# *ATLAS Trigger Strategies*

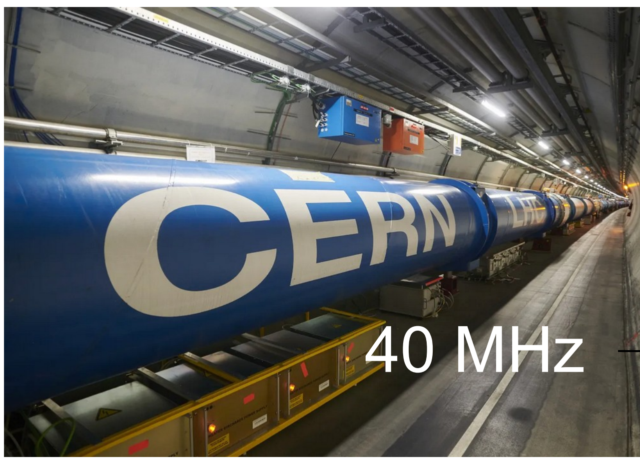
The Past, the Present and the Future

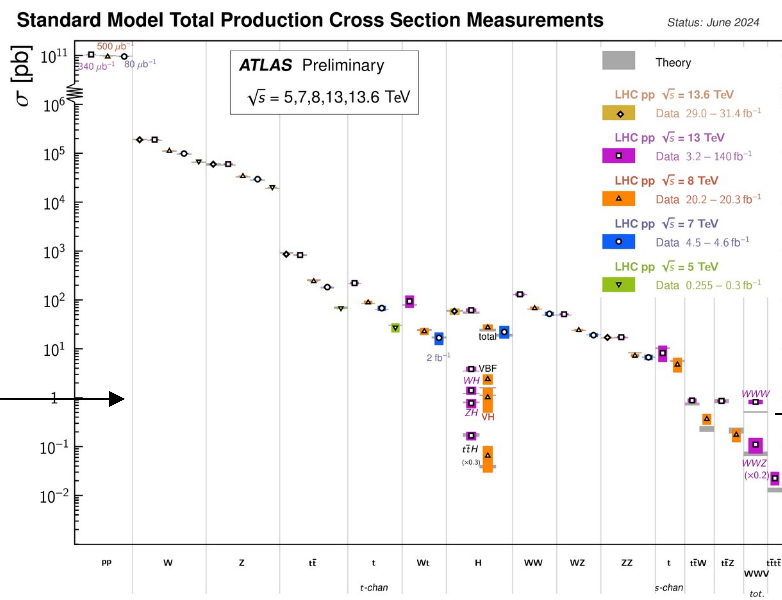
*L. Fiorini*

COMCHA Workshop
03/10/2024

# *Trigger motivations*

- We want to study SM processes + BSM searches.
- 40 MHz of events at the LHC.
- Events are large (~2MB/event → 80 TB/s) and complex.



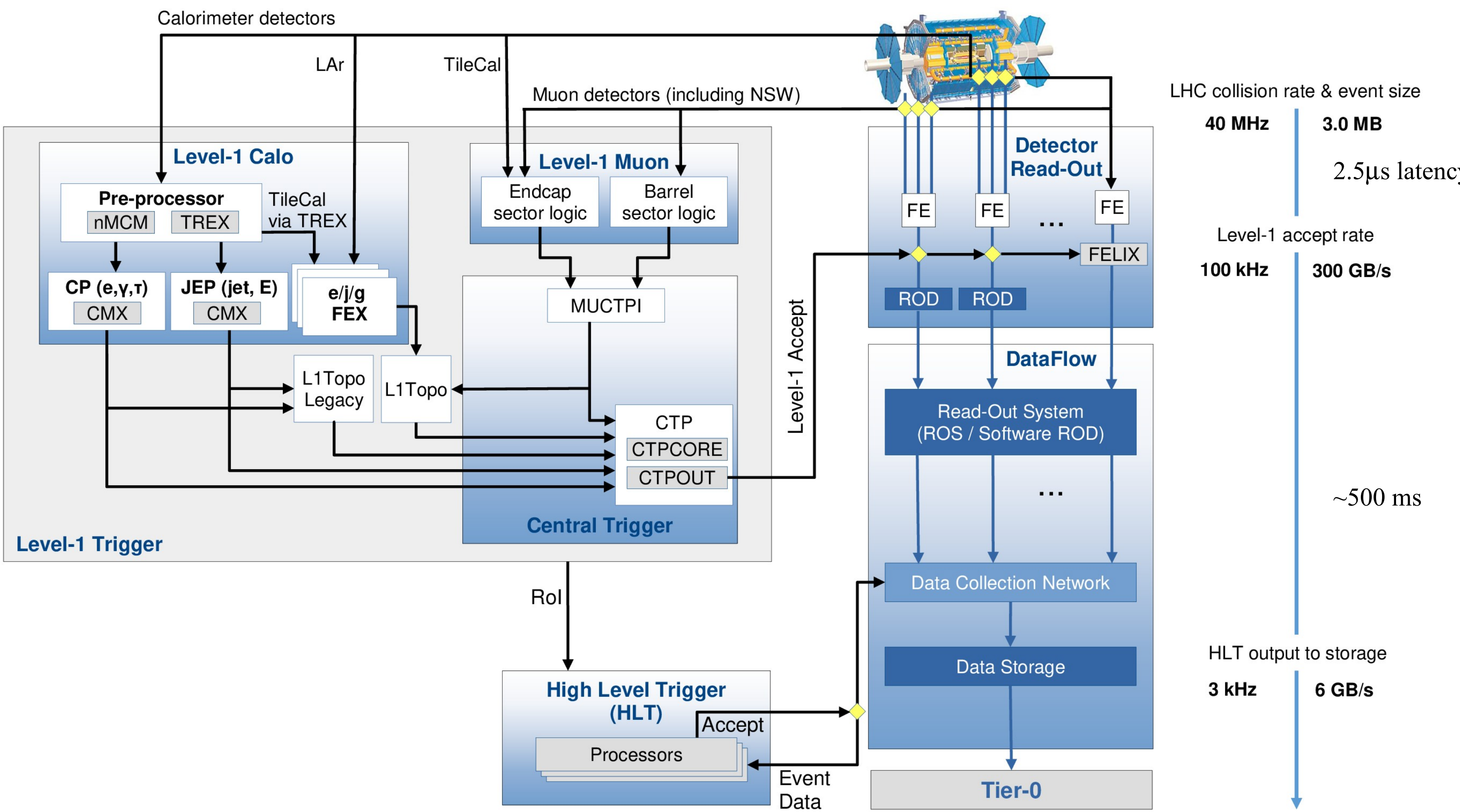




80 TB/s

- The trigger has to be **fast**, **flexible** and **selective**.
- But also simple and robust, it has be to work reliably all the time or we lose data forever.

# Trigger structure

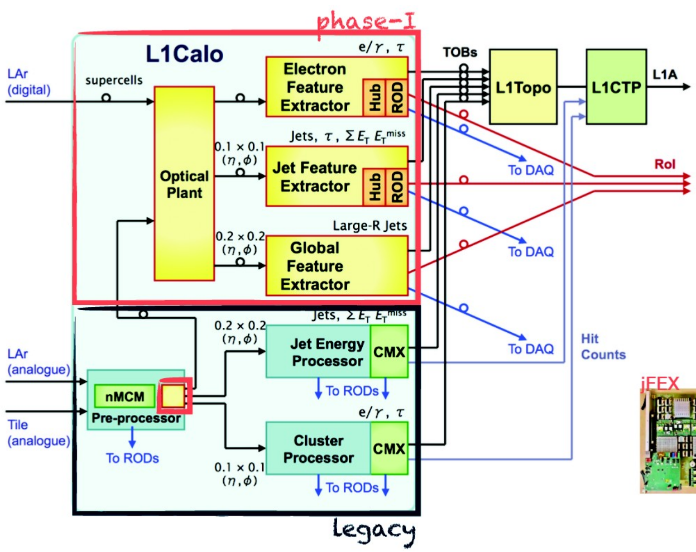
- ATLAS Trigger System is a 2-step staged system
- L1: Hardware trigger, based on a subset of subset of systems.
- HLT: Software trigger, full event reconstruction similar to offline.



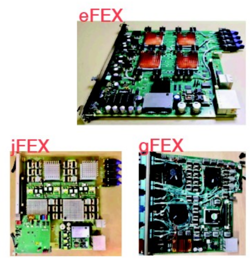
# L1 Trigger

- L1 Calo+L1Topo
- L1 Muons
- CTP

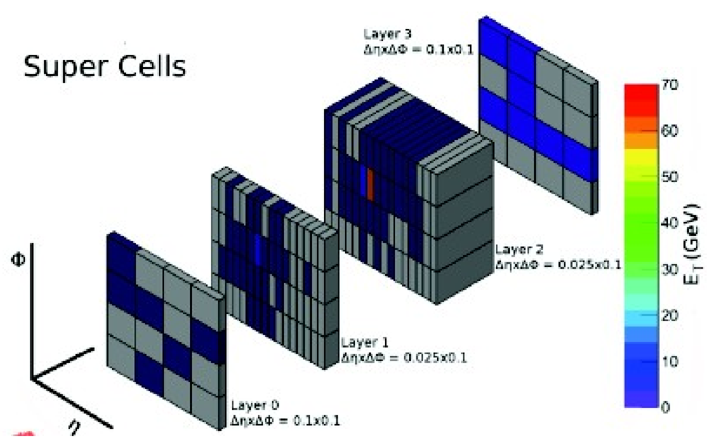
L1 Calo




Features EXtraction for jets, photons, electrons...



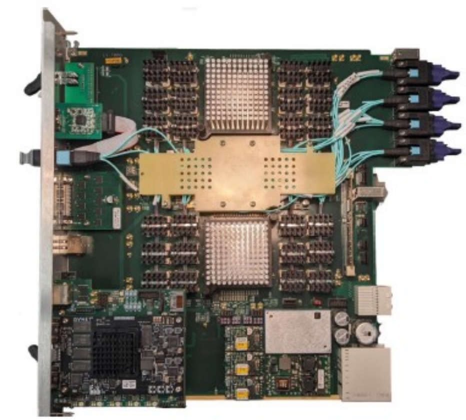

Super Cells



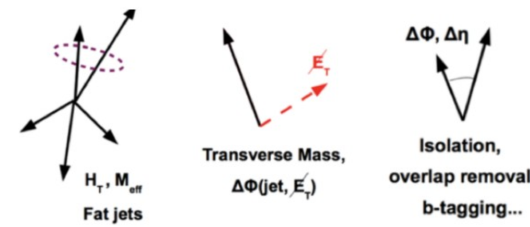

Hardware based on DSP, FPGAs

L1 Topo

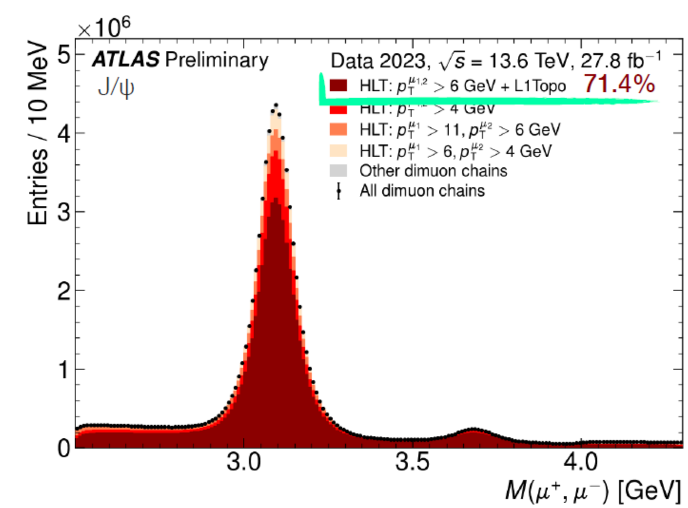
- Receives inputs from other L1 Triggers
- Applies real-time kinematic and angular selections at L1, e.g. ΔR
- Based on FPGAs
- Operative since 2016, upgraded for Run 3.



- Crucial to select B-physics and light states events



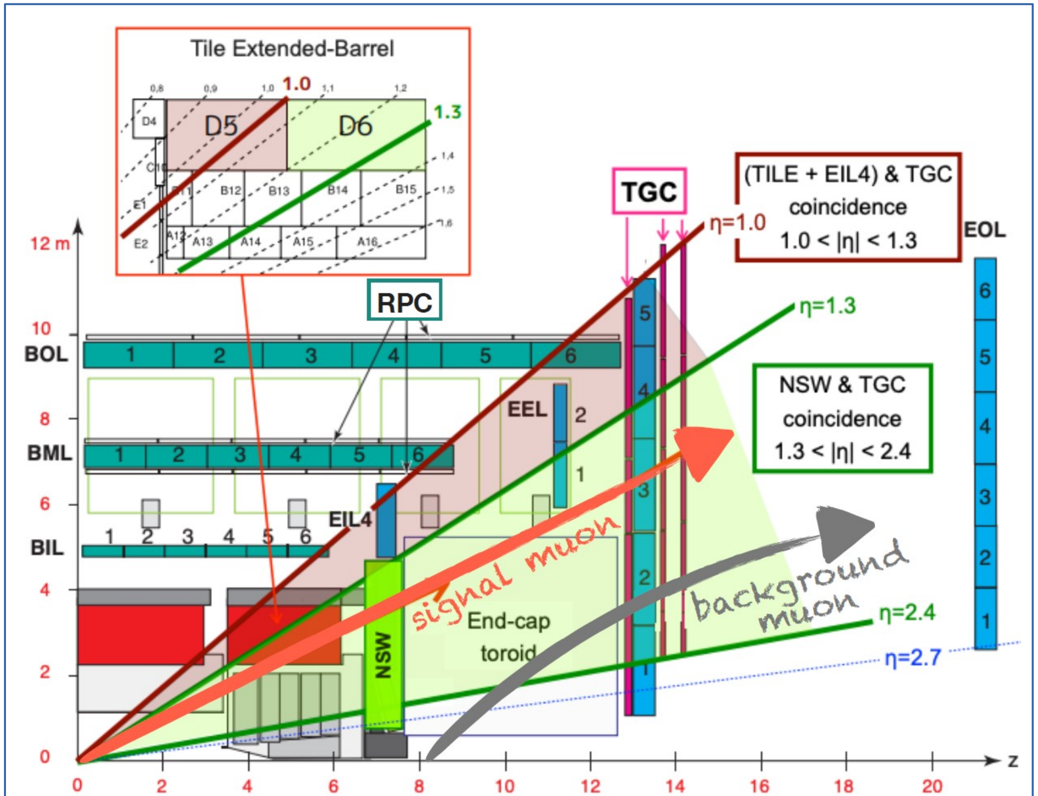

- L1Topo triggers provide about 70% of unique rate for J/ψ



# *L1 Trigger*

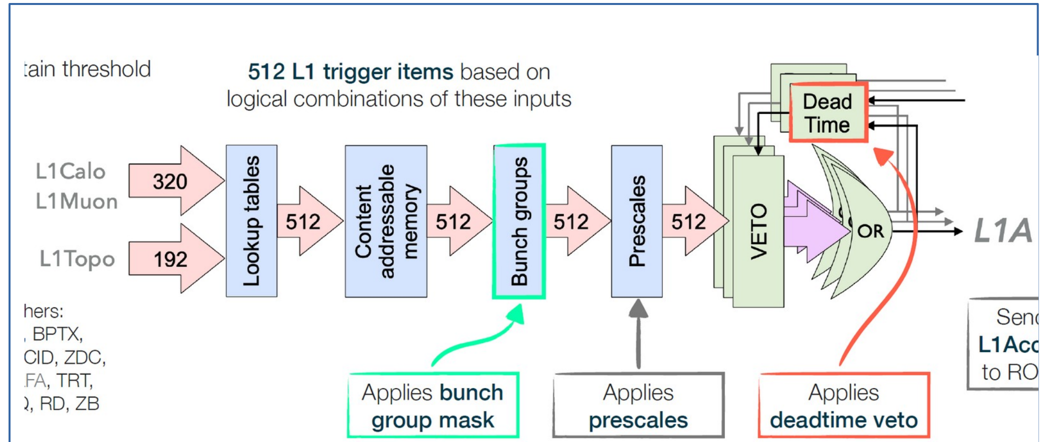
- L1 Calo+L1Topo
- L1 Muons
- CTP

L1 Muon




- Use coincidence patterns within parameterised geometrical "trigger roads" to identify muon candidates.
- Coincidence of TGC with TileCal and NSW to reduce background.

L1 CTP



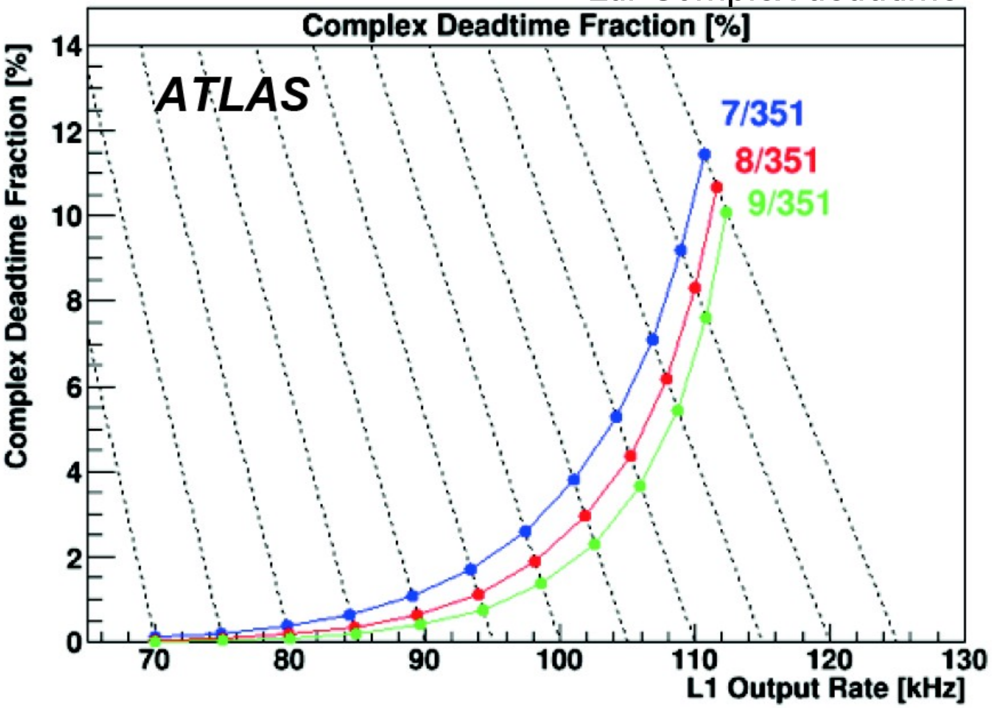

- Receives inputs from all L1 trigger systems
- Identifies Regions of Interest (RoIs) with coarse information:
  - ηxφ information in the muon spectrometer and the calorimeters
  - type of object (corresponding to EM, tau, jet, muon objects)
  - threshold passed (pT, ET)
- → Provide the information to the High Level Trigger

# Deadtime

Deadtime in practice

- Deadtime depends on:
  - the overall L1-trigger rate
  - its configuration (bucket size etc)
  - the LHC filling scheme (pattern do we trigger)
- Two types of deadtime:
  - **Simple deadtime** - Limits the minimum time between consecutive L1Accepts.
  - **Complex deadtime** - Restricts number of L1Accepts in a given period to protect read-out buffers from trigger bursts.
- The complex deadtime increases exponentially with L1-trigger rate
- Deadtime reduces the time physics triggers are 'alive'

Lar Complex deadtime




- Aim to process events at an L1-trigger rate as high as possible, but keep deadtime reasonably low to achieve a high physics live fraction:
  Target L1A: >90kHz, physics deadtime <~3%

# High Level Trigger

Fast custom-made software on commercial CPUs

- 60000 real CPU cores (2023).
- Uses algorithms similar to offline to reconstruct objects
  - Fast reconstruction, usually guided by RoIs.
  - Precision, slower, reconstruction on full detector data and applies physics selection.
  - Only focus on few types of objects
  - Early rejection
- Once an event is accepted by the HLT, it is recorded and processed at Tier-0 and distributed to the GRID for physics analyses.

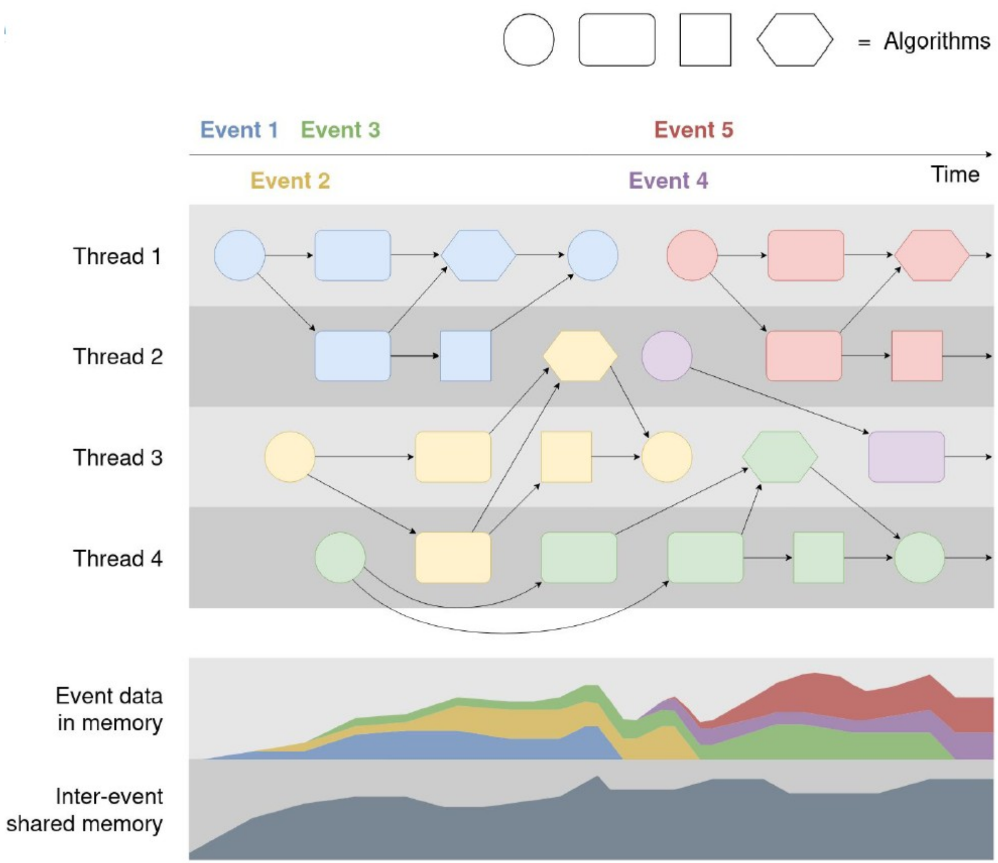
| year | 2018 | 2022 | 2023 |
|---|---|---|---|
| HS06 | 1.2M | 1.7M[1] | 2.0M |

[1]: 60% of racks replaced
HS06: benchmark used before HS23, definition here

# HLT code architecture

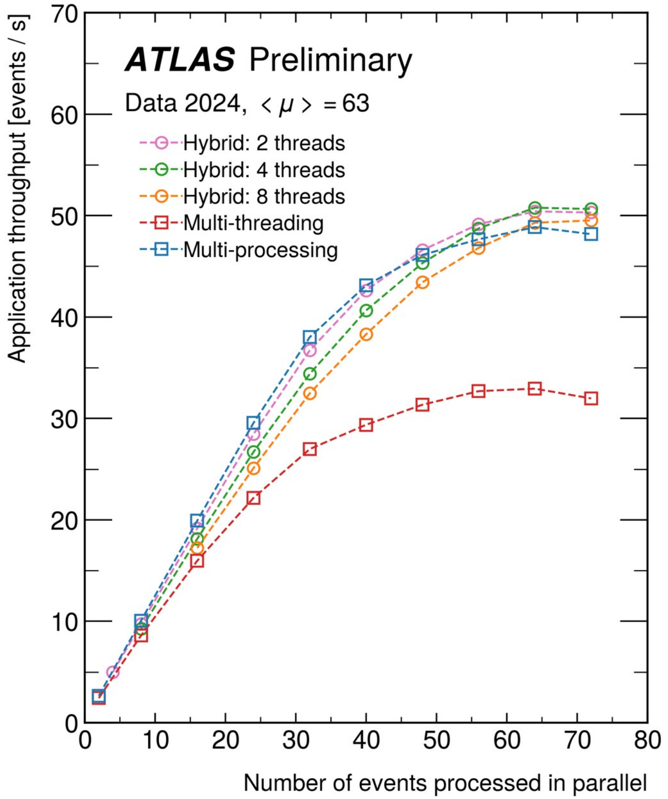
- The HLT was redesigned for Run 3 to share the same code with offline reconstruction
  - Support the **Multi-Threaded** mode
  - Reduce the memory footprint of the code
- **AthenaMT** offers three kinds of parallelism:
  - **inter-event**: multiple events are processed in parallel.
  - **intra-event**: multiple algorithms can run in parallel for an event.
  - **in-algorithm**: algorithms can utilize multi-threading and vectorisation.

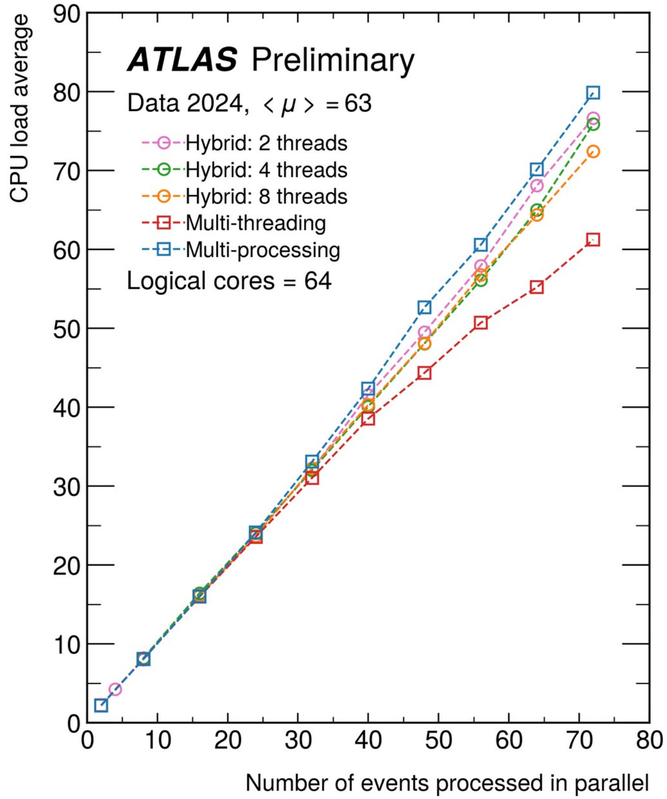
- The upgrade benefits:
  - Simplified maintenance of the code.
  - General performance improvements.
  - Integration of computing accelerator for future running periods.

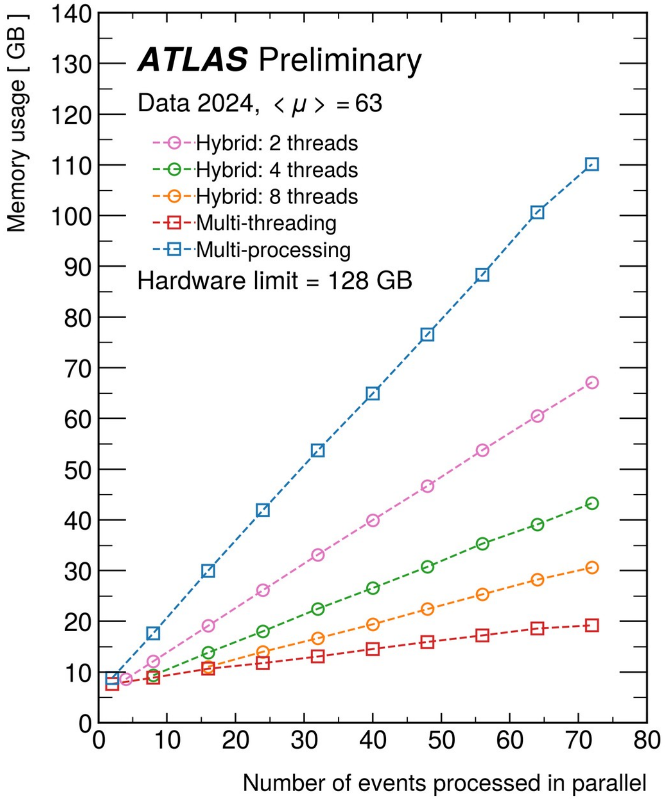


# HLT code architecture

- Multi-processing introduced for Run 2, Multi-Threading for Run 3.
- Despite being more sophisticated, a pure MT configuration show lower throughput than Multi-processing.
- During 2024 data-taking, the ATLAS HLT is using a **hybrid configuration with 16 forks and 4 threads.**
- Measurements were performed in a standalone local environment using a machine identical to those used in the ATLAS HLT (dual processor machine with 128 GB RAM and two AMD EPYC 7302 CPUs, where each CPU has 16 real cores with two hyper-threads per core, 64 logical cores in total, running Alma9 Linux).

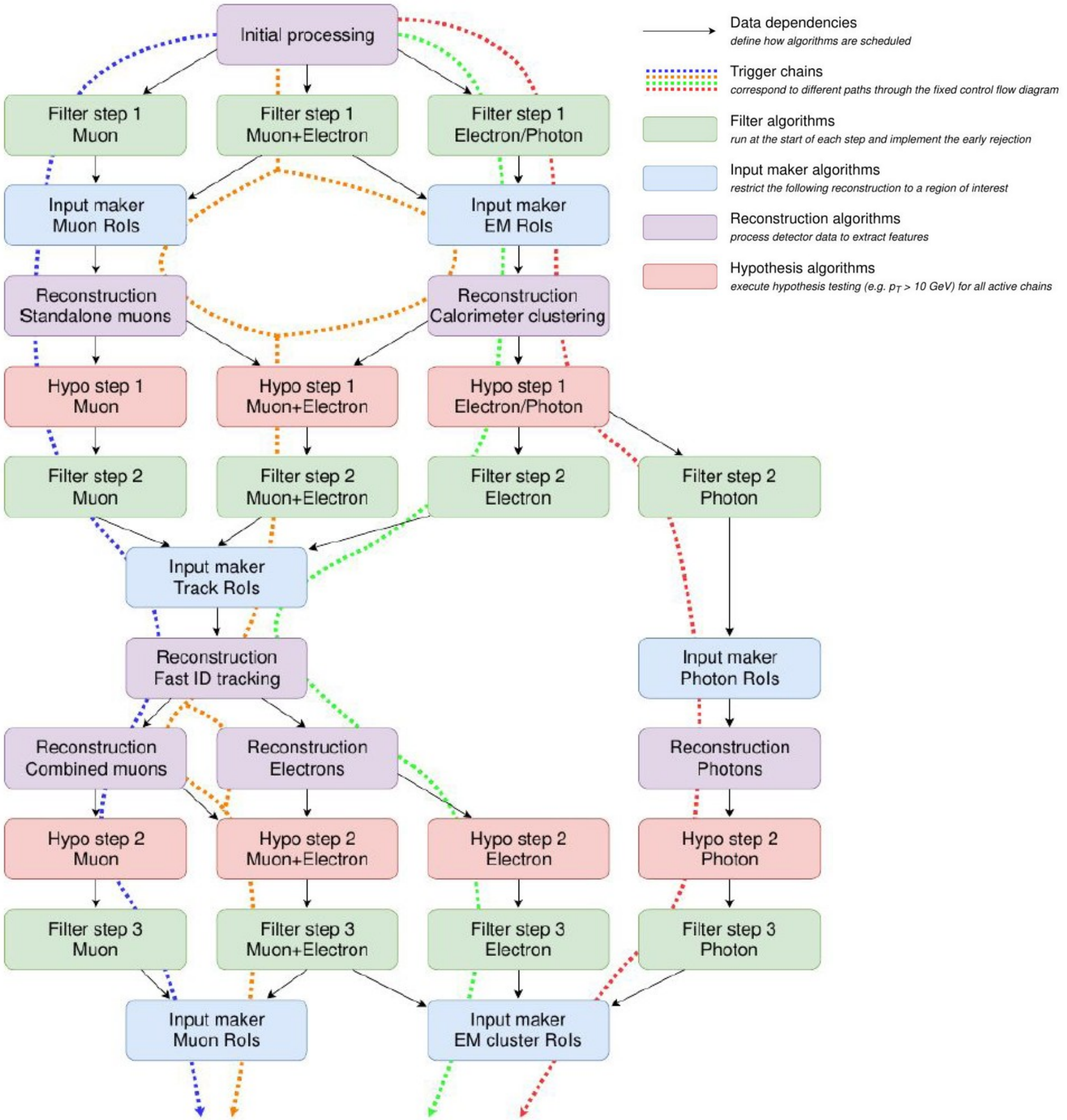


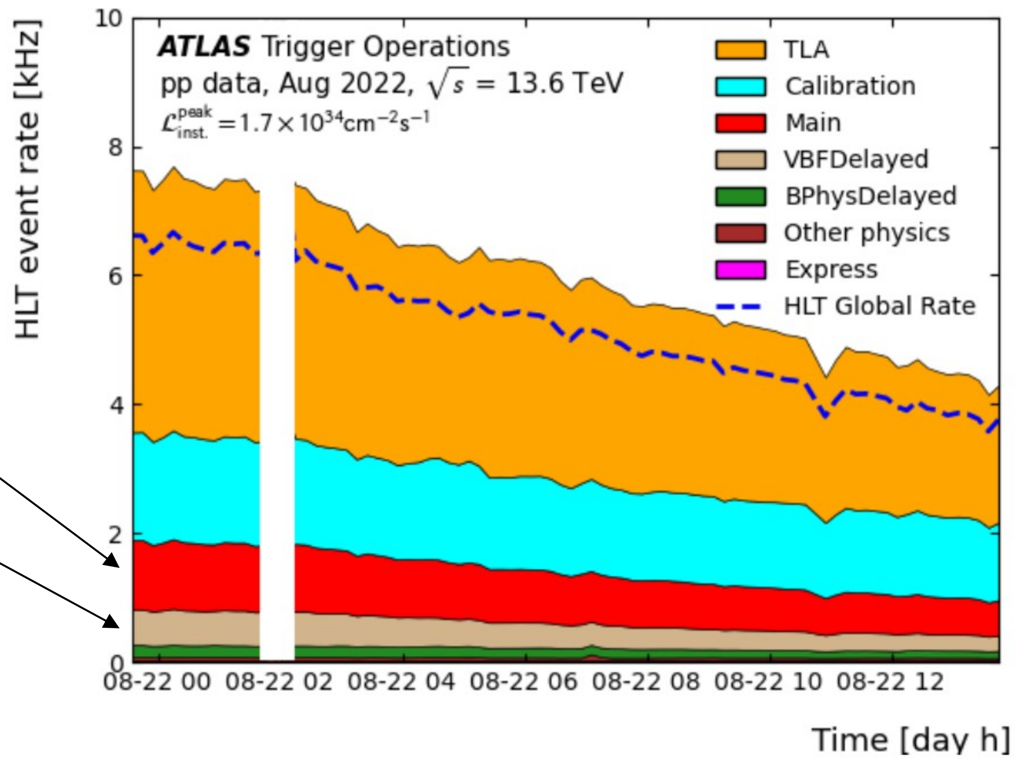
# HLT configuration

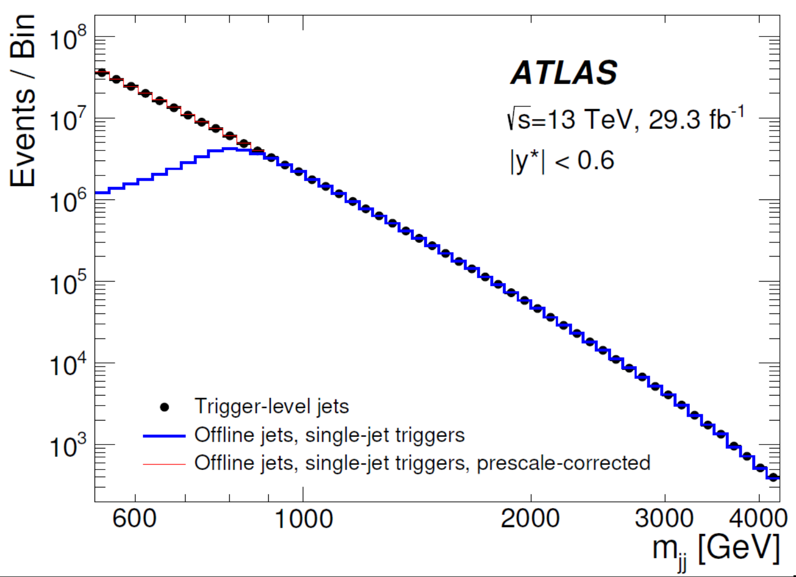
- The Run 3 HLT Control Flow is generated based on a list of algorithms organized in steps, performing reconstruction and selection.
  - Algorithms can be shared/reused
  - The steps are combined in chain and are organized in a selection menu
  - The configuration is stored in JSON blob format and can be provided transparently to HLT in different ways:
    - from a database,
    - from a file,
    - from a configuration in Python
    - from 'in-file meta-data' (mostly used for offline reconstruction)



# HLT streams

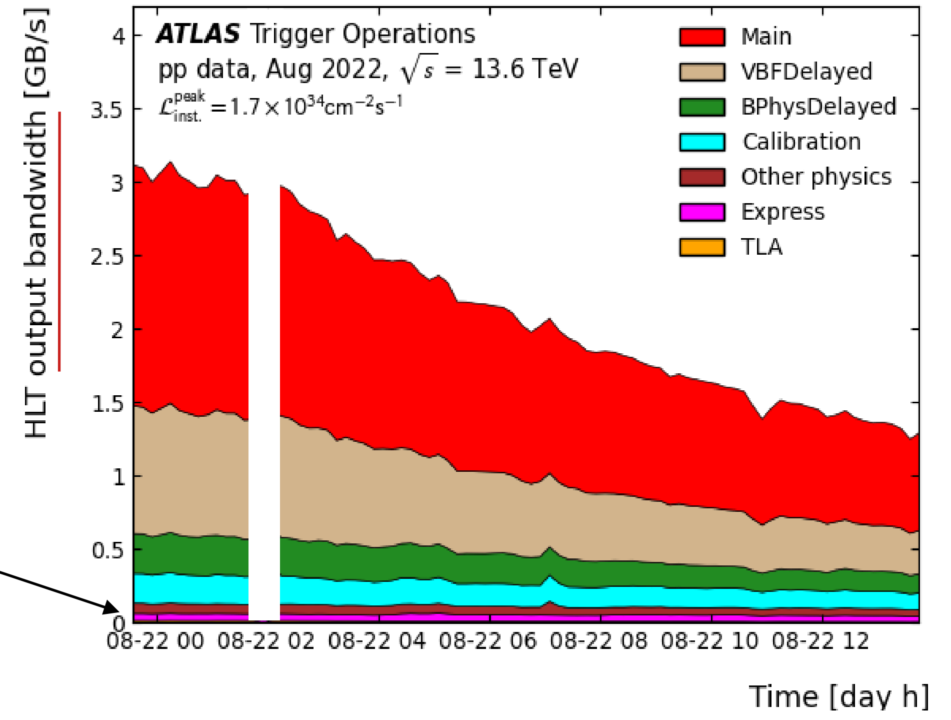
- HLT outputs are organized in “streams”, collection of events or event fragmentsorganized based on HLT decision
- Examples of streams in Run 3:
- **Physics streams**
  - Main stream — used for most physics analysis
  - Delayed streams (Bphys, VBF) — reconstructed during LHC downtime)
  - Trigger object Level Analysis (TLA) — only selected trigger level objects are written out, strong reduction of event size from 2 MB → 5 kB.
- **Express stream** — O(20 Hz), promptly reconstructed for DQ assessment, small sub-set of Main and Delayed streams



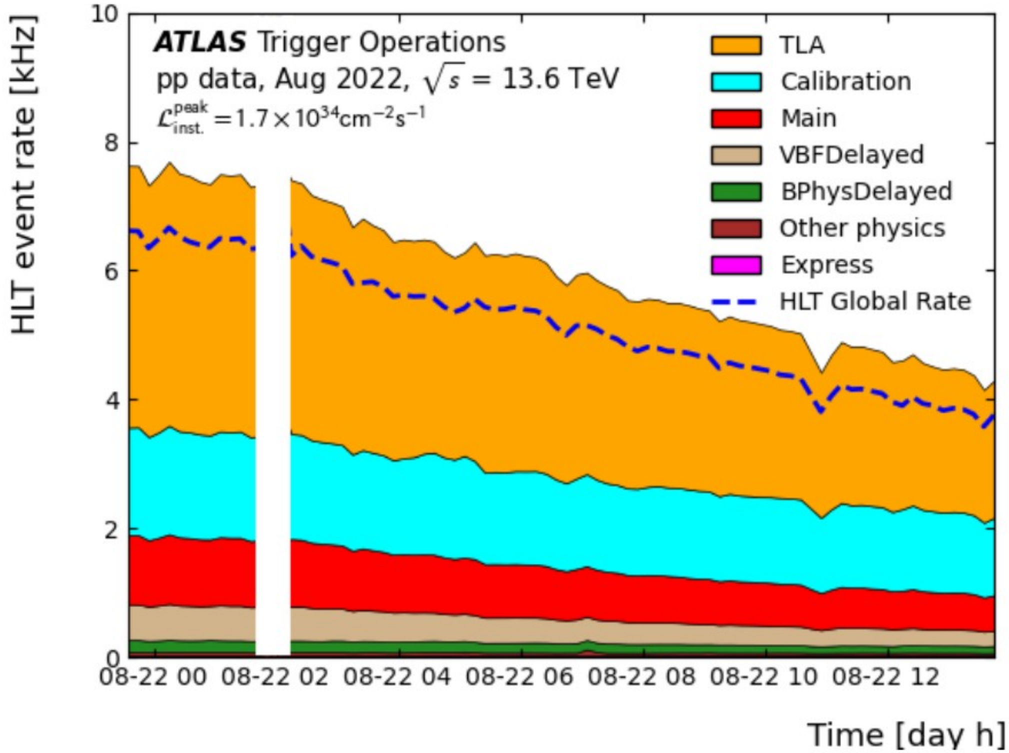
# HLT streams

- HLT outputs are organized in “streams”, collection of events or event fragmentsorganized based on HLT decision
- Examples of streams in Run 3:
- **Physics streams**
  - Main stream
  - Delayed streams (Bphys, VBF)
  - Trigger object Level Analysis (TLA) — despite the large rate, the TLA bandwidth is practically negligible.
- **Express stream**



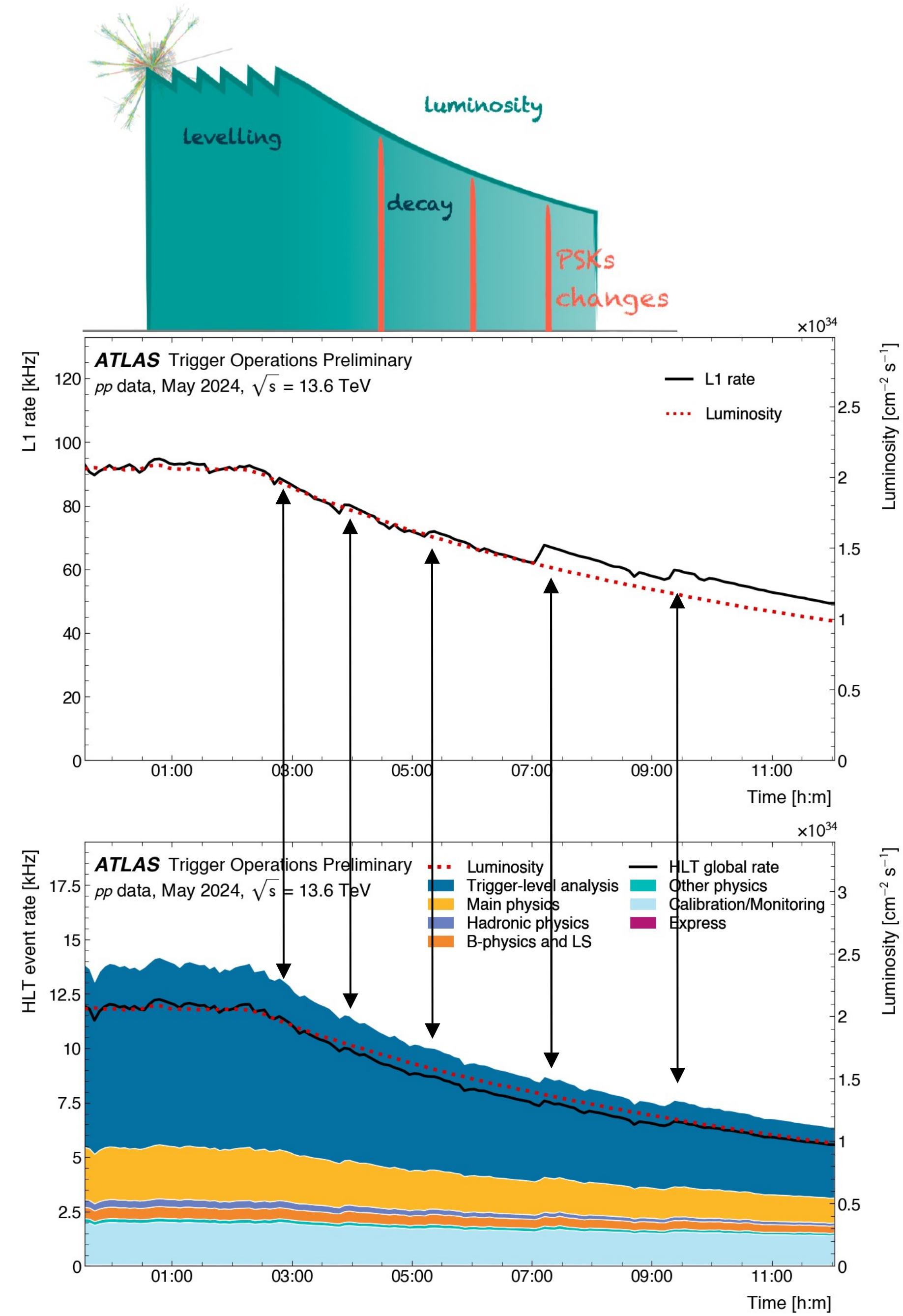
# HLT streams

- HLT outputs are organized in “streams”, collection of events or event fragmentsorganized based on HLT decision
- Examples of streams in Run 3:
- Physics streams
  - Main stream
  - Delayed streams (Bphys, VBF) —
  - Express stream
  - Trigger object Level Analysis (TLA)
- **Calibration stream**
  - minimum amount of information for detector calibration and monitorin→ Partial Event Building (PEB)
- **Debug stream**
  - Events without full trigger decision due to failures in a part of the HLT system (timeout, crashes...)



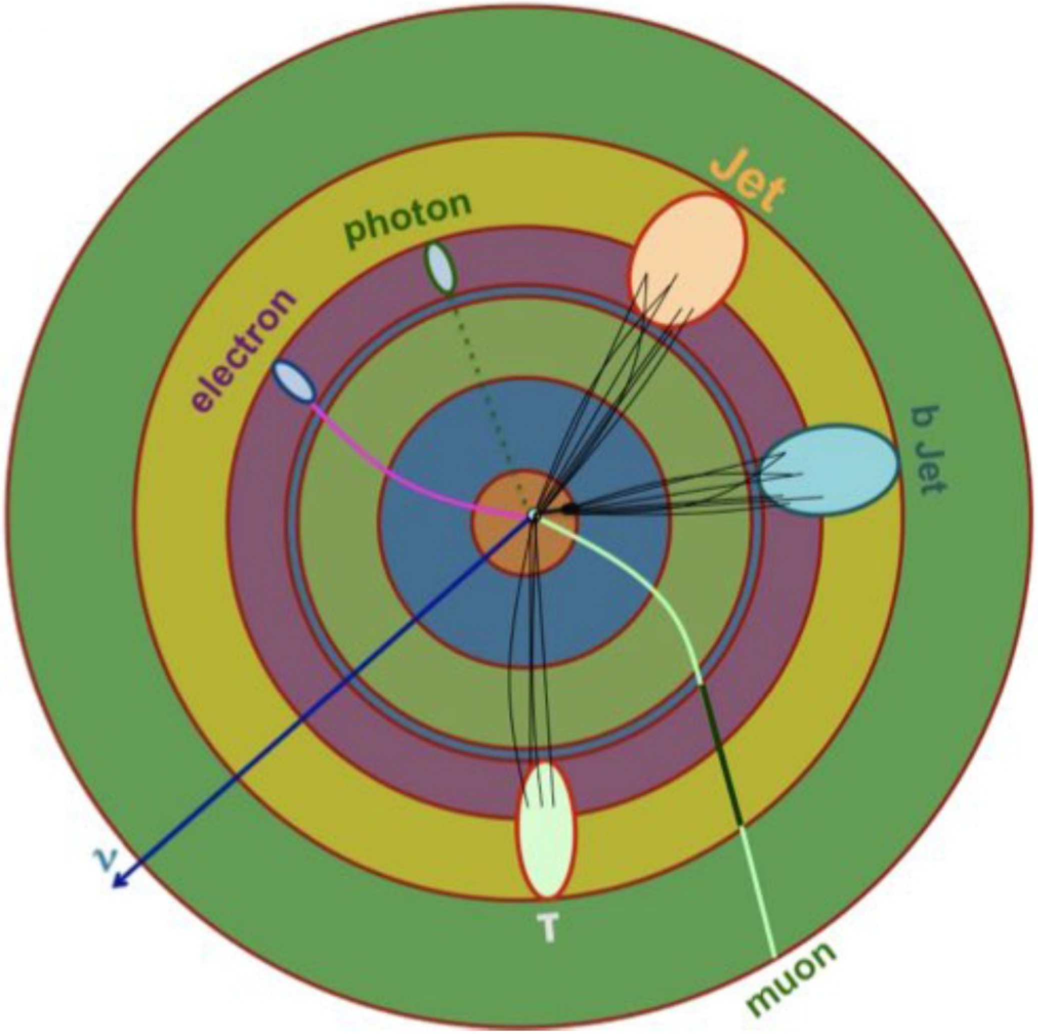
# Rate optimization during Fill

- At the start of the fill, luminosity is levelled.
- The, as the instantaneous luminosity decreases during a run, resources (CPU, bandwidth) can support lower thresholds.
- New triggers and prescales are adjusted throughout the run:
  - Based on preliminary performance studies of selection's cost.
  - Some of the chains are enable only in the end of the run.
  - The configuration changes are visible in the recorded rates of output streams.



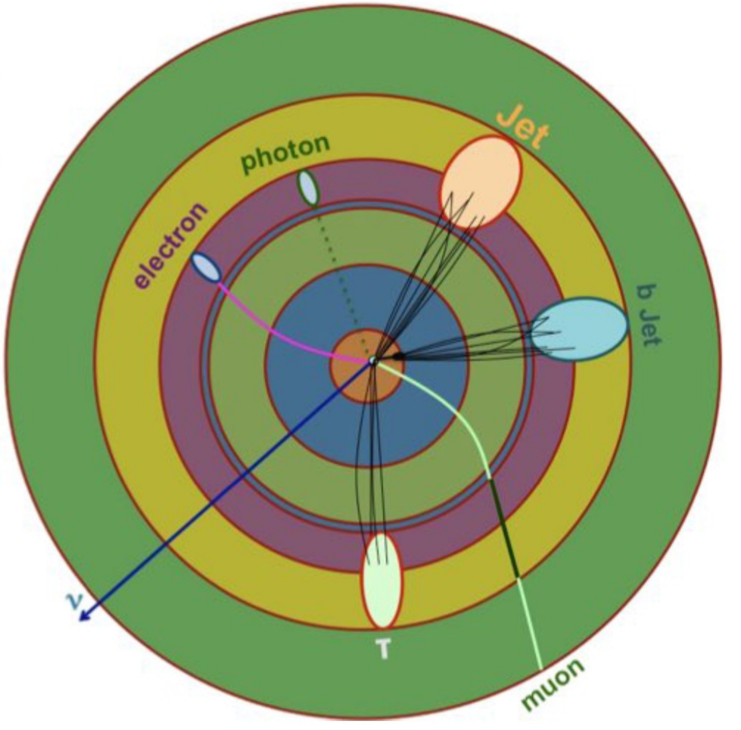
# Trigger signatures

- E/gamma
  - RoI-based fast and precision reconstruction steps
  - Getting closer to offline algorithms
  - Identification algorithms similar to offline
- Tau
  - Targets only hadronic tau decays
  - Calo-based preselection followed by fast and precise track reconstruction steps
  - RNN-based identification similar to offline
- Muon
  - RoI-based fast and precision track reconstruction
  - Matching of MS-based tracks with Idbased tracks in most cases
- Jet
  - Small(large)-radius jets using the anti-kt algorithm with R=0.4(1.0)
  - New for Run 3: moved from only calo-based topo-clustering inputs to PFOs exploiting full-scan tracking capabilities



# Trigger signatures

- MET
  - Various reconstruction algorithms provided, exploiting only calobased inputs or combination with full-scan tracking information.
- b-jet
  - RoI-based preselection followed by precision step combining jet reconstruction and primary vertex information.
- B-physics & light states
  - Primarily based on muons from B-hadron decay leading to pairs of close-by, soft muons.
  - High L1 rate (needs L1Topo) and high CPU needs.




| Signature | Rate per stream [Hz] | | |
|---|---|---|---|
| | Main | Delayed | TLA |
| Electron | 270 | | |
| Photon | 120 | | |
| Muon | 290 | | |
| Tau | 160 | | |
| Missing transverse momentum | 140 | | |
| Unconventional Tracking | 40 | | |
| *B*-physics and light states | | 240 | |
| Jet | 490 | 460 | 5000 |
| Jet with *b*-hadrons | 190 | 160 | |
| Combined | 240 | 50 | 830 |

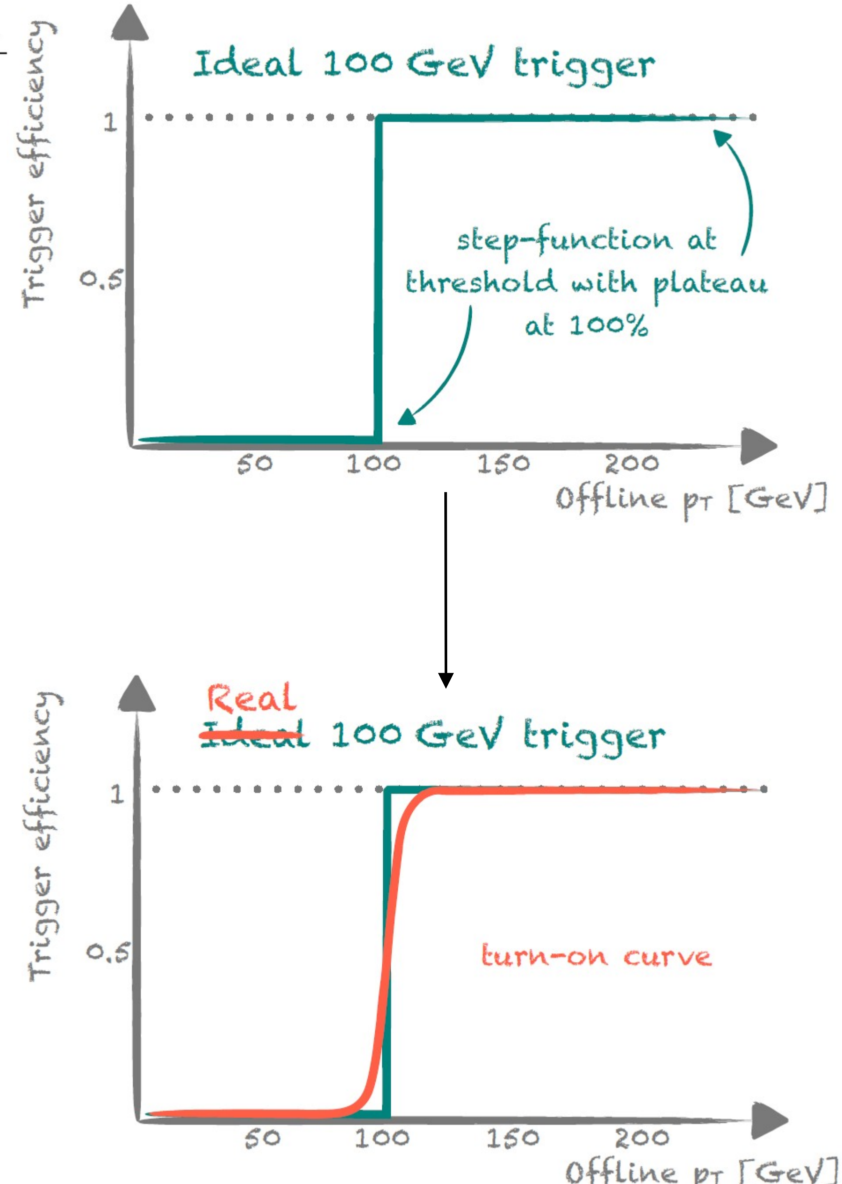
# Trigger and physics analysis

- Trigger efficiency usually measured w.r.t. offline reconstruction:

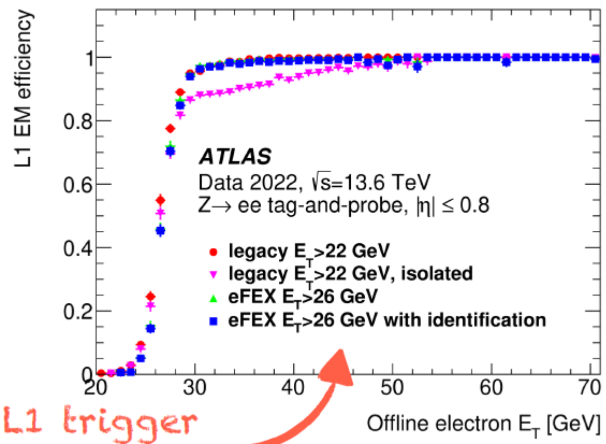
$$\epsilon_{trigger} = \frac{N_{trigger}}{N_{offline}}$$

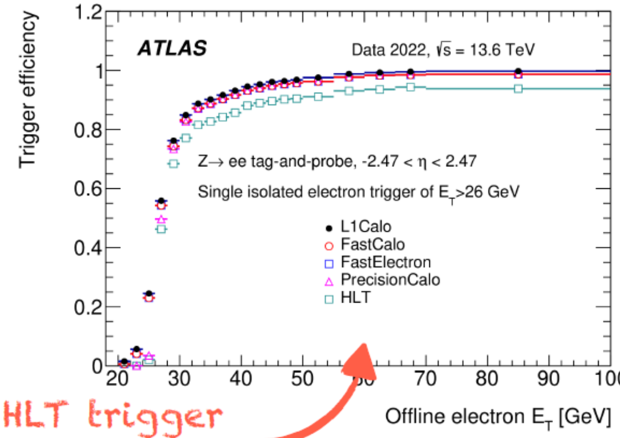
- Trigger efficiencies measured in data and used to correct MC simulation:
  - Tag-and-probe — Select on object triggered on (tag) and measure response of second particle (probe), e.g. $Z \rightarrow \mu\mu$
  - Boot-strap — Use looser (prescaled) trigger (e.g. 40 GeV jet to measure 60 GeV trigger efficiency)
  - Orthogonal trigger — Trigger on one physics signature, measure a different one

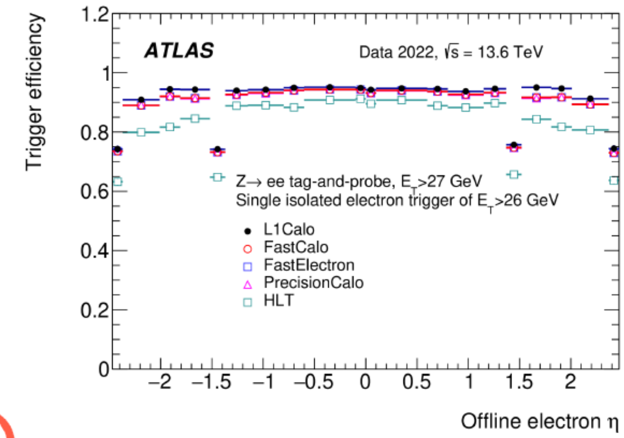
$$\epsilon_{trigger} = \frac{N_{trigger}}{N_{offline}}$$



# Trigger and physics analysis



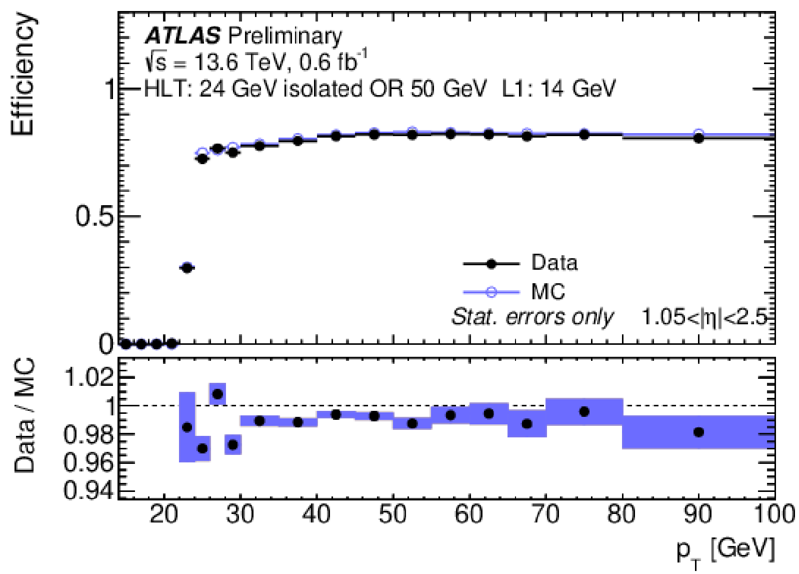



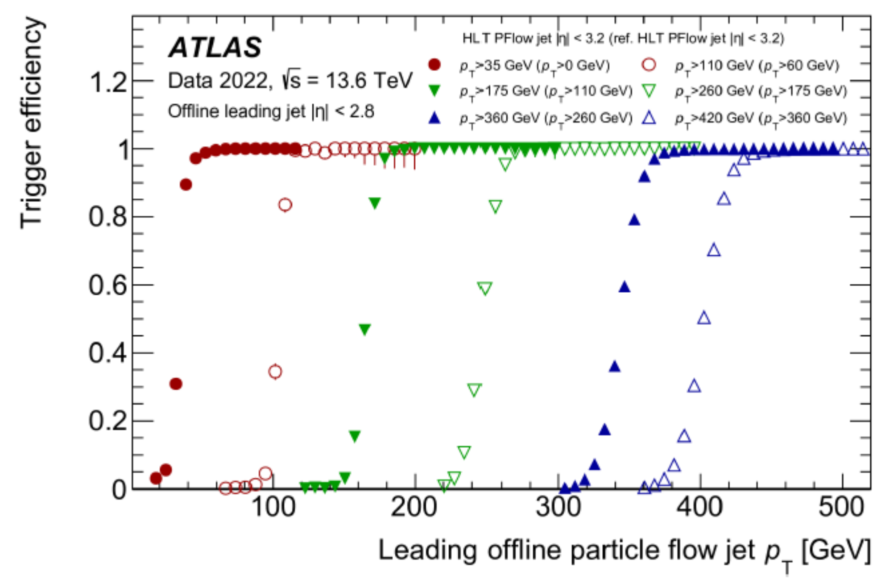




L1 trigger efficiency

HLT trigger efficiency

often w.r.t. $p_T$ or $E_T$ ... but doesn't have to be, e.g. $\eta$

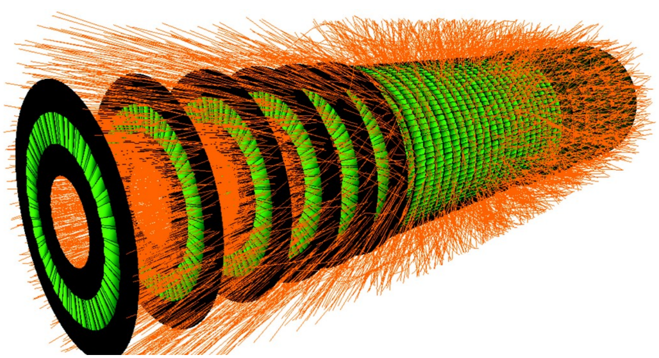


# HL-LHC Upgrade

- HL-LHC:
  - Data-taking: 2029-2042
  - Luminosity: from 2 → 7.5 $10^{34}$ $cm^{-2}s^{-1}$
  - Integrated luminosity: 300 $fb^{-1}$ → 3000 $fb^{-1}$
  - Pileup: from 20 → up to 200

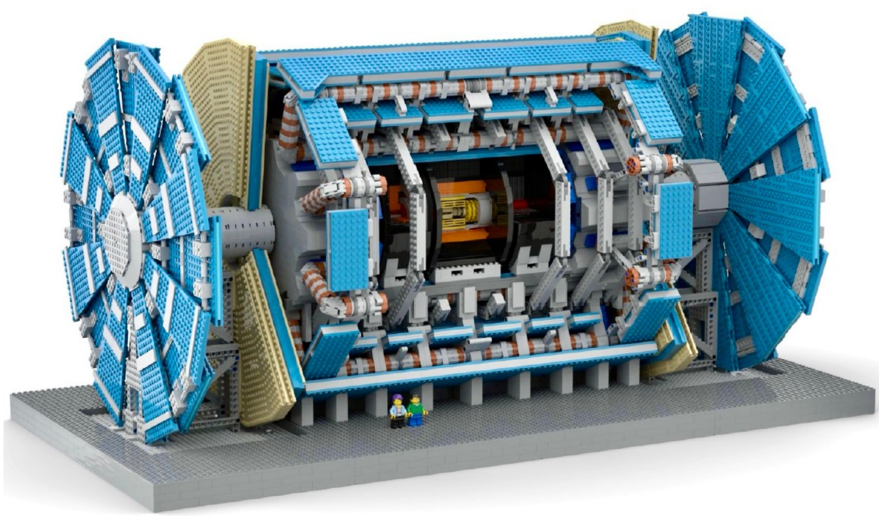
- ATLAS Upgrades:
  - New inner tracker (ITk, full silicon)
  - **New digital readout and trigger system**
  - Sub ns timing detector
  - New Muon chambers
  - etc...

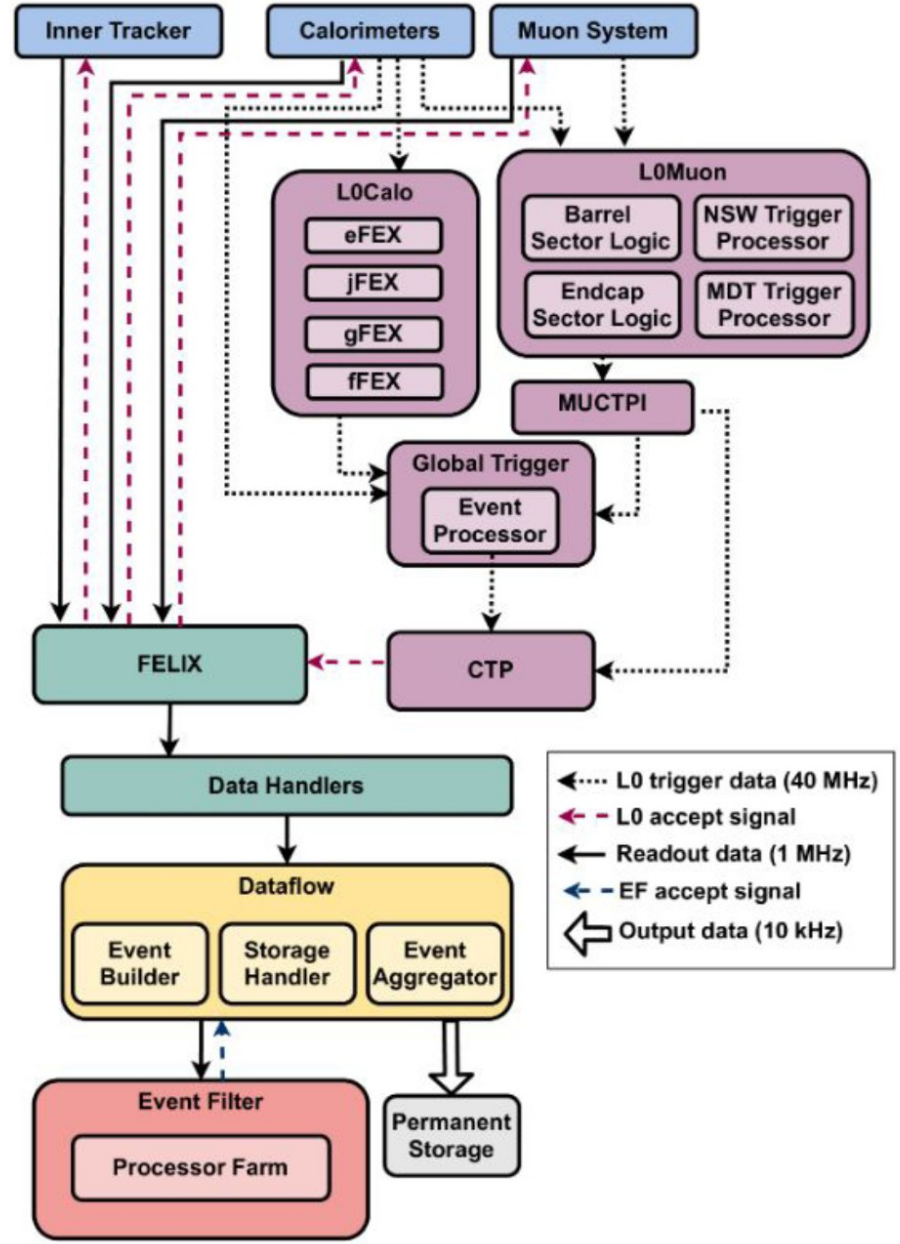


# TDAQ Upgrade

| | Run 3 | HL-LHC |
|---|---|---|
| Input trigger rate: | 40 MHz | 40 MHz |
| Output L0/1 rate: | 100 kHz | 1 MHz |
| Latency: | 2.5 µs | 10 µs |
| HLT input rate: | 100 kHz | 1 MHz |
| HLT output rate: | 3 kHz | 10 kHz |

- New Hardware-based L0 trigger.
- HLT on multiple types of computational units
  - Commodity CPU-servers
  - Possibly accelerators: **GPU, FPGA**
- Improve support for unconventional signatures:
- Large Radius Tracking, Disappearing tracks, etc.



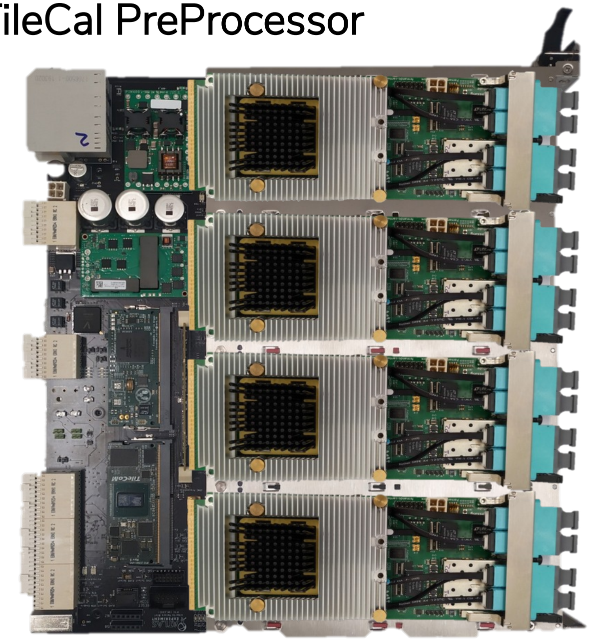


# L0 Calo

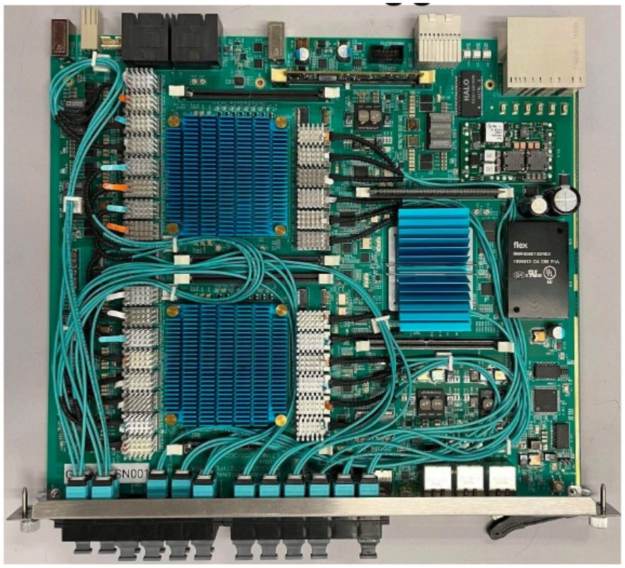
- Feature Extractors:
  - FPGA-based trigger boards optimized to trigger on different physics objects
  - Feature EXtractors from Run 3
    - e(lectron)FEX, j(et)FEX, g(lobal)FEX
    - Hardware retained with upgraded firmware
    - Needed modifications currently being evaluated
  - New forward Feature EXtractor (fFEX)
    - EM triggers for $|\eta| > 2.5$
    - Jet triggers for $|\eta| > 3.3$
    - Preliminary design being studied

- DAQ:
  - Unified backend electronics based on custom PCIe FPGA cards (FELIX)

TileCal PreProcessor



L0 Global Trigger



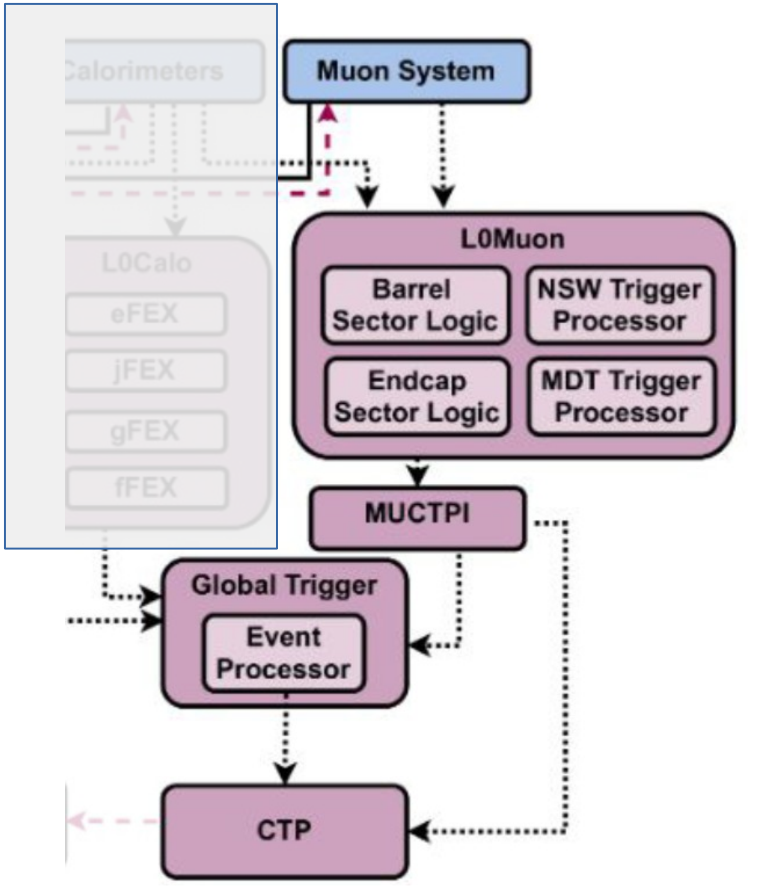
Felix

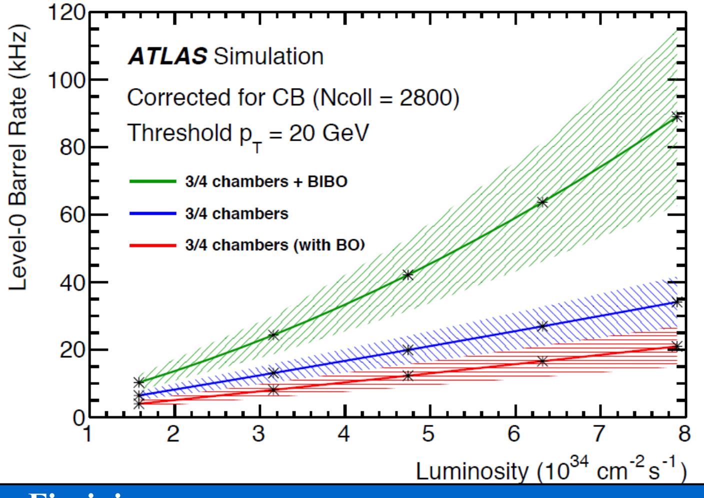
# L0 Muon and CTP

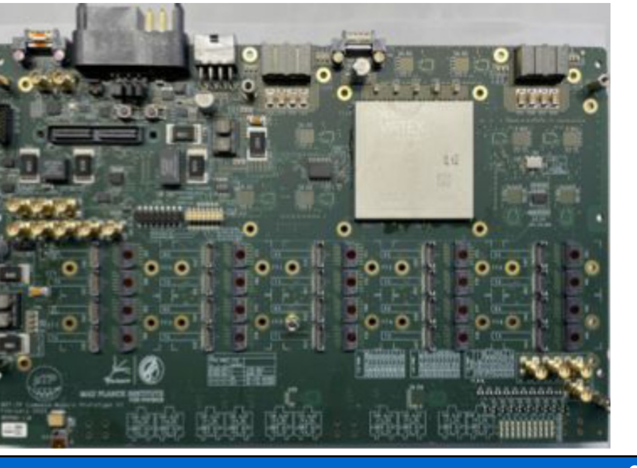
- L0 Muon ATCA:
  - Blades based on a common open source platform (Apollo), that simplifies custom ATCA blade design.

- L0 Muon Trigger Processors:
  - Receive data from precision muon chambers and new small wheel muon chambers.
  - Interact with sector logic.

- CTP:
  - Number of L0 triggers: 512 → 1024
  - CTP drives the Trigger, Timing and Control (TTC) system network



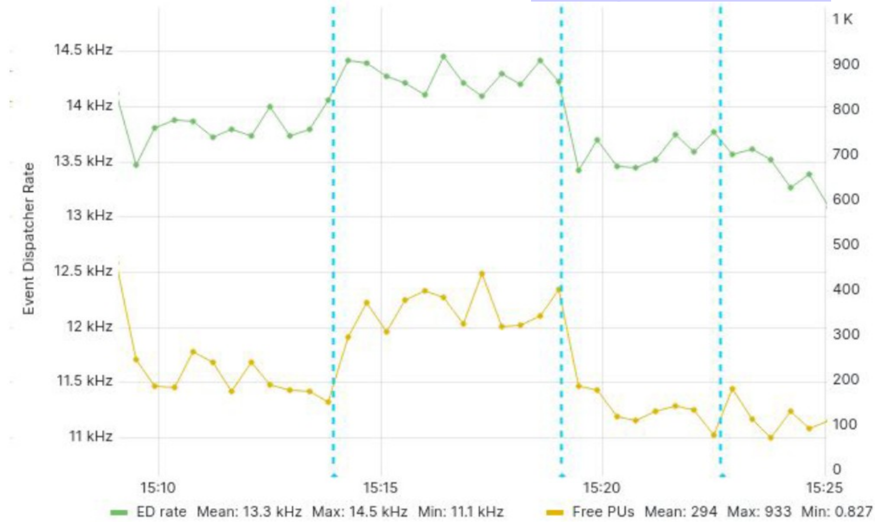



# HLT

Full event building at 1 MHz

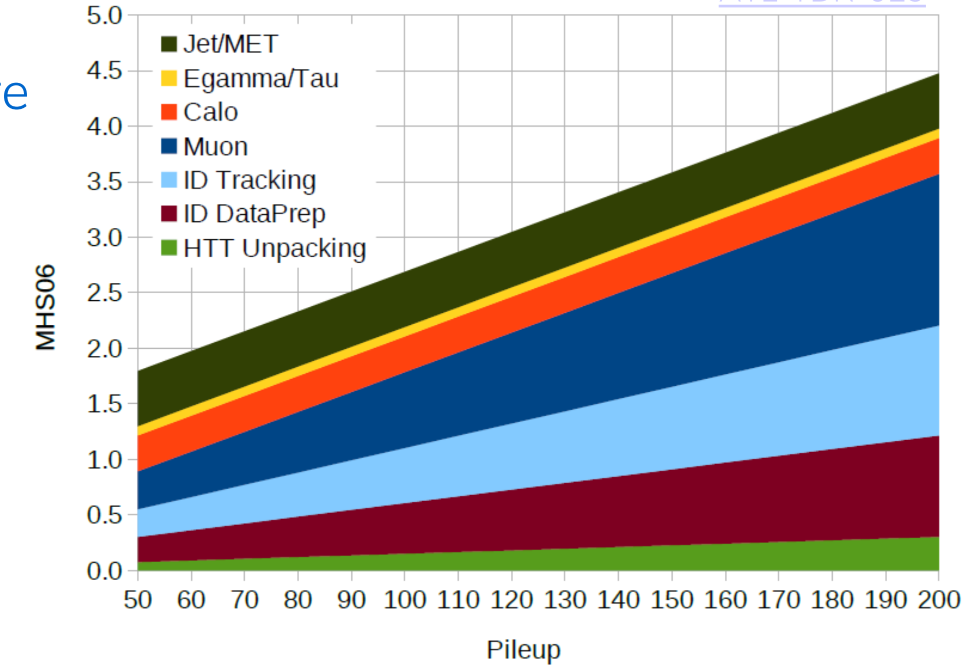
- 40 Tbps (5 TB/s)
- Heterogeneous commodity computing system.
- Offline-like algorithms
- Tracking is the most intensive computational task @HLT.
- Under evaluation the possibility to use accelerators (GPUs or FPGAs)

A. Negri, CHEP 2022




For the HLT tracking, use of COTS hardware is planned . Either pure software solution, or GPU or FPGA card acceleration (under evaluation).

ATL-TDR-029

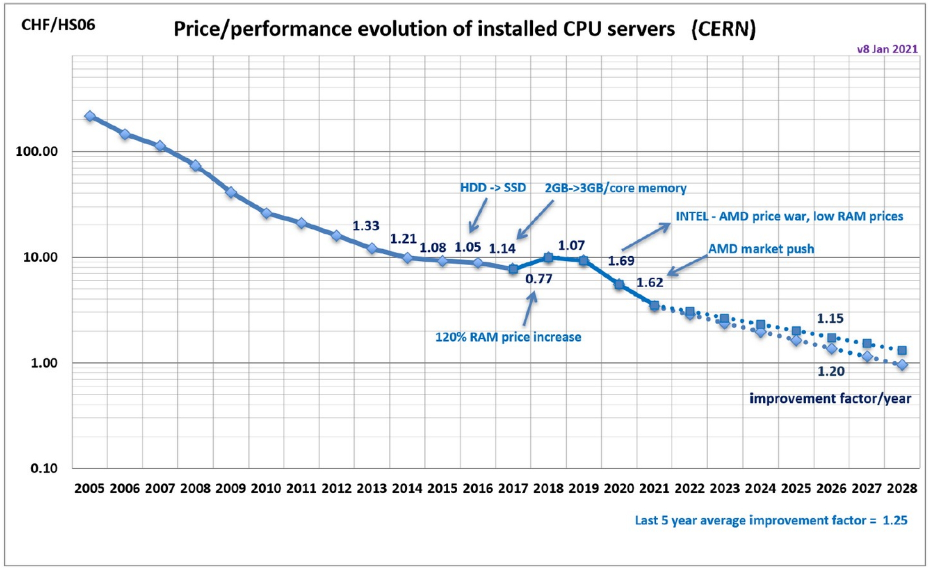



# CPU vs Accelerators for HLT

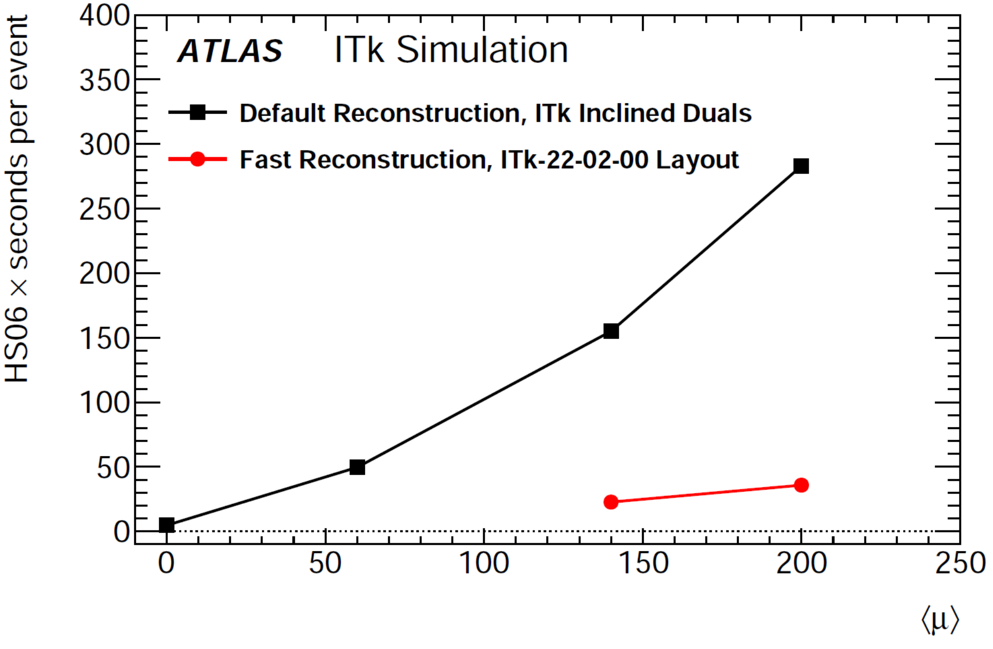
HLT for HL-LHC is based on Commodity hardware:

- Fully based on CPU: 7.8/11.4 MHS06 for Run 4/5
- Possibly w/ accelerators: GPU, FPGA
  - Preliminary feasibility studies
  - CPU showed x8 speed-up
  - Use of GPU/FPGA looked promising
- First demonstrators started using Fast Reconstruction:
  - tracking, muon, calorimeter
- Technology decision about the use of accelerators soon (in 2025).

ATL-TDR-029-ADD1



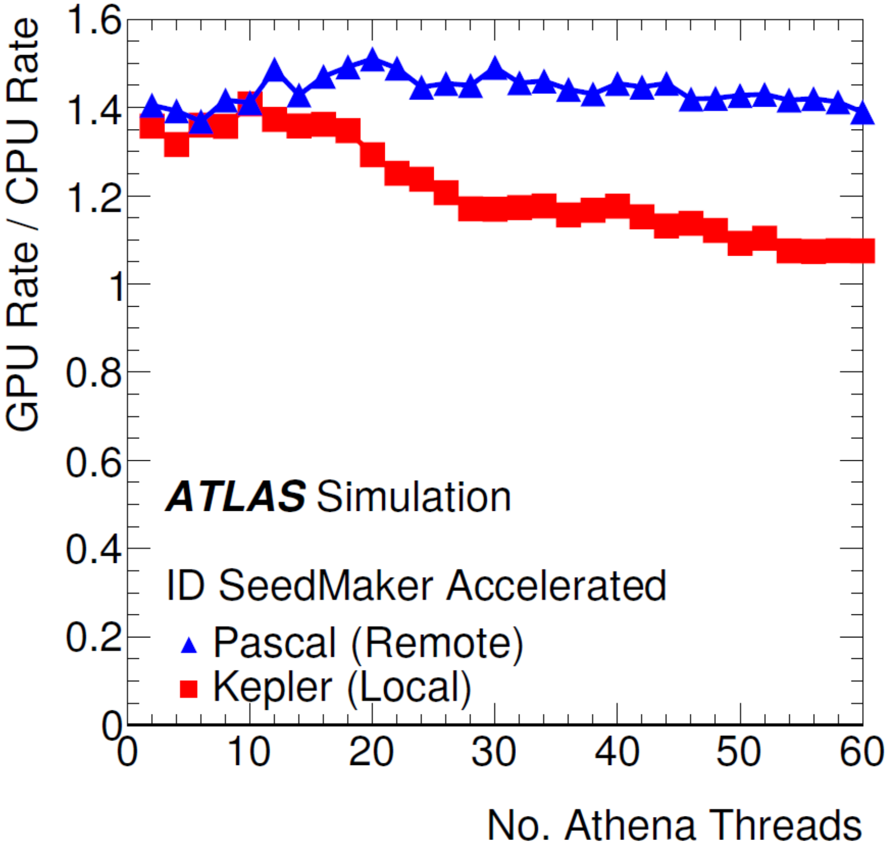

ATL-TDR-029-ADD1

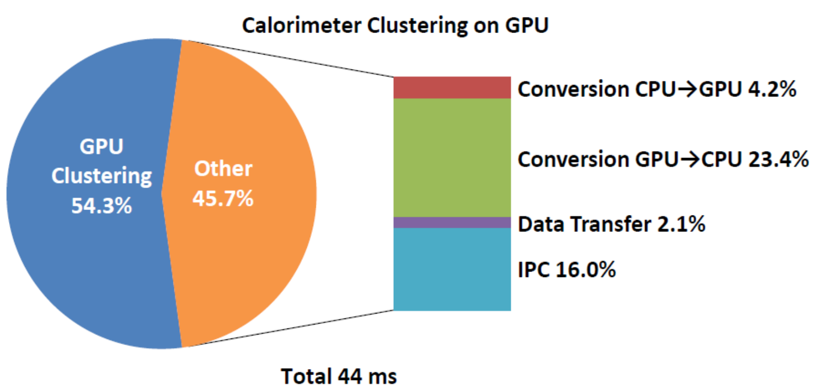


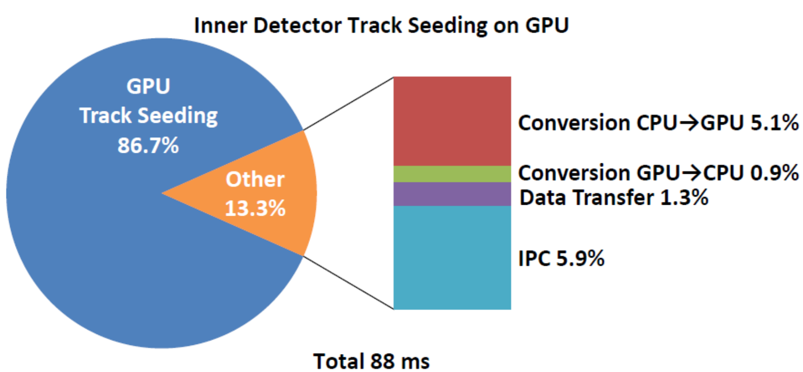
# GPU demonstrator for HLT

GPUs Preliminary tests performed:
- 4 K80 GPUs (Kepler) on the same PCIe bus of the CPU
- 1 GTX1080 (Pascal) on a different machine, 10 Gb/s connection.

- Speed-up:
  - ID track-seeding 28 (5.8) for Pascal (Kepler) GPU wrt CPU.
  - Overheads for data conversion and interprocess comm. reduce this to 15 (5) for the Pascal (Kepler).
  - In terms of events throughput the increase is 40% (15%).



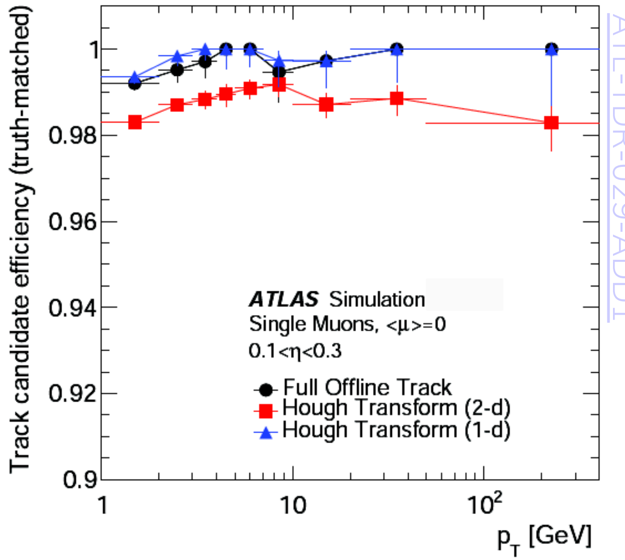



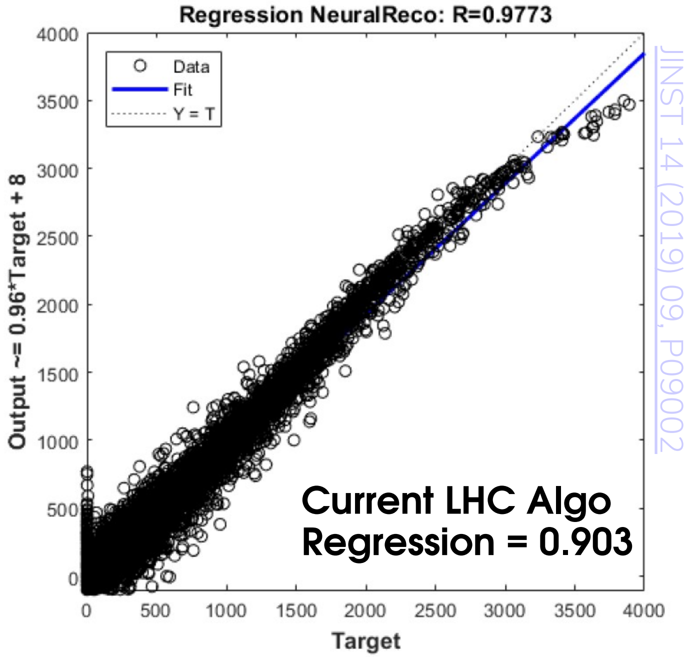
# FPGA demonstrator for HLT

- FPGA demonstrator based on Alveo U250.
- Hough Transform for Pattern Recognition.
- Graph Neural Networks (GNNs) will be considered in the future.
- High-level Synthesis (HLS) tools for FPGAs greatly facilitate the translation of new algorithm ideas into digital implementations.
- Total power of an FPGA-based heterogeneous system for effective tracking rate of 150 kHz (full-scan) + 50 kHz (regional):
  - 0.08-0.18 MW for FPGA+ 0.28/0.48 MW in Run 4/Run 5 for CPU.
  - CPU-only HLT would require 1.9-2.3 MW.

| | Run 4 | Run 5 |
|---|---|---|
| # of Accelerator Cards | 270-680 | |
| CPU resource requirement [MHS06] | 1.1-1.7 | 1.6-2.4 |
| Accelerator Power [MW] | 0.08-0.18 | |
| CPU Power [MW] | 0.28-0.42 | 0.32-0.48 |
| Total Power [MW] | 0.4-0.6 | 0.4-0.7 |




(a) *Muons in* $0.1 < \eta < 0.3$

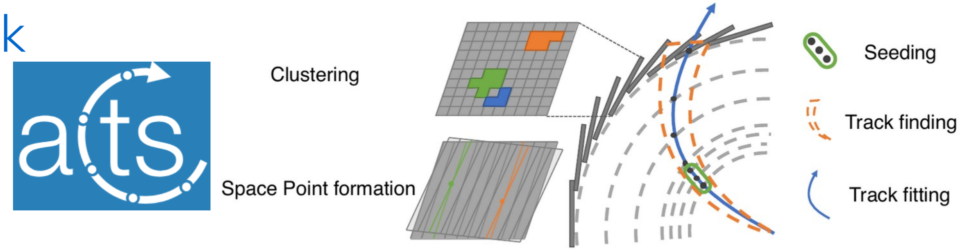


# ACTS

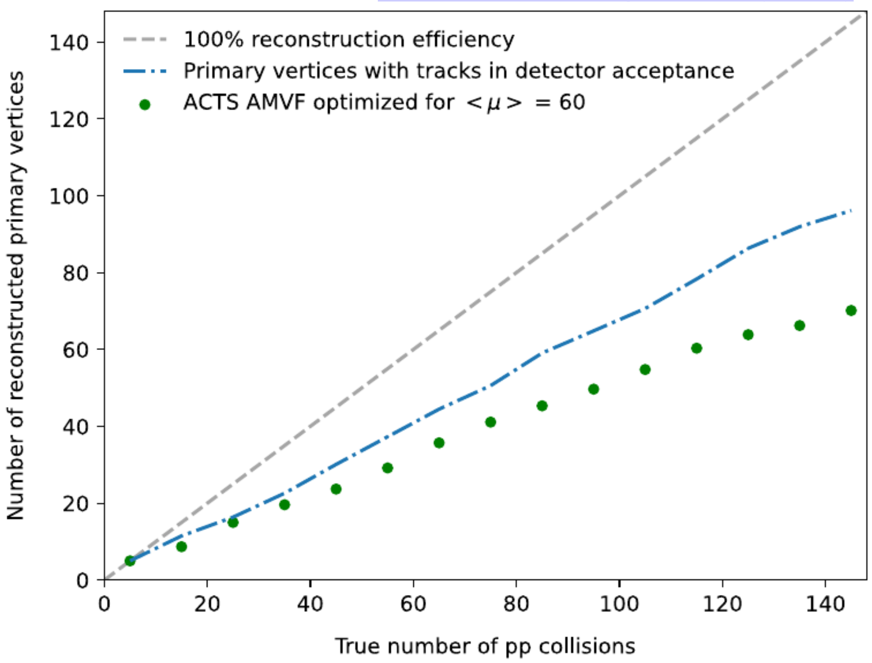
Plan to use [Acts Common Tracking Software]

Fully multi-threading ready code base for track reconstruction

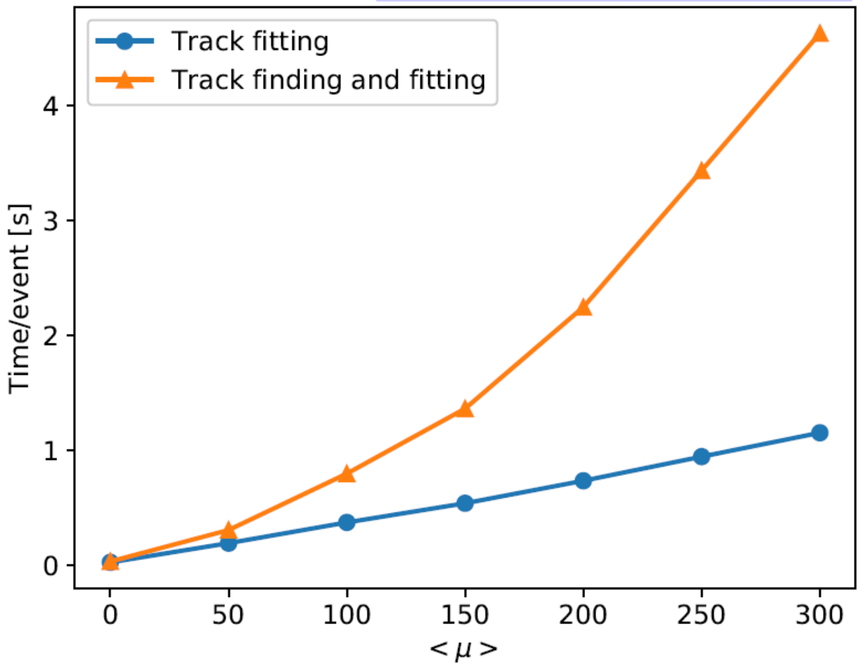
- Experiment independent toolkit for track reconstruction
- Support for accelerators and heterogeneous options




Comput Softw Big Sci 6, 8 (2022)

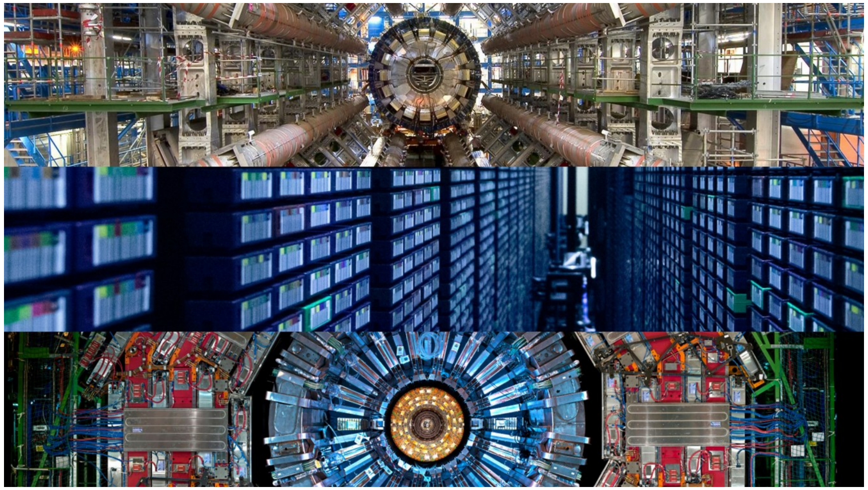



Comput Softw Big Sci 6, 8 (2022)



# NextGen Triggers

- The recently launched Next-Generation Triggers project is set to remarkably increase the efficiency, sensitivity and modelling of CERN experiments.
- The key objective of the five-year NextGen project is to get more physics information out of the HL-LHC data.
- The foundations of the NextGen project were laid in 2022 when a group of private donors, including former Google CEO Eric Schmidt, visited CERN.
- This visit evolved into an agreement with the Eric and Wendy Schmidt Fund for Strategic Innovation, approved by the CERN Council in October 2023, to fund a project for the future trigger systems at the HL-LHC and beyond:
- The intellectual property generated as part of the NextGen Triggers project, owned by CERN, will be released under CERN Open Science Policy.



# Summary

- ATLAS is successfully running a high speed and bandwidth trigger system
  - Trigger staged in a L1 hardware and a HLT software system.
  - Rate reduction 40 MHz → 100 kHz → 3 kHz
  - Additional data streams (TLA and delayed streams) and topological requirements are used to further enhance the physics reach of the experiment.
- HL-LHC upgrade requires more speed and bandwidth.
  - Current approach is based on a mix of commodity and custom solutions
  - Level-0 trigger requires 40 MHz → 1 MHz rate reduction.
  - HLT is investigating accelerator options (GPUs, FPGAs), technology decision in 2025.
  - Support for unconventional signatures will be further expanded for HL-LHC.

# Bibliography


- ATLAS Collaboration, The ATLAS Trigger System for LHC Run 3 and Trigger performance in 2022, JINST 19 (2024) P06029
- A. Poreba, Operational experience with the new ATLAS HLT framework for LHC Run 3, CHEP2023
- S. Morgenstern, ATLAS Trigger System, Lecture
- ATLAS Collaboration, ATLAS TDAQ Phase II Upgrade, ATL-TDR-029
- ATLAS Collaboration, Technical Design Report for the Phase-II Upgrade of the ATLAS Trigger and Data Acquisition System - EF Tracking Amendment, ATL-TDR-029-ADD1
- A. Negri, ATLAS Trigger and Data Acquisition Upgrades for the High Luminosity LHC, CHEP2023
- JL Ortiz et al, FPGA implementation of a deep learning algorithm for real-time signal reconstruction in particle detectors under high pile-up conditions, JINST 14 (2019) 09, P09002
- X. Ai et al,. A Common Tracking Software Project, Comput Softw Big Sci 6, 8 (2022)