout servers.
  - **LHCb** uses **FPGAs** for the **clustering** for the **Velo** (talk from CHEP2023).
  - **FPGAs** used a lot in the **low-level triggers** (**ATLAS** and **CMS** L1 trigger).
  - Efforts by **ATLAS** and other experiments to use **FPGAs** for **tracking**.



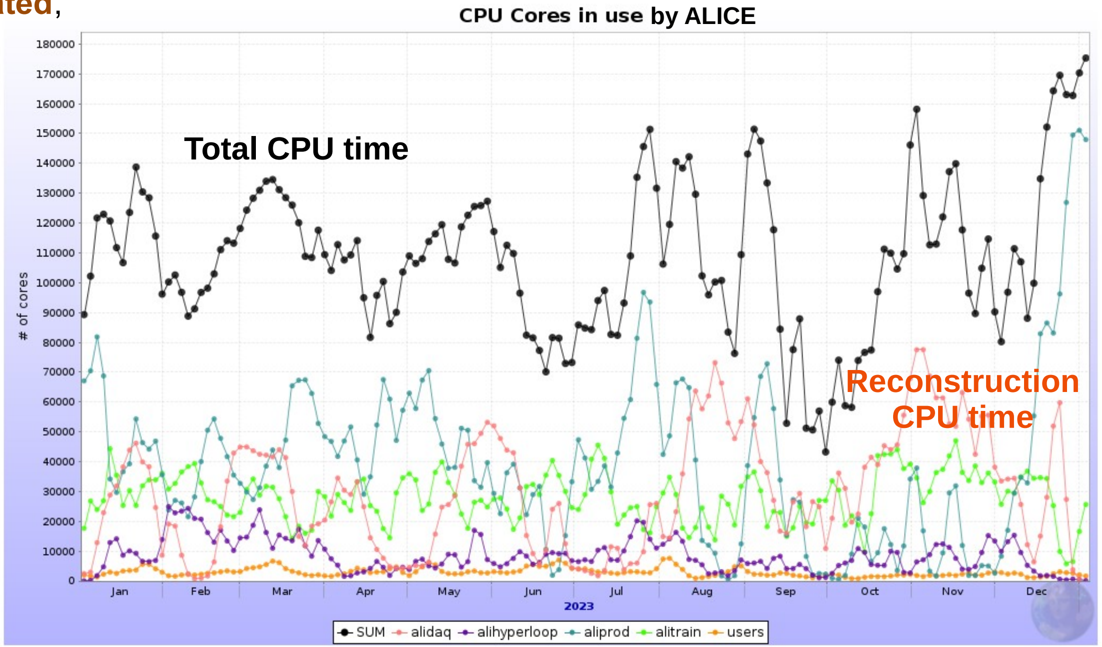




- But **not always successful**, long complex developments, **software** is just **simpler**. For instance:
  - **ALICE dropped** its **FPGA TPC clustering** efforts in Run 3 since FPGA was too full, doing clustering on GPUs.
  - **ATLAS** attempted **associative-memory-based tracking** with custom ASICs, **not followed up** any more.

Usage of GPU beyond online reconstruction

- **Online compute farms** need a lot of **on-site processing power** for data-taking, **GPUs** are one solution.
- On the GRID: Reconstruction (**alidaq**) not dominant, simulation (**aliprod**) and analysis similar or higher
  - E.g. **simulation** is **70%** in **ATLAS GRID** time.
- **Focusing GPU efforts on reconstruction insufficient** to solve the overall compute problem – but a start!
  - **GPU** usage in **simulation more complicated**, most time goes into libraries (**Geant4**, **Generators**)
    - **GPU-accelerated digitization** could be the **first step** for **ALICE** (since fully under our control).
    - More later.
- **Offline reconstruction** on **GPUs**:
  - **ALICE running** on GPUs since **2023**.
  - **CMS** ported code to **Alpaka** as **preparation for the future**. (online and offline share much code).
  - **ATLAS** foresees **opportunistic GPU** usage first in the **future**.



ALICE Reconstruction Compute Time Overview

- **Relative contribution of processing steps varies greatly:**
  - **ALICE online** processing fully **dominated** by **TPC** (largest detector).
  - **Offline processing** more heterogeneous and depend on multiplicity, **other bottlenecks** with **TPC** on **GPU**.
  - Complicates GPU offloading, since **no single bottleneck**.

Online processing (50 kHz Pb-Pb, MC, no QA / calib)

Processing step	% of time
TPC Processing (Tracking, Clustering, Compression)	99.37 %
EMCAL Processing	0.20 %
ITS Processing (Clustering + Tracking)	0.10 %
TPC Entropy Encoder	0.10 %
ITS-TPC Matching	0.09 %
MFT Processing	0.02 %
TOF Processing	0.01 %
TOF Global Matching	0.01 %
PHOS / CPV Entropy Coder	0.01 %
ITS Entropy Coder	0.01 %
Rest	0.08 %

Offline processing (650 kHz pp, 2022, no Calorimeters)

Processing step	% of time
TPC Processing (Tracking)	61.41 %
ITS TPC Matching	6.13 %
MCH Clusterization	6.13 %
TPC Entropy Decoder	4.65 %
ITS Tracking	4.16 %
TOF Matching	4.12 %
TRD Tracking	3.95 %
MCH Tracking	2.02 %
AOD Production	0.88 %
Quality Control	4.00 %
Rest	2.32 %

Offline processing (47 kHz Pb-Pb, 2023)

Processing step	% of time
TPC Processing (Tracking)	52.39 %
ITS Tracking	12.65 %
Secondary Vertexing	8.97 %
MCH	5.28 %
TRD Tracking	4.39 %
TOF Matching	2.85 %
ITS-TPC Matching	2.64 %
Entropy Decoding	2.63 %
AOD Production	1.72 %
Quality Control	1.64 %
Rest	4.84 %

ALICE Reconstruction Compute Time Overview

- **Relative contribution of processing steps varies greatly:**
  - **ALICE online** processing fully **dominated** by **TPC** (largest detector).
  - **Offline processing** more heterogeneous and depend on multiplicity, **other bottlenecks** with **TPC** on **GPU**.
  - Complicates GPU offloading, since **no single bottleneck**.

Online processing
(50 kHz Pb-Pb, MC, no QA / calib)

Processing step	% of time
TPC Processing (Tracking, Clustering, Compression)	99.37 %
EMCAL Processing	0.20 %
ITS Processing (Clustering + Tracking)	0.10 %
TPC Entropy Encoder	0.10 %
ITS-TPC Matching	0.09 %
MFT Processing	0.02 %
TOF Processing	0.01 %
TOF Global Matching	0.01 %
PHOS / CPV Entropy Coder	0.01 %
ITS Entropy Coder	0.01 %
Rest	0.08 %

Offline processing
(650 kHz pp, 2022, no Calorimeters)

Processing step	% of time
TPC Processing (Tracking)	61.41 %
ITS TPC Matching	6.13 %
MCH Clusterization	6.13 %
TPC Entropy Decoder	4.65 %
ITS Tracking	4.16 %
TOF Matching	4.12 %
TRD Tracking	3.95 %
MCH Tracking	2.02 %
AOD Production	0.88 %
Quality Control	4.00 %
Rest	2.32 %

Offline processing
(47 kHz Pb-Pb, 2023)

Processing step	% of time
TPC Processing (Tracking)	52.39 %
ITS Tracking	12.65 %
Secondary Vertexing	8.97 %
MCH	5.28 %
TRD Tracking	4.39 %
TOF Matching	2.85 %
ITS-TPC Matching	2.64 %
Entropy Decoding	2.63 %
AOD Production	1.72 %
Quality Control	1.64 %
Rest	4.84 %

Running on GPU in baseline scenario

1st phase of GPU offload (today) – code needed for online

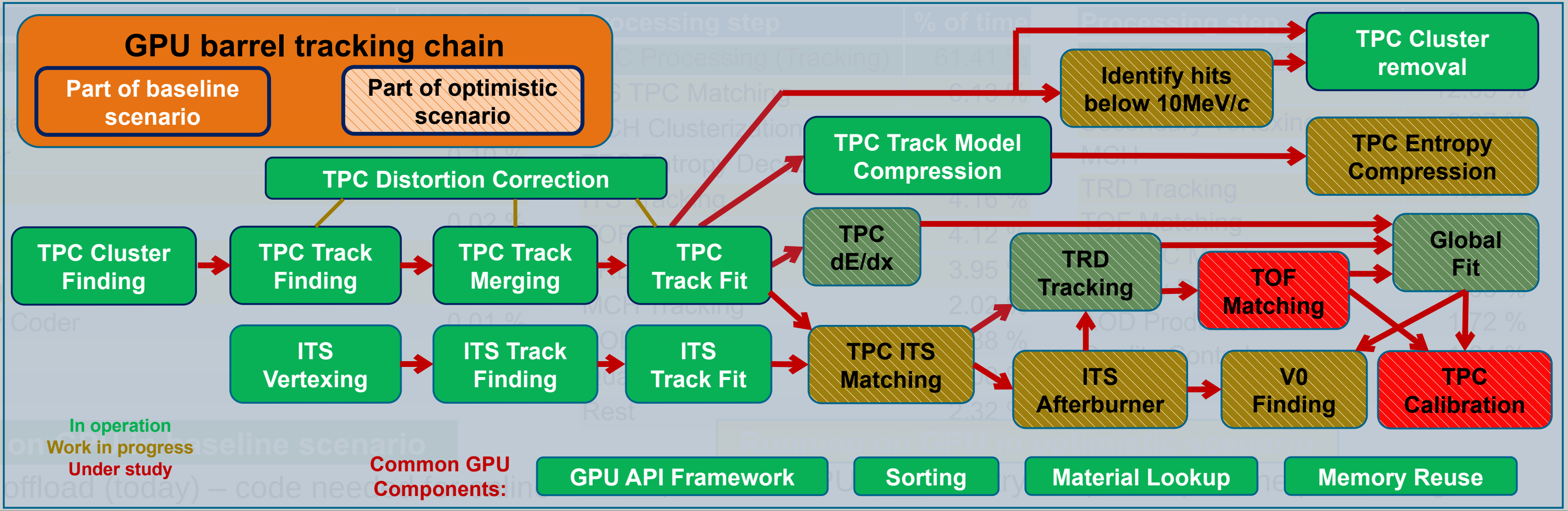
Running on GPU in optimistic scenario

2nd phase of GPU offload – try to speed up offline processing

ALICE – Offloading more tasks for offline reco

Central barrel tracking chosen as best candidate for optimistic scenario for offline reconstruction:

- Mandatory **baseline scenario** includes everything that must run on the GPU during online reconstruction.
- **Optimistic scenario** includes everything related to the barrel tracking.
- Since these are consecutive processing steps, all can run on the GPU at once without data transfer forth and back.



Expected GPU speedup in Offline Reconstruction

Running on GPU in optimistic scenario. Offloading **80%** should give **5x** speedup.

Running on GPU in baseline scenario. Offloading **60%** should give **2.5x** speedup.

Online processing
(50 kHz Pb-Pb, MC, no QA / calib)

Processing step	% of time
TPC Processing (Tracking, Clustering, Compression)	99.37 %
EMCAL Processing	0.20 %
ITS Processing (Clustering + Tracking)	0.10 %
TPC Entropy Encoder	0.10 %
ITS-TPC Matching	0.09 %
MFT Processing	0.02 %
TOF Processing	0.01 %
TOF Global Matching	0.01 %
PHOS / CPV Entropy Coder	0.01 %
ITS Entropy Coder	0.01 %
Rest	0.08 %

Offline processing
(650 kHz pp, 2022, no Calorimeters)

Processing step	% of time
TPC Processing (Tracking)	61.41 %
ITS TPC Matching	6.13 %
MCH Clusterization	6.13 %
TPC Entropy Decoder	4.65 %
ITS Tracking	4.16 %
TOF Matching	4.12 %
TRD Tracking	3.95 %
MCH Tracking	2.02 %
AOD Production	0.88 %
Quality Control	4.00 %
Rest	2.32 %

Offline processing
(47 kHz Pb-Pb, 2023)

Processing step	% of time
TPC Processing (Tracking)	52.39 %
ITS Tracking	12.65 %
Secondary Vertexing	8.97 %
MCH	5.28 %
TRD Tracking	4.39 %
TOF Matching	2.85 %
ITS-TPC Matching	2.64 %
Entropy Decoding	2.63 %
AOD Production	1.72 %
Quality Control	1.64 %
Rest	4.84 %

Running on GPU in baseline scenario

1st phase of GPU offload (today) – code needed for online

Running on GPU in optimistic scenario

2nd phase of GPU offload – try to speed up offline processing

Real speedup in offline reconstruction

- **ALICE online farm is split in online and offline partition.**
  - **Totally separate, online and offline jobs cannot share a node.**
  - **Nodes are moved between partitions as needed.**
- **Before comparing GPUs and CPUs, optimized different CPU and GPU configurations to find the fastest.**
  - E.g. can split into 8 * 16 core queue v.s. 16 * 8 core queue, or in 8 * 1 GPU v.s. 2 * 4 GPUs (**details in backup**).

Real speedup in offline reconstruction

- **ALICE online farm is split in online and offline partition.**
  - **Totally separate, online and offline jobs cannot share a node.**
  - **Nodes are moved between partitions as needed.**
- **Before comparing GPUs and CPUs, optimized different CPU and GPU configurations to find the fastest.**
  - E.g. can split into 8 * 16 core queue v.s. 16 * 8 core queue, or in 8 * 1 GPU v.s. 2 * 4 GPUs (**details in backup**).
- **Processing time per time-frame (neglecting overhead at start).**
  - Measured with 2023 software, 1st phase (baseline scenario):

Configuration (2022 pp, 650 kHz, 7400 collisions per TF)	Time per TF (1 instance)	Time per TF (full server)
CPU 8 cores	76.91s	4.81s
CPU 16 cores	34.18s	**4.27s**
1 GPU + 16 CPU cores	14.60s	1.83s
1 NUMA domain (4 GPUs + 64 cores)	3.5s	**1.70s**

Factor 2.51
Matches expected factor 2.5

Real speedup in offline reconstruction

- **ALICE online farm is split in online and offline partition.**
  - **Totally separate, online and offline jobs cannot share a node.**
  - **Nodes are moved between partitions as needed.**
- **Before comparing GPUs and CPUs, optimized different CPU and GPU configurations to find the fastest.**
  - E.g. can split into 8 * 16 core queue v.s. 16 * 8 core queue, or in 8 * 1 GPU v.s. 2 * 4 GPUs (**details in backup**).
- **Processing time per time-frame (neglecting overhead at start).**
  - Measured with 2023 software, 1st phase (baseline scenario):

Configuration (2022 pp, 650 kHz, 7400 collisions per TF)	Time per TF (1 instance)	Time per TF (full server)
CPU 8 cores	76.91s	4.81s
CPU 16 cores	34.18s	**4.27s**
1 GPU + 16 CPU cores	14.60s	1.83s
1 NUMA domain (4 GPUs + 64 cores)	3.5s	**1.70s**

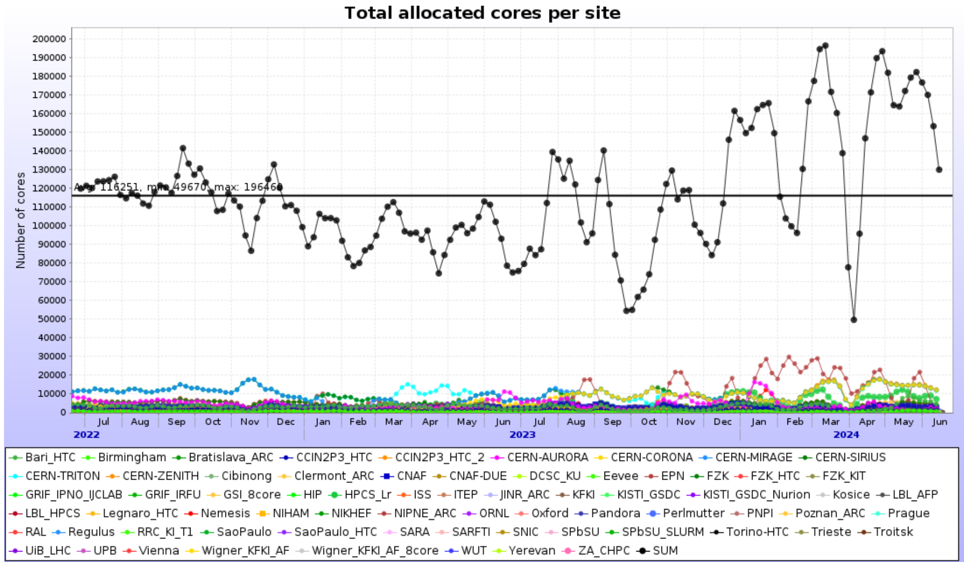
Factor 2.51
Matches expected factor 2.5

- **Work on 2nd phase with optimistic scenario ongoing gradually**:
  - In 2024 already gained **1.5% to 5%** by offloading **TPC data decoding** (as master thesis) (**LHCP talk**).

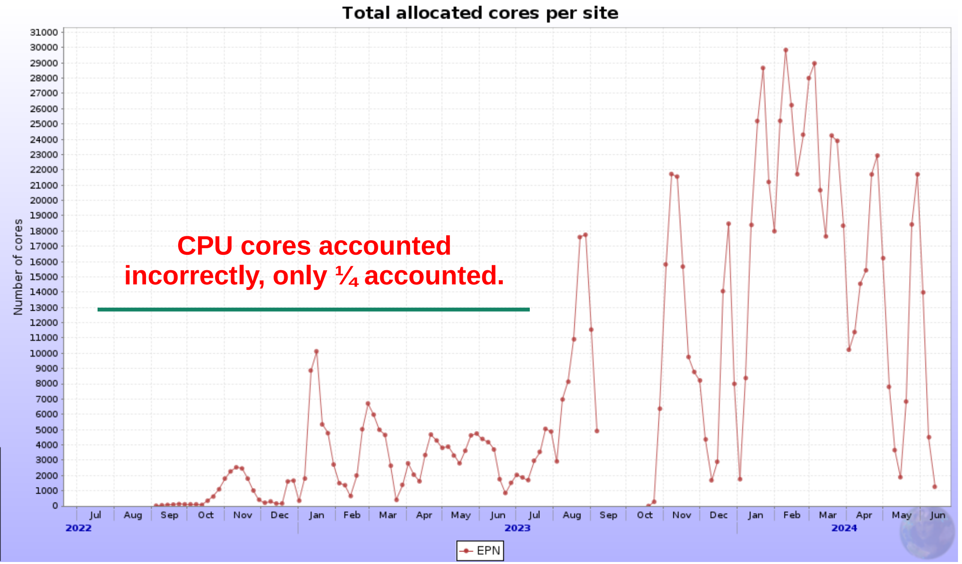
ALICE standard and GPU GRID jobs

- **Since 2023**, ALICE is running a significant fraction of the **offline reconstruction** GRID jobs on the **online farm** with **GPUs**.
- Usage fluctuating, due to online farm usage for data taking : **no GRID** jobs during **Pb-Pb**, **many jobs** during **YETS**.
- In peak phases, EPN online computing farm provides **~20% of ALICE GRID CPU cores**, and **GPUs in addition**.
- Working to use **GPUs** also on **other GRID sites**, e.g. **NERSC** with **NVIDIA GPUs**.

Total number of cores used in GRID jobs

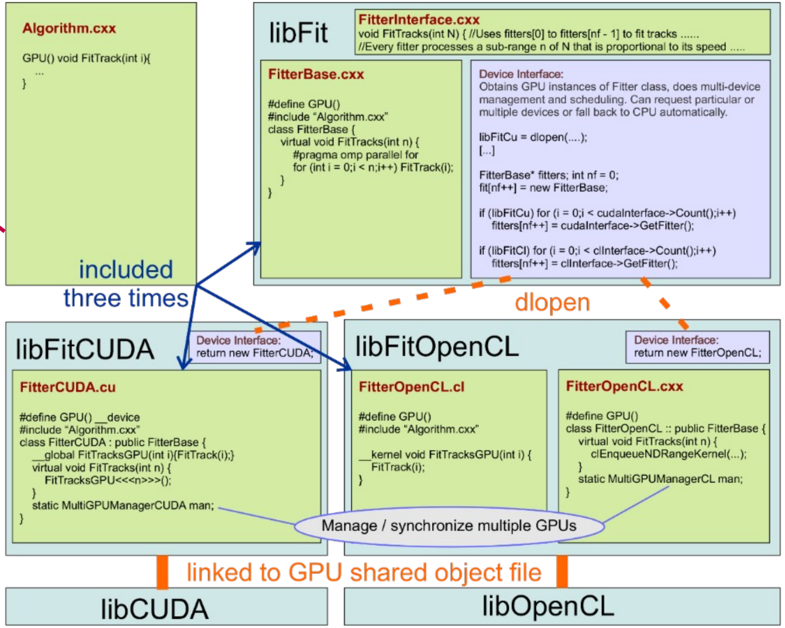


Number of cores in GPU-enabled online farm



Generic vendor-independent GPU implementations

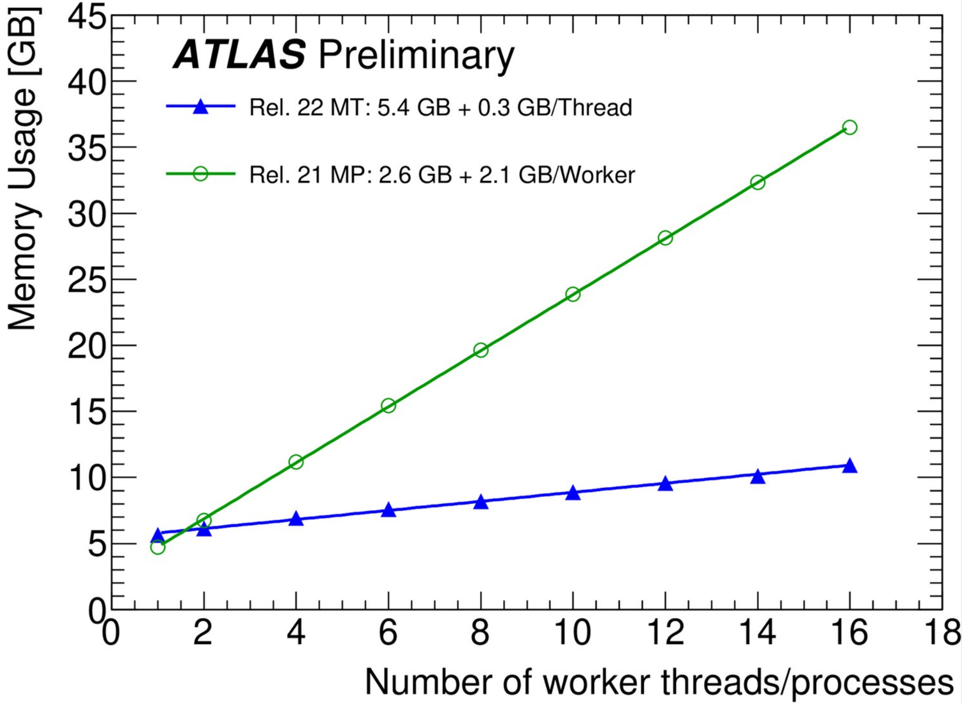
- **All experiments work to have code not tightly coupled to a single vendor.**
- **Generic GPU programming avoids code duplication.**
  - **ALICE** has **custom framework** (developed in Runs 1 & 2).
    - Dispatches **generic GPU** code to: **HIP**, **CUDA**, **OpenCL**, and **OpenMP for CPUs**.
    - *Why not go for e.g. Alpaka? Solution is working and not much time for studies.*
  - **LHCb Allen** supports and runs on **NVIDIA CUDA GPU** today, ports (not fully maintained) exist for **AMD HIP, Intel oneAPI**.
  - **CMS** has ported its GPU code to **Alpaka** in 2024, supporting **different GPU models**.
  - **ATLAS** will **decide** on Run 4 hardware in **2025**, but build system already supports **CUDA**, **HIP**, **Alpaka**, **Sycl**.



Paper reference

Challenges of processing frameworks

- **Multi-threading** essential for throughput / memory footprint:
  - Multi-threading support one of the most important **ATLAS** improvements for **Run 3**.
  - **CMS** running multi-threaded for **simulation**, data **processing** and **analysis** since **Run 2**.



Challenges of processing frameworks

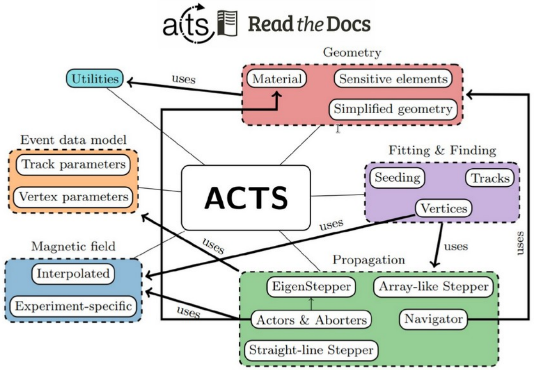
- **Multi-threading** essential for throughput / memory footprint:
  - Multi-threading support one of the most important **ATLAS** improvements for **Run 3**.
  - **CMS** running multi-threaded for **simulation**, data **processing** and **analysis** since **Run 2**.
  - **ALICE** had **monolithic** framework for **Runs 1** / **2**, new **message-passing multi-process** framework for **Run 3**.
    - **Same framework** for **simulation**, **reconstruction** (online and offline) and **analysis**.
    - Multiple levels of parallelism (horizontal (different data) and vertical (same data)).
    - **Multi-threading** supported **internally by algorithms**, if thread-safe and not interacting with the framework.
    - **Work in progress**: **automatic multi-threading** of algorithms for Run 3.
  - **My personal wishlist**:
    - Seamless support for multi-threading and multi-process of thread-safe well-defined algorithms with inputs and outputs.
    - Multi-threading for efficient operation, multi-process for better debugging, and any combination possible.
    - Probably asking for too much...
- **Scheduling** for **many-core architectures** with multi-process / multi-thread can be **challenging**:
  - Want to **load all cores**, but with **as little data as possible** to **reduce memory footprint**.
- See backup for details on the ALICE framework.

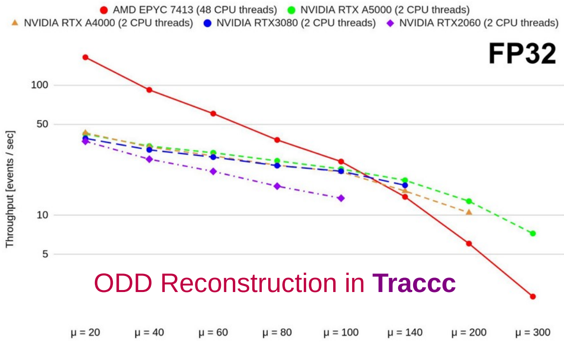
Common efforts / frameworks

- Experiments working together to save effort on common problems.
  - CERN **NextGen Trigger Project** is an effort to support the developments (hardware and software) for HL-LHC, mostly for ATLAS and CMS but ALICE and LHCb have small contributions.
    - Work packages for experiment-specific and common developments, e.g. **efficient data structures** for **accelerators**, **ML interfaces**, ...

Common efforts / frameworks

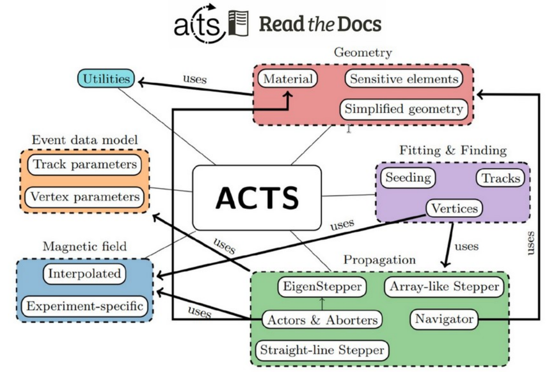
- Experiments working together to save effort on common problems.
  - CERN **NextGen Trigger Project** is an effort to support the developments (hardware and software) for HL-LHC, mostly for ATLAS and CMS but ALICE and LHCb have small contributions.
    - Work packages for experiment-specific and common developments, e.g. **efficient data structures** for **accelerators**, **ML interfaces**, ...
  - **ACTS**: common tracking framework developed at CERN, aiming at HL-LHC and also FCC.
    - **Experiment-independent toolkit for tracking**.
    - Use by several experiments, e.g. by **ALICE** for Run 5 studies, **ATLAS** for HL-LHC, **FASER**, **sPHENIX**, ...
    - **Traccc**: Ongoing work to integrate tracking with GPUs / accelerators
  - Other common HL-LHC challenges, e.g. **DC24** (recent network transfer challenge).

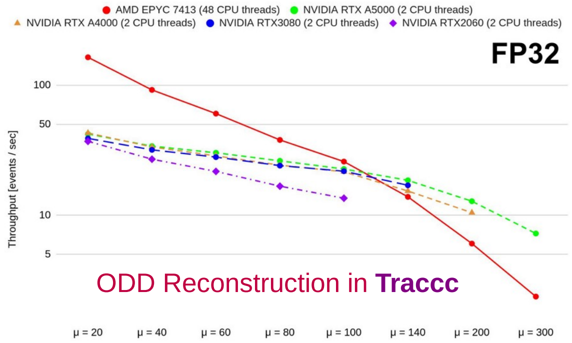




Common efforts / frameworks

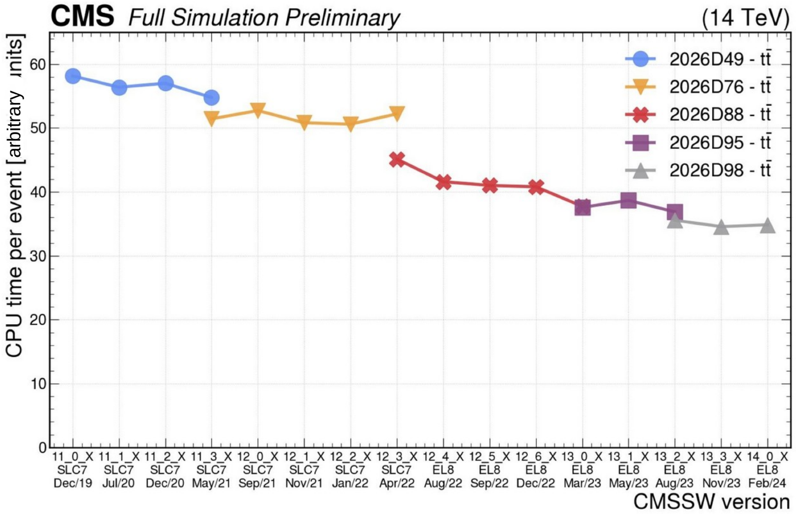
- Experiments working together to save effort on common problems.
  - CERN **NextGen Trigger Project** is an effort to support the developments (hardware and software) for HL-LHC, mostly for ATLAS and CMS but ALICE and LHCb have small contributions.
    - Work packages for experiment-specific and common developments, e.g. **efficient data structures** for **accelerators**, **ML interfaces**, ...
  - **ACTS**: common tracking framework developed at CERN, aiming at HL-LHC and also FCC.
    - **Experiment-independent toolkit for tracking**
    - Use by several experiments, e.g. by **ALICE** for Run 5 studies, **ATLAS** for HL-LHC, **FASER**, **sPHENIX**, ...
    - **Traccc**: Ongoing work to integrate tracking with GPUs / accelerators
  - Other common HL-LHC challenges, e.g. **DC24** (recent network transfer challenge).
- Many obvious **common software frameworks**: **Geant4**, **ROOT**, **Generators**
- Experiments following their efforts to accelerate **simulation** using **GPUs**:
  - **Adept** and **Celeritas** for Geant4:
    - **ATLAS** is looking into them, **LHCb** testing Adept in Gauss.
  - **Opticks** and **Mitsuba3** for optical photons in LHCb.
  - **MG4GPU**, **Pepper** for Sherpa, ...





Evolution of simulation timing

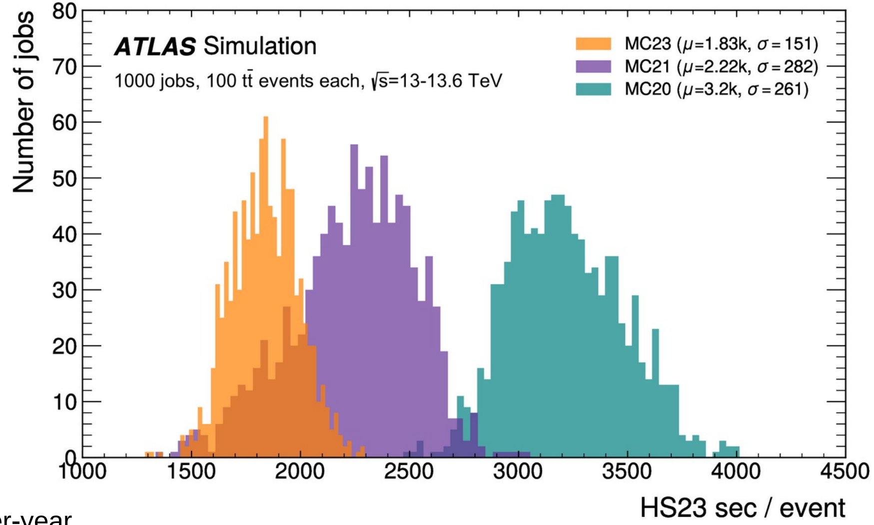
Improving Full CMS Geant4 Phase-2 Simulation over 5 years.



Last major computing model review (Nov. 2021): CMS aspired to 10% year-over-year improvements in SIM & RECO timing. *See [CMS Note-2022/008](), Table 7, Rows 1 & 6.*

- Plot of simulation wall time per event in arbitrary units as a function of releases shows we approximately achieved that for the G4 simulation.

ATLAS MC Speedup



- 2x speed improvement in ATLAS simulation from a wide variety of optimizations. (ICHEP talk)

- On top of **gradual FullSim speedup**, **all experiments** working on **FastSim**, new efforts to use **ML**, directly or to refine FastSim, and GPU usage (last slide). (**LHCP Poster (CMS Full and Fast Sim)**, **Talk on ATLAS FastSim**)

ARM, Open Data, Storage / Analysis

- **Important customers for IT companies are ML, HPC centers, not HEP.**
- **CPU + GPU** Classical heterogeneous systems **transitioning** to e.g. **NVIDIA Grace Hopper**, **AMD MI300**, **Apple M3**, ...
  - Usage of **ARM** CPUs: All experiments testing and validating ARM (**link**):
    - **ATLAS** already **supporting ARM** with some **serious workloads** (**60%**), will **accept ARM pledge next year**.
    - **ALICE**, **LHCb** can **compile** on **ARM** but **still validating**, **CMS** has **validated** but is **redoing MC**.

ARM, Open Data, Storage / Analysis

- **Important customers for IT companies are ML, HPC centers, not HEP.**
- **CPU + GPU** Classical heterogeneous systems **transitioning** to e.g. **NVIDIA Grace Hopper**, **AMD MI300**, **Apple M3**, ...
  - Usage of **ARM** CPUs: All experiments testing and validating ARM (**link**):
    - **ATLAS** already **supporting ARM** with some **serious workloads** (**60%**), will **accept ARM pledge next year**.
    - **ALICE**, **LHCb** can **compile** on **ARM** but **still validating**, **CMS** has **validated** but is **redoing MC**.
- **Open Data**:
  - This years **CMS** and **ATLAS open data releases** are already out (**ATLAS link, CMS link**)
  - **LHCb** has **released** its entire Run 1 open data set and developed the **Ntuple wizard** system to ease access.
  - **ALICE** has released 6.5 TiB of its Run 2 ESD dataset (5% of Pb-Pb, 7% of pp).
    - Ongoing work to convert Run 2 data to Run 3 format, and release Run 2 and Run 3 data sets.
- **Different data formats** are a **problem** - both for **open data** releases and for **reconstruction** / **analysis** of the experiments.
  - **Support** of Run 1 **legacy data** with **HL-LHC** software: **LHC spans 30+ years**.
  - **ALICE Run 3**: **Events** v.s. **time frames** (~2.8 ms of continuous data).

ARM, Open Data, Storage / Analysis

- **Important customers for IT companies are ML, HPC centers, not HEP.**
- **CPU + GPU** Classical heterogeneous systems **transitioning** to e.g. **NVIDIA Grace Hopper**, **AMD MI300**, **Apple M3**, ...
  - Usage of **ARM** CPUs: All experiments testing and validating ARM (**link**):
    - **ATLAS** already **supporting ARM** with some **serious workloads** (**60%**), will **accept ARM pledge next year**.
    - **ALICE**, **LHCb** can **compile** on **ARM** but **still validating**, **CMS** has **validated** but is **redoing MC**.
- **Open Data**:
  - This years **CMS** and **ATLAS open data releases** are already out (**ATLAS link, CMS link**)
  - **LHCb** has **released** its entire Run 1 open data set and developed the **Ntuple wizard** system to ease access.
  - **ALICE** has released 6.5 TiB of its Run 2 ESD dataset (5% of Pb-Pb, 7% of pp).
    - Ongoing work to convert Run 2 data to Run 3 format, and release Run 2 and Run 3 data sets.
- **Different data formats** are a **problem** - both for **open data** releases and for **reconstruction** / **analysis** of the experiments.
  - **Support** of Run 1 **legacy data** with **HL-LHC** software: **LHC spans 30+ years**.
  - **ALICE Run 3**: **Events** v.s. **time frames** (~2.8 ms of continuous data).
- **Not only storage itself, but data access (more for anaylsis than for reconstruction):**
  - Already **reading from tape / disk** can be a **bootleneck**.
  - Storage **implementations** are **experiment-specific**, need to improve and work on common solutions.
  - Try to **reduce number of replicas**, but also complicates / slows down data access further.
  - **Avoid framework bottlenecks** moving the data through our applications.
  - **See next slide**.

Analysis

See also multi-experiment
LHCP talk on analysis

- New analysis paradigms: **Analysis facilities**, **columnar** analysis, **notebooks** / **python** based analysis, **interactive** / **fast turnaround** time, **avoid re-reading** the same data.
- **Analysis facilities** with data consolidated on-site.
  - Accessible from outside
  - All **tools available**
  - **Distributed** and **interactive analysis**

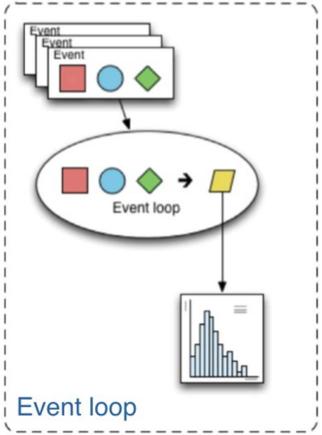
Analysis

See also multi-experiment **LHCP talk** on analysis

- New analysis paradigms: **Analysis facilities**, **columnar** analysis, **notebooks** / **python** based analysis, **interactive** / **fast turnaround** time, **avoid re-reading** the same data.

- **Analysis facilities** with data consolidated on-site.
  - Accessible from outside
  - All **tools available**
  - **Distributed** and **interactive analysis**

- **Columnar analysis** / data types reduce data size, speed up reading / processing, e.g. ALICE uses **apache arrow**.



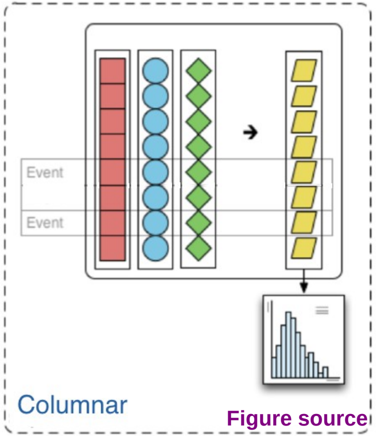


Figure source

Analysis

- New analysis paradigms: **Analysis facilities**, **columnar** analysis, **notebooks** / **python** based analysis, **interactive** / **fast turnaround** time, **avoid re-reading** the same data.
- **Analysis facilities** with data consolidated on-site.
  - Accessible from outside
  - All **tools available**
  - **Distributed** and **interactive analysis**
- **Columnar analysis** / data types reduce data size, speed up reading / processing, e.g. ALICE uses **apache arrow**.

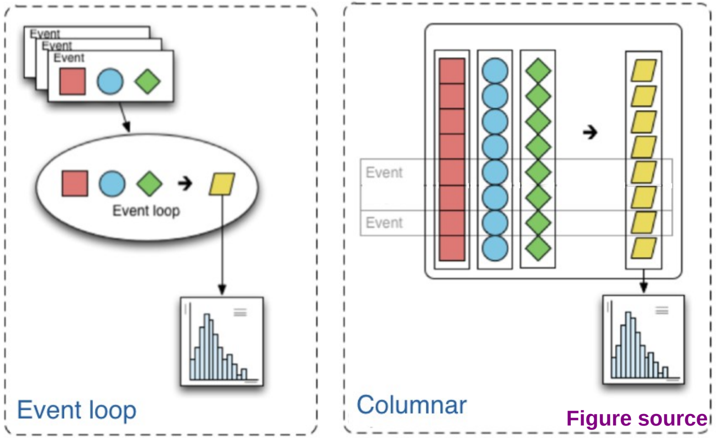
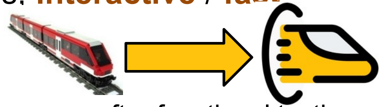
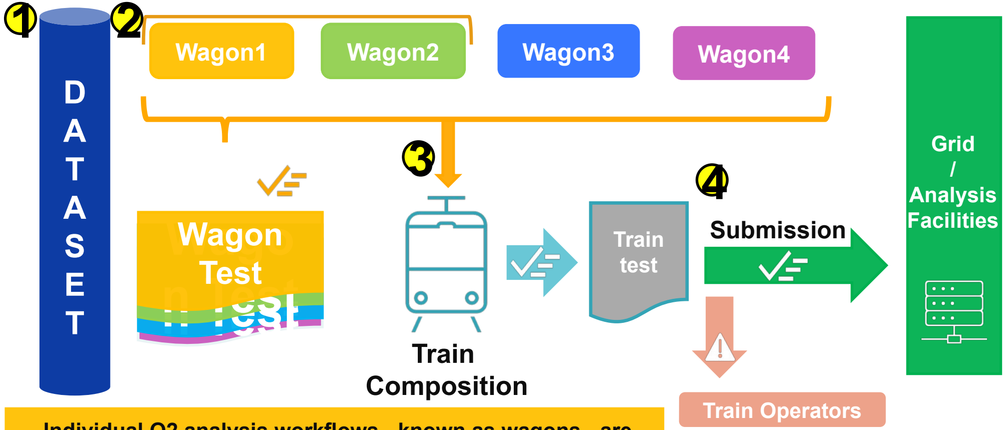


Figure source

- ALICE transition: Run 2 **LEGO-trains** to **Hyperloop**



  - **Web-assisted** composition of **trains** from **many users** after functional testing.
  - **Train** reads data once for all tasks – on **GRID** and **AFs**.
  - Central operators, **fair usage**, **automatic grouping** of tasks.
  - **24/5 support** with **4 institutes** in **different time zones**.
  - Hyperloop last ½ **year**: **7500 CPU years**, 10400 trains, **400 PB input**.
  - Processed **more data** than LEGO-trains in **20000 CPU** years (same ½ year).

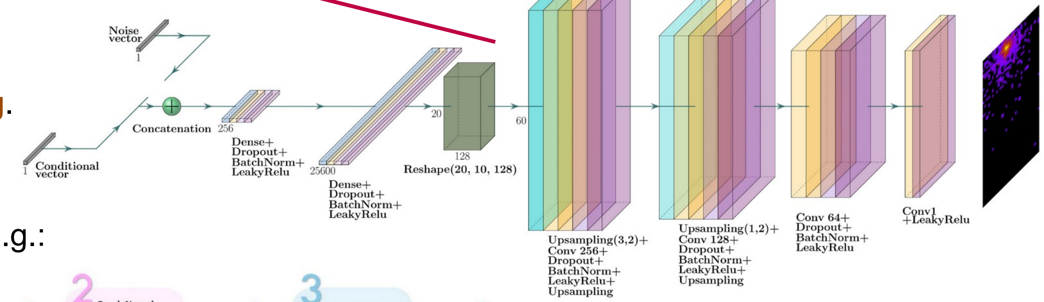


Individual O2 analysis workflows - known as wagons - are combined into trains to run over the data together

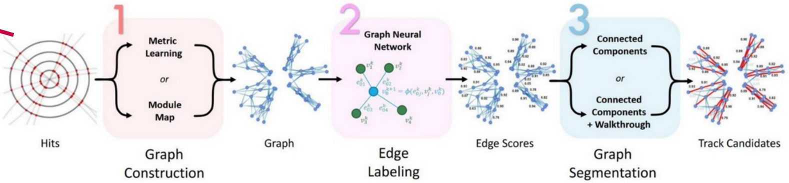
Machine Learning

Talk on LHCb ML realtime analysis
Talk on ATLAS ML for calorimeter processing
Talk on LHCb ML GNN-based tracking

- **ML started mostly in analysis, now pervasive everywhere in all experiments (simulation, calibration, reconstruction).**
- **Only a quick list of related topics, since ML was covered in Javier's talk.**
  - **ATLAS** using **GPUs** for ML-related parts in **Analysis** / **Reconstruction**, **ALICE** uses **GPUs** for **ML** in **Analysis**.
  - **ML** used in **FastSim**, e.g. **ALICE** Zero Degree Calorimeter (**ZDC**) – **100x speedup** (**link**).
  - **GANs** used in **ATLAS** for **FastSim** (**link**).



  - **ML** also used for **calibration** and **clustering**.
  - **LLMs**, e.g. **chATLAS**.
  - Continuous efforts to use **ML** for **tracking,** E.g.:
    - **ATLAS GNN4ITk**
    - **GNN tracking @LHCb**



- **Large ML models** are **challenging**, how to train, manage, integrate them?
  - **CERN NextGen Trigger project**: purchase **powerful hardware** for **development, ML training** for **all experiments**.
  - **Work package** looking at **integrating inferencing** in **experiment frameworks** (on **GPUs** or **FPGAs**).
    - Particularly **efficient data flow** and **integration with non-ML algorithms** in **online processing**.

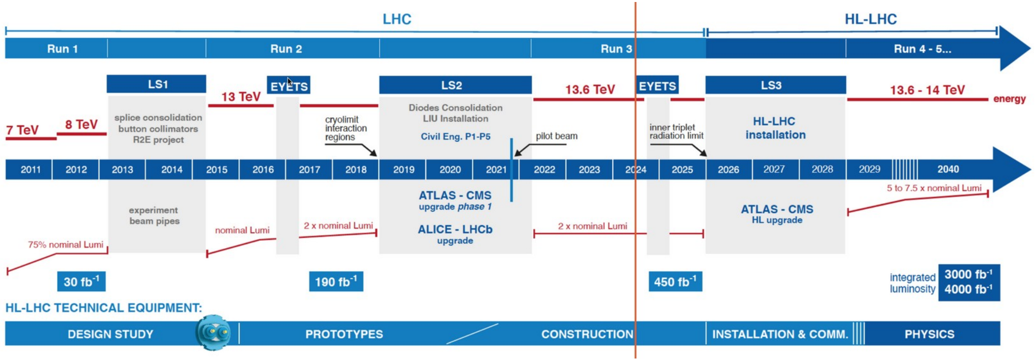
Conclusions

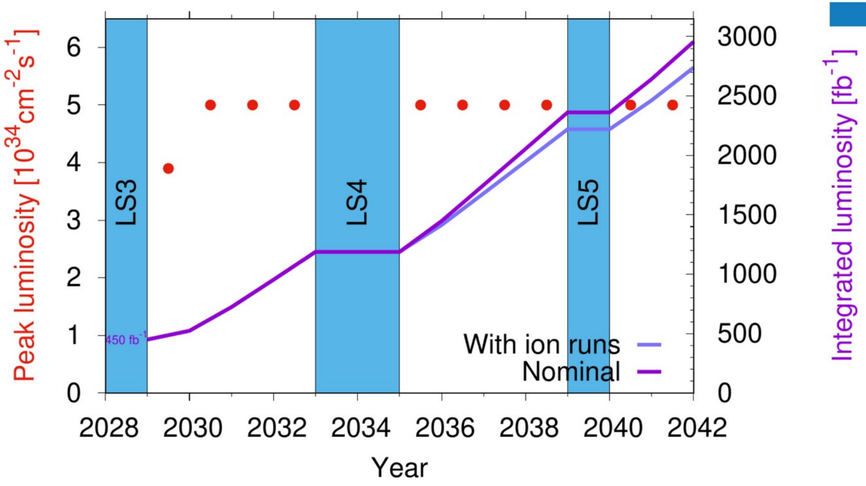
- **HL-LHC Upgrades** challenging for computing of all LHC experiments.
  - → Adapt to **modern architectures** / **HPC centers** / etc. to keep step ... and make them available in the **GRID**.
- **Computing** tough, but **storage** capacity & storage speed **grow even slower**!
  - – **Efficient triggers** and **data compression** mandatory, **optimize analysis data** for **size**.
  - – **Reading data fast** and only **once** is critical for **analysis** (**trains**, **columnar analysis**, **analysis facilities**).
- **ALICE** and **LHCb** (**HLT1**) do **full online processing** in **software**, **ATLAS** and **CMS** need **HW triggers** for **HL-LHC**.
  - – **ALICE** and **LHCb** run **full online computing** on **GPUs**, **large savings** on online farms.
  - – **CMS** runs **mixed** on **CPU+GPU**, **ATLAS** is investigating and developing for Run 4.
- **Offline** reconstruction on **GPUs** more **complicated:** more different algorithms, **no clear hotspot**.
  - – **ALICE** runs **~60%** of **offline reconstruction** on GPU **since 2023**, **2.5x speedup**.
  - – **All experiments** work on **GPU** usage beyond online processing, **first attempts** in **GRID** could be **opportunistic**.
- Difficult to improve simulation / analysis : **many 3rd party libraries**, **many code developers**.
  - – All experiments strive to improve their full simulation speed, **~2x improvements** from **ATLAS** and **CMS** shown.
  - – Studies / developments for **simulation** on **GPU** ongoing (**Adept**, **Celeritas**, ...).
  - – **FastSimulation** and new approaches with **ML** will help in the future.
- **Evolution** of **data model** critical for everyone, should be **GPU-friendly**, more **efficient for Analysis**, e.g. **RNTuple**.
- **Multi-threading** mandatory, both for speed and to reduce **memory footprint**, pure **multi-processing no longer working**.
- **Pervasive ML** use **from simulation to analysis**, **lacking turn-key systems** and **seamless handling** of **large models**.

Backup

Recent and upcoming upgrades

- **HL-LHC**: **ATLAS** / **CMS** luminosity $2x10^{34}s^{-1}cm^{-2}$ to $\mathbf{7.5x10^{34}s^{-1}cm^{-2}}$
- **Nominal for μ = 132, ultimate could be 200.**





Real speedup in offline reconstruction

- **For offline reconstruction, online nodes are used as GRID nodes.**
  - **Identical workflow** as on other **GRID** sites, only different configuration using GPU, more memory, more CPU cores.
  - Online farm split in **2 scheduling pools**: online and offline.
    - Unused nodes in the online pool are moved to the offline pool.
    - As needed for data-taking, nodes are moved to the online pool with lead time to let the current jobs finished.
      - If needed immediately, GRID jobs are killed and nodes moved immediately.
- **Performance benchmarks cover multiple cases:**
  - Server split into 16 * **8 cores**, or into 8 * **16 cores**, ignoring the GPU : to compare CPUs and GPUs.
  - Server split into 8 or 2 identical fractions: **1 NUMA** domain (4 GPUs) or **1 GPU**.
- **Processing time per time-frame while the GRID job is running (neglecting overhead at start / end, 2023 software).**
  - In all cases server **fully loaded** with **identical jobs**, to avoid effects from HyperThreading, memory, etc.

Configuration (2022 pp, 650 kHz, 7400 collisions per TF)	Time per TF (1 instance)	Time per TF (full server)
CPU 8 cores	76.91s	4.81s
CPU 16 cores	34.18s	**4.27s**
1 GPU + 16 CPU cores	14.60s	1.83s
1 NUMA domain (4 GPUs + 64 cores)	3.5s	**1.70s**

Factor 2.51
Matches expected factor 2.5

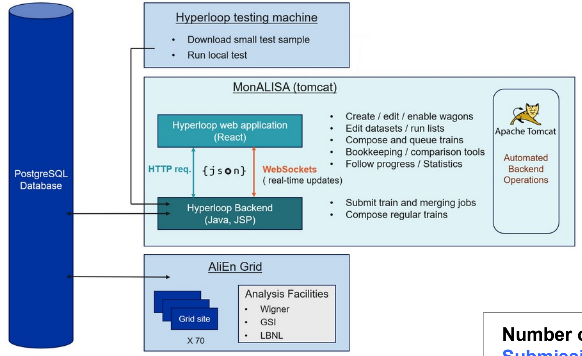
- Ongoing effort to port more code to GPU – In 2024 gain **1.5% to 5%** by offloading **TPC data decoding** (as master thesis).

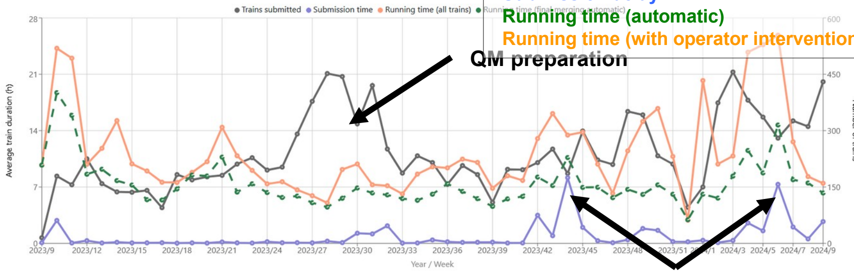
Hyperloop

See also multi-experiment **LHCP talk** on analysis

- Fair usage, small tasks auto-submitted

100 CPU years = more than a lifetime on your laptop!

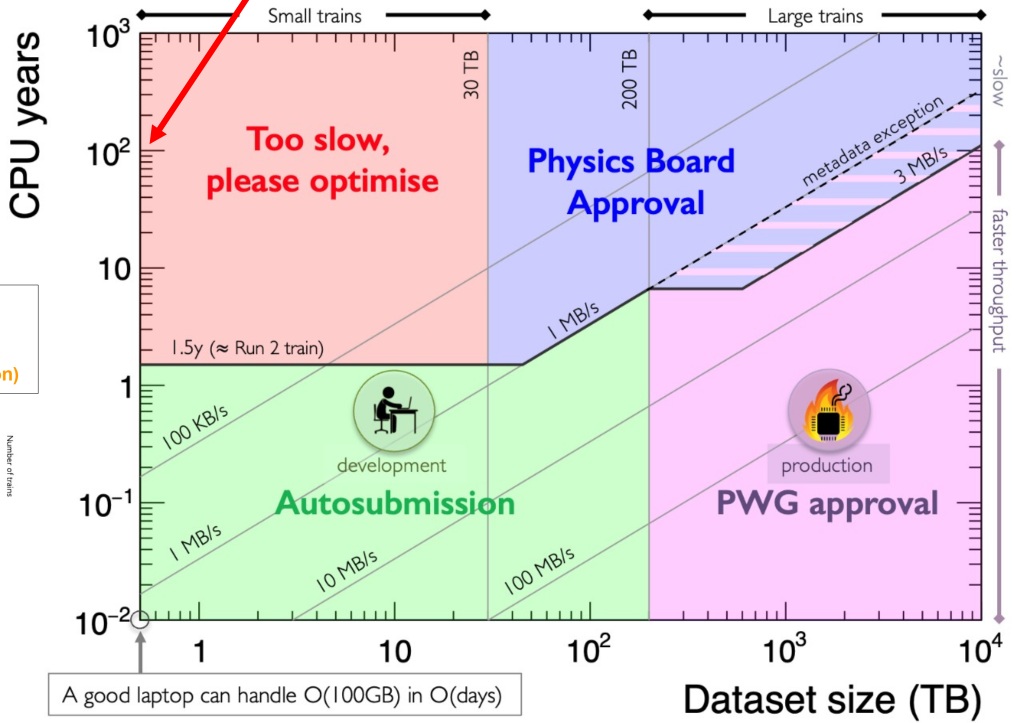




Number of trains
Submission delay
Running time (automatic)
Running time (with operator intervention)

QM preparation

Year and Week no.



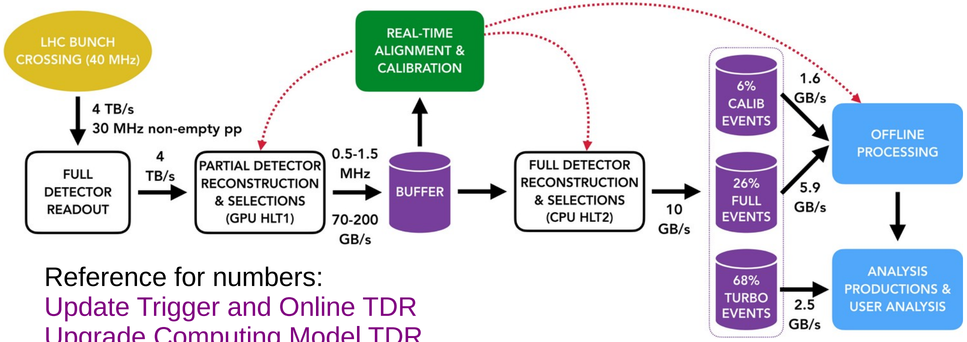
- **About 330 analyzers**
- **Up to 450 trains/week**
- **Average train duration about 13 hours**

Heavy trains cause departure delays (for others) up to 12 hours (@ users: check Dashboard)

Online Computing & GPU / FPGA Usage

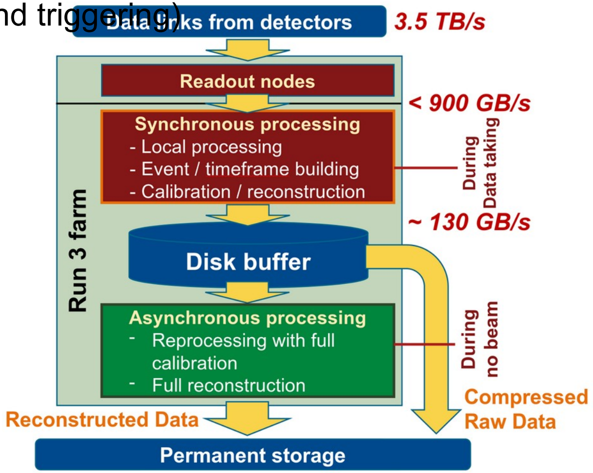
- Experiments going in different directions:
  - **ALICE** & **LHCb removed** the **hardware trigger**, doing **full online processing** with **GPUs**:
    - **LHCb** runs the **full HLT1 trigger** on **GPUs** with the **Allen** framework.
    - **ALICE** does **not trigger at all** (in **Pb-Pb**), and runs **online calibration and compression**, having **~99%** of the reconstruction on **GPU** (more does not make sense).
      - For **pp**, **ALICE** runs a delayed analysis trigger on the data stored on the disk-buffer (see later), to **skim** which **data** is **written to permanent storage**.
    - **ALICE** & **LHCb** use a local **disk buffer**, with the **online farm** processing data when **no beam** in LHC. (ALICE: asynchronous (offline) reprocessing, LHCb: final processing and triggering).

LHCb



Reference for numbers:
Update Trigger and Online TDR
Upgrade Computing Model TDR

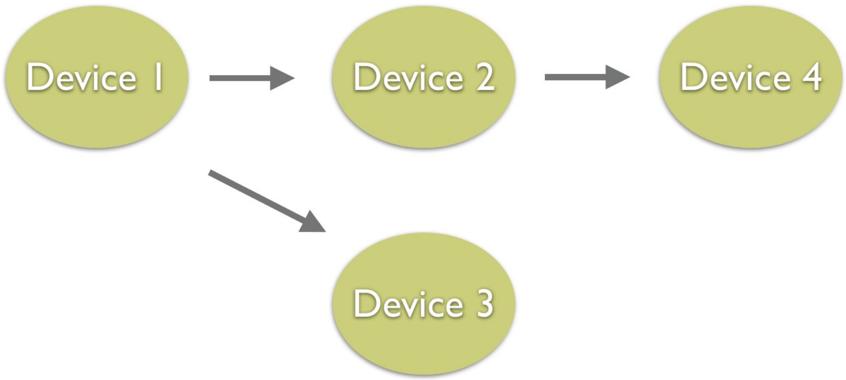
ALICE



N.B. All experiments are / were using the HLT farms for GRID jobs when idle.

ALICE DPL Processing Framework

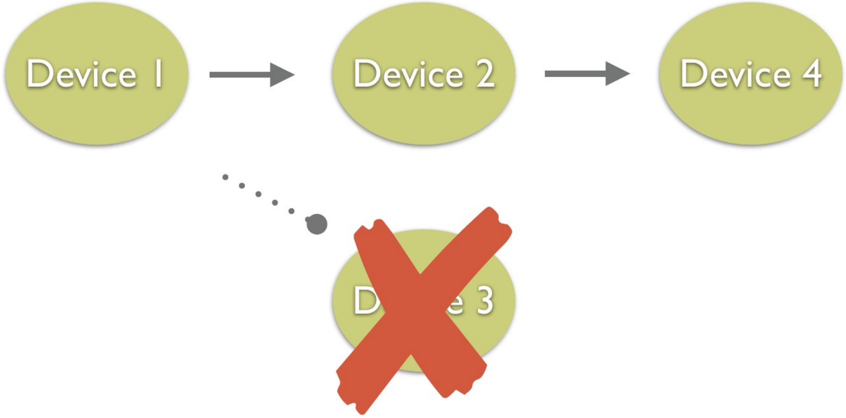
- **ALICE Data Processing Layer (based on FairMQ) is a message-passing based multi-process processing framework.**
  - Processes (devices) are arranged in a **directed graph** (**no loops**, **outputs** are **read only**).



ALICE DPL Processing Framework

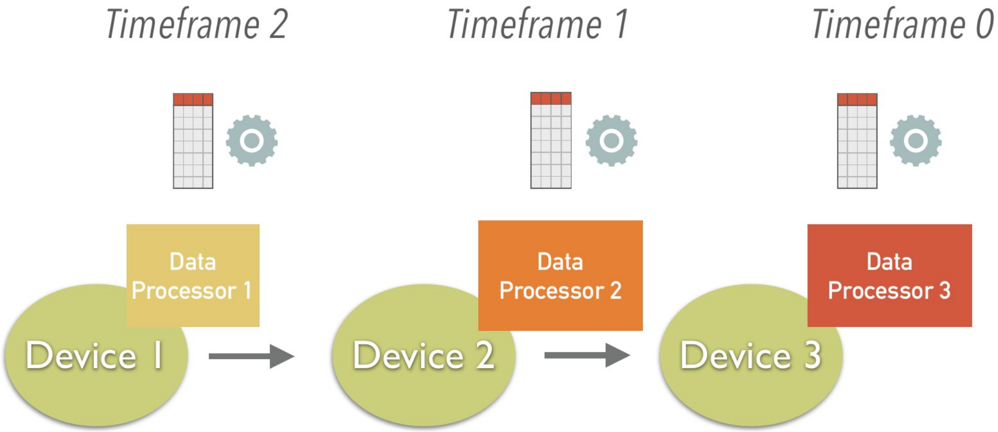
- **ALICE Data Processing Layer (based on FairMQ) is a message-passing based multi-process processing framework.**
  - Processes (devices) are arranged in a **directed graph** (**no loops**, **outputs** are **read only**).
  - Devices can be **expandable**.

E.g. non-critical QA task are **expendable**,
failures will be ignored till next run (or restarted in the future)



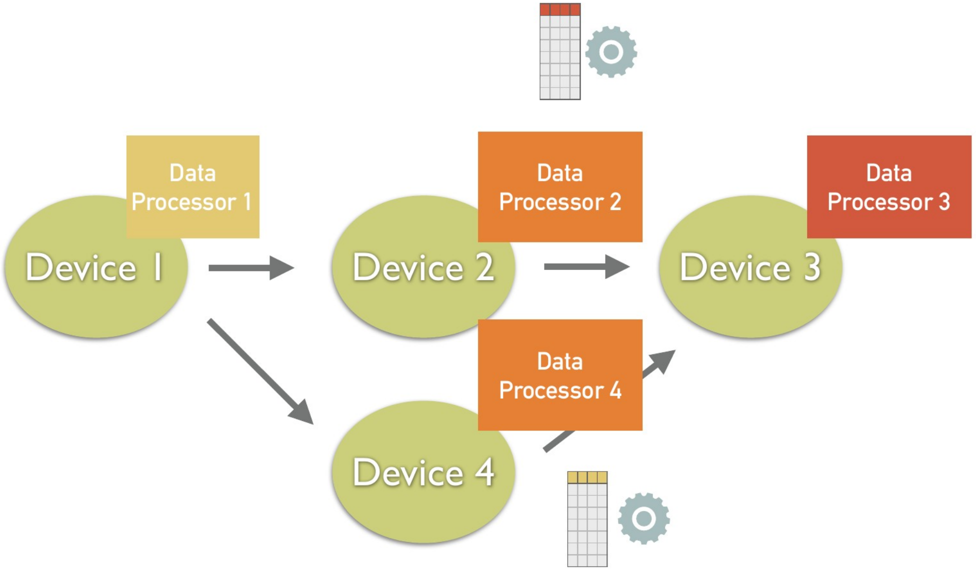
ALICE DPL Processing Framework

- **ALICE Data Processing Layer (based on FairMQ) is a message-passing based multi-process processing framework.**
  - Processes (devices) are arranged in a **directed graph** (**no loops**, **outputs** are **read only**).
  - Devices can be **expandable**.
  - Time perculate through the processing graph.
  - **Horizontal parallelism**:
    Consecutive devices, different time frames.



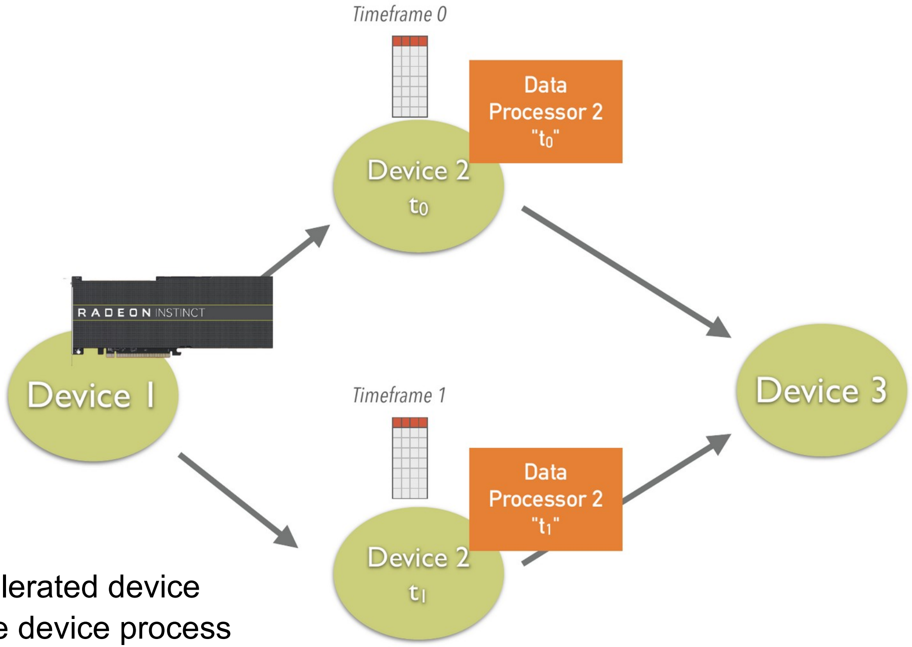
ALICE DPL Processing Framework

- **ALICE Data Processing Layer (based on FairMQ) is a message-passing based multi-process processing framework.**
  - Processes (devices) are arranged in a **directed graph** (**no loops**, **outputs** are **read only**).
  - Devices can be **expandable**.
  - Time perculate through the processing graph.
  - **Horizontal parallelism**:
    Consecutive devices, different time frames.
  - **Vertical parallelism**:
    Independent devices, same time frame.



ALICE DPL Processing Framework

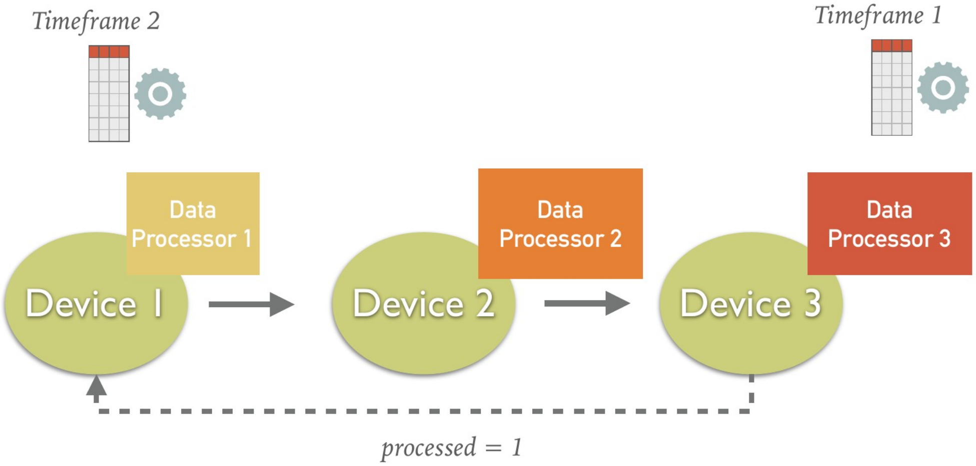
- **ALICE Data Processing Layer (based on FairMQ) is a message-passing based multi-process processing framework.**
  - Processes (devices) are arranged in a **directed graph** (**no loops**, **outputs** are **read only**).
  - Devices can be **expandable**.
  - Time perculate through the processing graph.
  - **Horizontal parallelism**:
    Consecutive devices, different time frames.
  - **Vertical parallelism**:
    Independent devices, same time frame.
    Multiple device instances, data round-robin.



If devices are slower than others (e.g. a GPU-accelerated device could be much faster), multiple instances of the same device process incoming time frames round-robin.

ALICE DPL Processing Framework

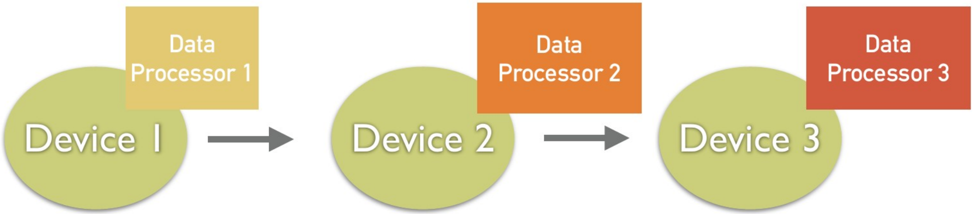
- **ALICE Data Processing Layer (based on FairMQ) is a message-passing based multi-process processing framework.**
  - Processes (devices) are arranged in a **directed graph** (**no loops**, **outputs** are **read only**).
  - Devices can be **expandable**.
  - Time perculate through the processing graph.
  - **Horizontal parallelism**:
    Consecutive devices, different time frames.
  - **Vertical parallelism**:
    Independent devices, same time frame.
    Multiple device instances, data round-robin.
  - **Rate limiting**.



A feedback channel counts how many time frames are in flight.
Limit is configurable, source device will throttle to stay within memory.

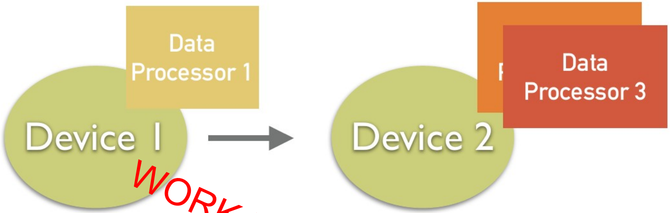
ALICE DPL Processing Framework

- **ALICE Data Processing Layer (based on FairMQ) is a message-passing based multi-process processing framework.**
  - Processes (devices) are arranged in a **directed graph** (**no loops**, **outputs** are **read only**).
  - Devices can be **expandable**.
  - Time perculate through the processing graph.
  - **Horizontal parallelism**:
    Consecutive devices, different time frames.
  - **Vertical parallelism**:
    Independent devices, same time frame.
    Multiple device instances, data round-robin.
  - **Rate limiting**.
  - **Multi-processing** of different algorithms...



ALICE DPL Processing Framework

- **ALICE Data Processing Layer (based on FairMQ) is a message-passing based multi-process processing framework.**
  - Processes (devices) are arranged in a **directed graph** (**no loops**, **outputs** are **read only**).
  - Devices can be **expandable**.
  - Time perculate through the processing graph.
  - **Horizontal parallelism**:
    Consecutive devices, different time frames.
  - **Vertical parallelism**:
    Independent devices, same time frame.
    Multiple device instances, data round-robin.
  - **Rate limiting**.
  - **Multi-processing** of different algorithms...
    ... or **multi-threading** in the same device.

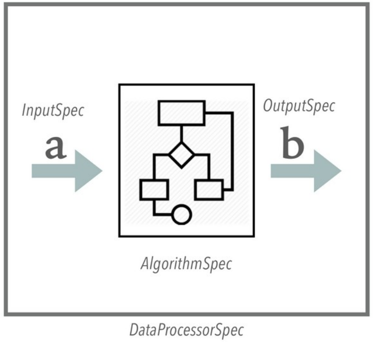


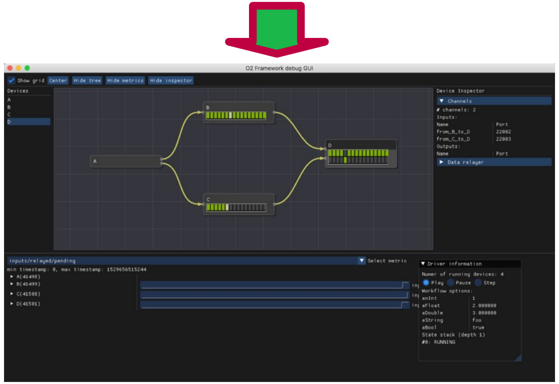
WORK IN PROGRESS

Mandatory next step to **safe memory**.
So far devices can multi-thread internally.
Working on framework support to automatically multithread different thread-safe algorithms.

ALICE DPL Processing Framework

- **ALICE Data Processing Layer (based on FairMQ) is a message-passing based multi-process processing framework.**
  - Processes (devices) are arranged in a **directed graph** (**no loops**, **outputs** are **read only**).
  - Devices can be **expandable**.
  - Time perculate through the processing graph.
  - **Horizontal parallelism**:
    Consecutive devices, different time frames.
  - **Vertical parallelism**:
    Independent devices, same time frame.
    Multiple device instances, data round-robin.
  - **Rate limiting**.
  - **Multi-processing** of different algorithms...
    ... or **multi-threading** in the same device.
- **Declarative topology description translated implicitly to explicit workflow.**
- **DebugGUI enables on-the-fly workflow inspection.**

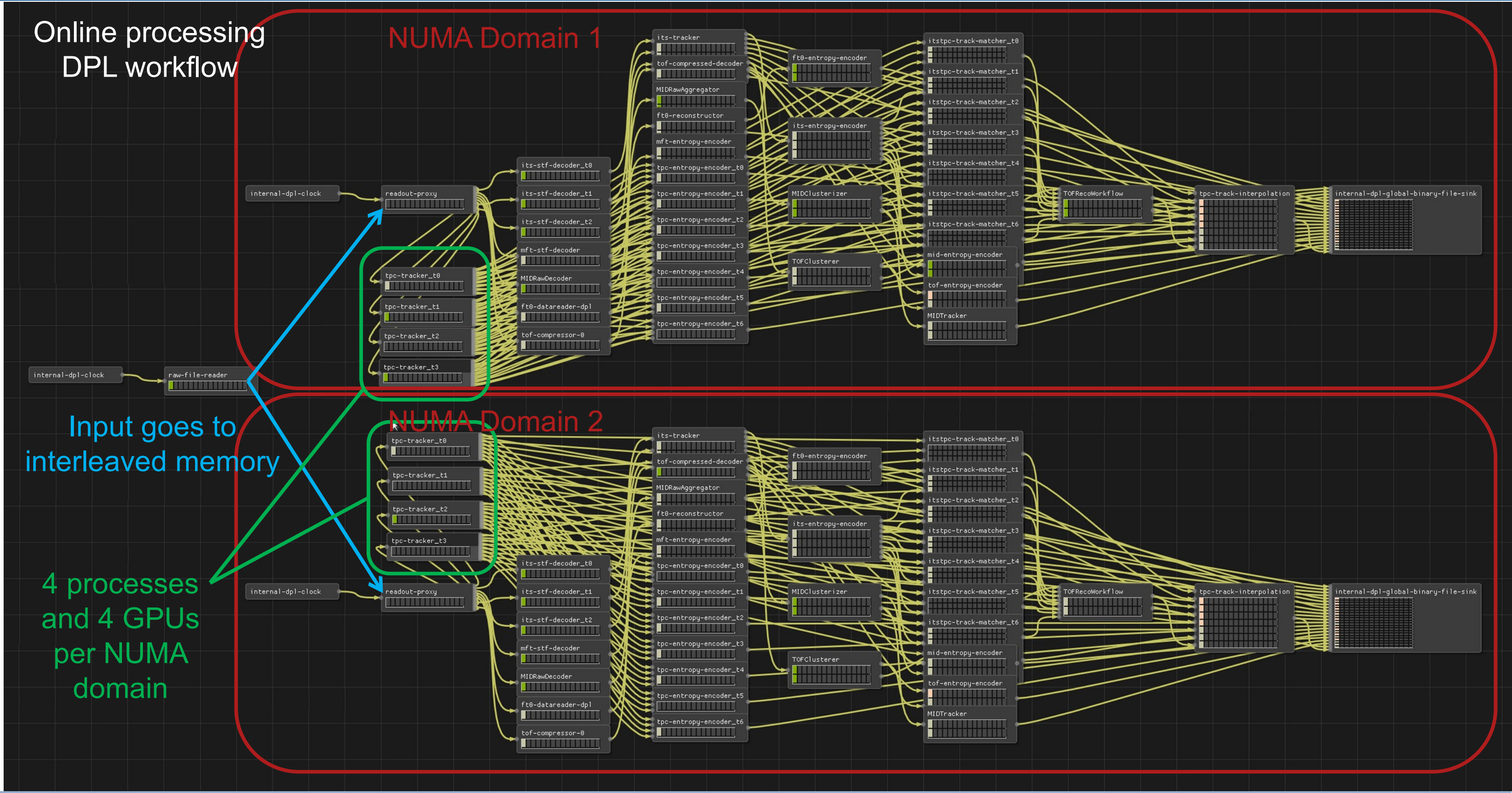




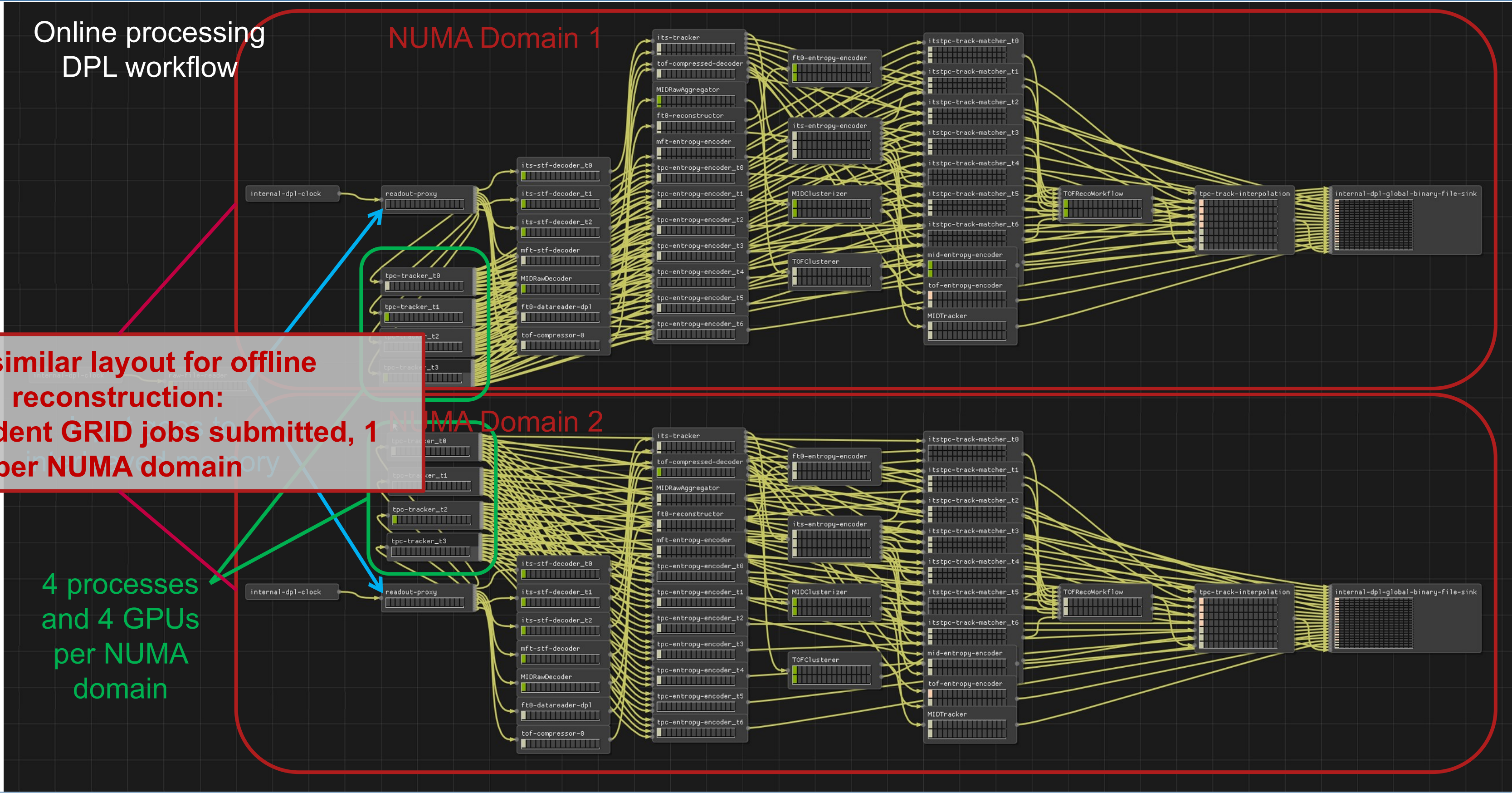
ALICE DPL Processing Framework

- **ALICE Data Processing Layer (based on FairMQ) is a message-passing based multi-process processing framework.**
  - Processes (devices) are arranged in a **directed graph** (**no loops**, **outputs** are **read only**).
  - Devices can be **expandable**.
  - Time perculate through the processing graph.
  - **Horizontal parallelism**:
    Consecutive devices, different time frames.
  - **Vertical parallelism**:
    Independent devices, same time frame.
    Multiple device instances, data round-robin.
  - **Rate limiting**.
  - **Multi-processing** of different algorithms...
    ... or **multi-threading** in the same device.
- **Declarative topology description translated implicitly to explicit workflow.**
- **DebugGUI enables on-the-fly workflow inspection.**
- **Same framework used for simulation, online and offline reconstruction, and analysis.**
- **Workflow can be tuned for the architecture / available resources.**
  - E.g. more memory / more CPU cores allows for more parallel devices.
  - See example of **online processing workflow** on the next slide.

ALICE Online Processing Workflow



ALICE Online Processing Workflow



Very similar layout for offline reconstruction:
2 independent GRID jobs submitted, 1 per NUMA domain