GPU-Accelerated Online Reconstruction in ALICE

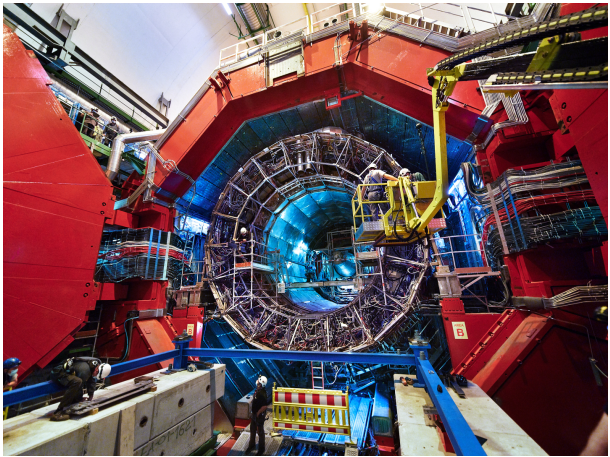
Felix Weiglhofer on behalf of the ALICE collaboration

CERN

ML4JETS 2026
18 September 2026

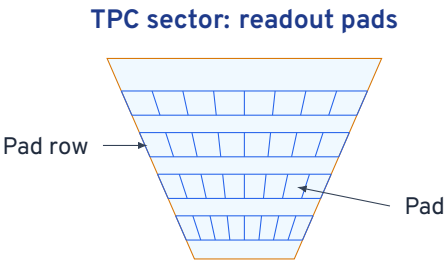
ALICE: A Large Ion Collider Experiment

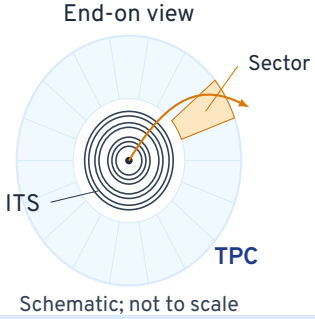
- LHC experiment studying strongly interacting matter, including the quark–gluon plasma
- Wide physics programme including pp, Pb–Pb and lighter ions such as oxygen
- Thousands of charged particles in a central Pb–Pb collision: reconstruct their trajectories, momenta and identities



The TPC inside the ALICE magnet.

Tracking Detectors: TPC and ITS





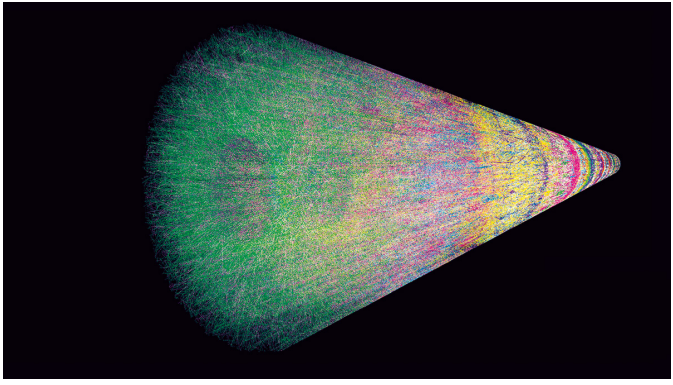
Time Projection Chamber (TPC)

Main tracking detector: many measurements per track. Charge gives dE/dx, helping distinguish particle species at a given momentum.

Inner Tracking System (ITS)

Seven silicon-pixel layers close to the beam. Precision tracking and reconstruction of collision and decay vertices.

Continuous Readout in Run 3



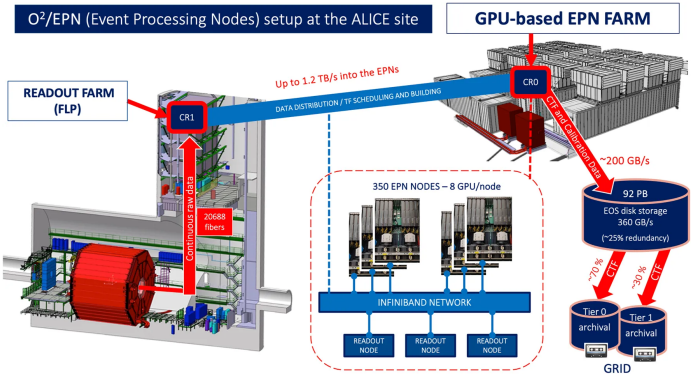
Pb–Pb at 50 kHz, 2 ms of TPC data. Different colours show different overlapping events.

- Pb–Pb collision rate: $\sim$8 kHz in Run 2 $\rightarrow$ up to 50 kHz in Run 3
- Collisions every $\sim$20 µs; TPC signals span $\sim$100 µs
- A hardware trigger cannot isolate one collision's TPC signals
- Read out continuously; reconstruct overlapping collisions in software

- Run 2: triggered selection; Run 3: all Pb–Pb events stored in compressed form

Run 3 Online Reconstruction

Online Processing and Data Reduction



- Reconstructed tracks enable TPC compression
- **Compressed time frames:** TPC clusters and other detector data, for later reconstruction with final calibrations

TPC workload

≈99%

of online reconstruction workload

CPU-only reference; excludes calibration and quality monitoring.

Online Compute Farm

The Run 3 online farm

Servers

350

8 GPUs per server

- TPC reco is massively data-parallel and bandwidth-bound, ideal for GPUs
- One time frame per GPU, round-robin
- About 4 s per 2.8 ms Pb–Pb time frame
- Per server: 64–96 CPU cores to feed the GPUs, 512 GB–1 TB RAM

GPU EVOLUTION IN ALICE

2010

64 × NVIDIA GTX 480

Run 1 (TPC sector tracking)

→

2015

180 × AMD S9000

Run 2 (TPC sector tracking)

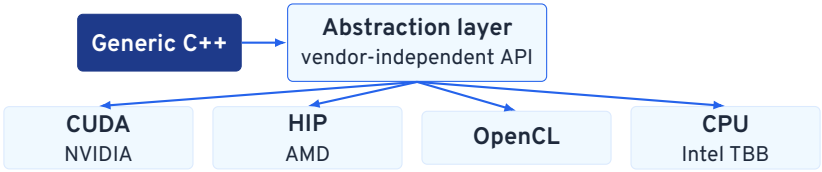
→

Since 2019

2,800 × AMD MI50/MI100

Run 3 (Full TPC reco)

GPU Framework: Portable Generic C++



GPU reconstruction process

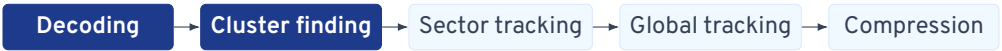
- Consecutive stages run in one process; intermediate data stays on the device
- Raw data streamed in, hiding the host-to-GPU transfer; all clusters of a time frame must fit in GPU memory
- Custom allocator reuses buffers across steps

ALICE O^2 framework

- Each workflow task is its own process; the whole GPU reconstruction is one of them
- Tasks exchange references to shared memory on a node
- Same framework used online and offline processing

Run-time compilation: kernels are recompiled at startup with the configuration as compile-time constants. Unused code paths are removed, 5–20% faster.

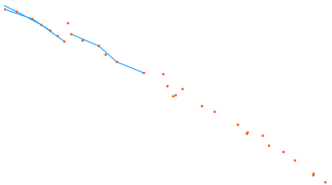
Decoding and Cluster Finding



- Raw ADC values are decoded and calibrated
- Charge on neighbouring pads and time bins is merged into clusters
- A cluster sits on one pad row and stores position and width in pad and time, plus its total charge

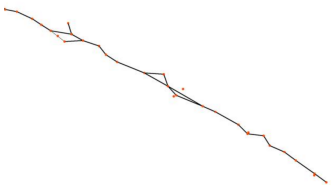
Tracking within a Sector

Decoding → Cluster finding → **Sector tracking** → Global tracking → Compression



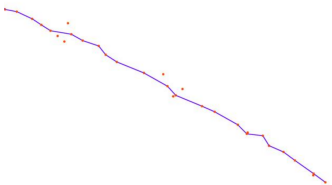
Seeding

A cellular automaton links neighbouring clusters into track segments.



Track following

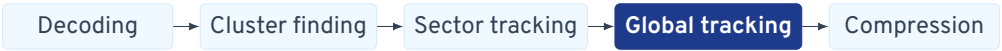
A Kalman filter extends each segment row by row.



Ambiguity solving

Duplicate tracks are removed, clusters are assigned to the best candidate.

Global Tracking



Sector tracks are combined into global TPC tracks.

Merging

- Track segments are joined across sector boundaries and across the central electrode

Refit

- Each merged track is refitted with a Kalman filter over all its clusters

Loopers

- Low-momentum particles curl into loops; their legs are linked into one track

Compression

Decoding → Cluster finding → Sector tracking → Global tracking → **Compression**

Cluster removal

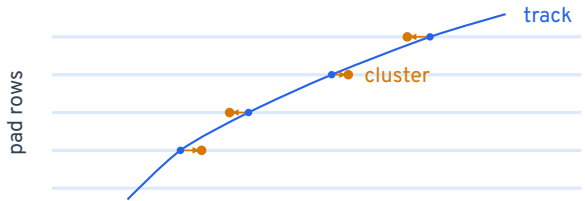
- Clusters not needed for physics are dropped

Track model compression

- Cluster positions stored as residuals relative to the reconstructed track

Entropy encoding

- Clusters copied back to the host
- Custom rANS coder on the CPU, close to the entropy limit



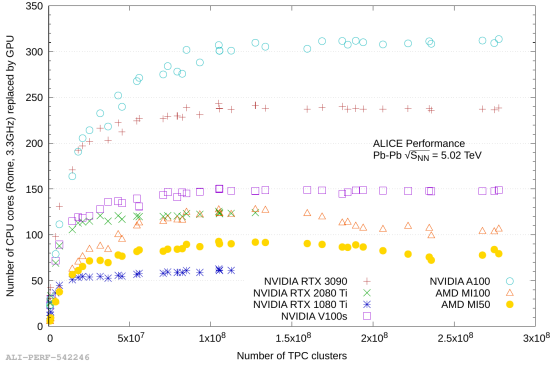
Residuals are small and centred around zero, so they have low entropy and compress well.

Why Do These Algorithms Fit GPUs?

Each stage exposes substantial parallelism.

- **Cluster finding:** Parallel across ADC values or cluster candidates
- **Tracking and compression:** Parallel across clusters, track segments, or tracks, depending on the stage
- **Decoding:** Parallel across data packets, with threads within a GPU warp sharing the work of decoding each packet

Performance at the Start of Run 3



AMD Rome CPU (@3.3 GHz) cores replaced by one GPU for the full TPC processing, corrected for the cores needed to drive the GPU.

One AMD MI50 replaced

80

CPU cores at Pb–Pb occupancy

CPU-only alternative

$>$2,000

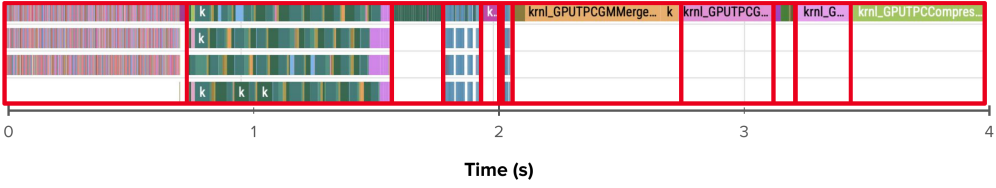
64-core servers would have been needed, not 350

G. Eulisse, D. Rohr, arXiv:2402.01205

Developments since the start of Run 3

Parameter Optimization

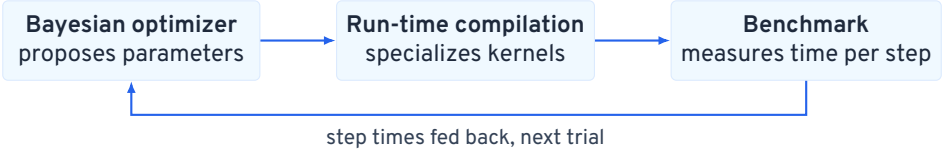
- 50 parameters control launch geometry, occupancy, shared memory, and algorithms
- Kernels that run concurrently share GPU resources and must be tuned together
- Split the parameters into smaller, independent sets—one per processing step—and optimize each set separately



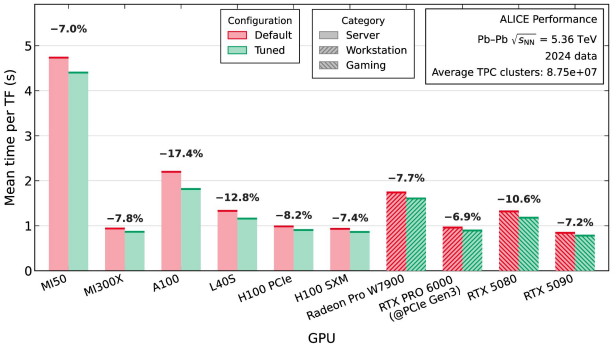
GPU kernel timeline for one time frame. Each red box groups kernels into a processing step with its own independently tuned parameter set.

Automated Parameter Tuning

- Bayesian optimization (Optuna) models past trials and samples parameters where gains are likely
- Run-time compilation specializes the kernels for each proposal and GPU
- Independent steps tuned together, one compile and benchmark per iteration



Automated Parameter Tuning: Results



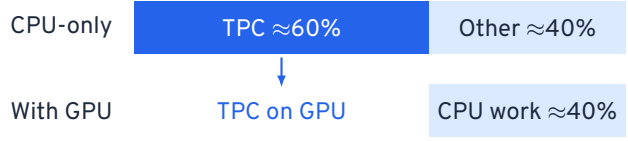
ALI-PERF-630310

- Time per time frame drops by 7% on the production MI50, 7–17% across ten GPU models
- 7% of the online farm is about 200 GPUs, or 25 of the 350 servers
- Enables automatic re-tuning: new GPU architectures, new parameters

Offline Reconstruction on GPUs

Same GPU code also runs on GPU-equipped Grid sites.

CPU workload — TPC-only offload baseline

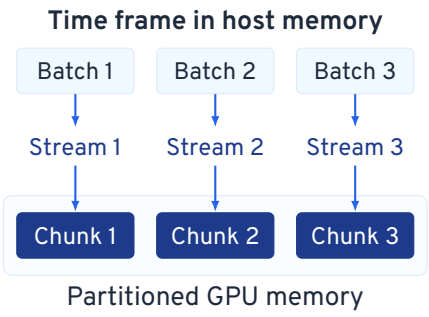


Throughput increase

2.5× observed

ITS Tracking on GPUs

- **Challenge:** building track candidates is combinatorial and memory-intensive
- **GPU strategy:** split the time frame into batches of readout frames, each assigned a GPU memory chunk
- **Execution:** process batches using multiple GPU streams, overlapping transfers with computation



ITS tracking

$\sim 3\times$ faster

MI50 vs. 16 CPU cores, high-rate Pb–Pb

Full offline reconstruction

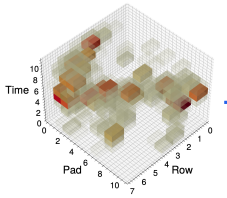
26% faster

ITS on GPUs in production since end of 2025

M. Concas, CHEP 2024; F. Schlepper, CHEP 2026

Neural Network Cluster Finding

Replace hand-tuned cluster creation using charges from neighbouring pad rows.



Input: local 3D charges
Row × pad × time

Classifier NN → **Selection score** 0–1

Regressor NN → **Center-of-gravity, widths, total charge**

Separate fully connected models

Classification

Keep candidates above a configurable score threshold.

Regression

Estimate CoG, widths and total charge for tracking and particle identification.

Neural Networks in the Online GPU Chain

The NN replaces classical cluster creation.

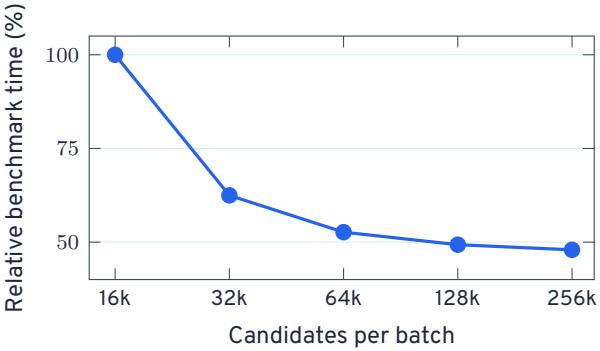
Inside the GPU process

- Charge windows assembled in GPU memory
- ONNX Runtime binds the existing device buffers
- Inference uses the reconstruction stream for each processing lane

Existing reconstruction interface

- Creation of cluster candidates
- Predictions become standard cluster objects
- Downstream tracking consumes the same cluster format

Inference Optimizations



AMD MI50; fully connected model, 5 layers × 32 neurons.
Normalised to batch size 16,384; each tick doubles the batch size.

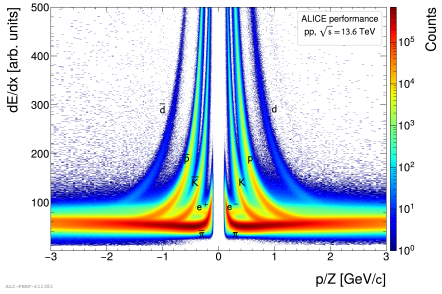
- Small fully connected ReLU models were fastest in the tested configurations
- Half precision reduces inference time
- Larger batches improve throughput, with diminishing gains

NN Cluster Finding: Validation and Commissioning

Online commissioning — pp

Two 30-minute runs

pp collisions at 500 kHz



TPC d*E*/d*x* from the first NN commissioning run.

Separate validation — Pb–Pb

Up to 18%

fewer clusters

No loss in physics performance

processing-time cost

1.25–1.4×

vs. the classical pipeline on AMD MI50

Machine Learning across ALICE

Data quality monitoring	Autoencoder flags anomalies in TPC data; runs alongside the QC framework. Z. Sourpi
Fast simulation	End-to-end fast simulation of the Zero Degree Calorimeter. D. Fuligno
	WGAN simulation of slow-neutron induced TPC loopers. M. Giacalone Covered by plenary talk on Tuesday
Particle identification	Transformer and XGBoost classifiers on track features, tolerant to missing detectors. L. Graczykowski & L. Sirko
	Neural networks calibrate the TPC energy-loss response for particle identification. C. Sonnabend

Conclusion

- **Online reconstruction**
  Real-time reconstruction and compression of up to 1.2 TB/s on 2,800 GPUs.
- **Shared online and offline software**
  Portable CPU/GPU code; the online farm also runs offline reconstruction between data-taking periods.
- **Developments since the start of Run 3**
  Automated tuning reduces processing time by 7% on MI50; GPU ITS tracking makes offline reconstruction 26% faster.
- **Neural-network cluster finding**
  Demonstrated during online pp commissioning; separate Pb–Pb validation shows up to 18% fewer clusters, with no observed loss of tested reference-particle yields.
- **Outlook**
  Extend offline GPU coverage and reduce the additional processing cost of NN cluster finding.

References

- G. Eulisse, D. Rohr, *The O^2 software framework and GPU usage in ALICE online and offline reconstruction in Run 3*, CHEP 2023, arXiv:2402.01205
- D. Rohr, *Improvements of the GPU processing framework for ALICE*, CHEP 2024, EPJ Web Conf. 337, 01362 (2025), arXiv:2511.17018
- D. Rohr, *Usage of GPUs for ALICE Run 3 offline reconstruction on the GRID*, CHEP 2026, arXiv:2608.11819
- G. Cimador, *Automated tuning of GPU kernel parameters via run-time compilation for the ALICE online reconstruction*, CHEP 2026, indico.cern.ch/event/1471803/contributions/6966811
- M. Concas, *Extending ALICE's GPU tracking capabilities: towards a comprehensive accelerated barrel reconstruction*, CHEP 2024, EPJ Web Conf. 337, 01019 (2025), epj-conferences.org
- F. Schlepper, *3+1D GPU reconstruction of the Inner Tracking System 2 for ALICE Run 3*, CHEP 2026, indico.cern.ch/event/1471803/contributions/6967219
- C. Sonnabend, *Neural network cluster finding for the ALICE TPC online computing*, CHEP 2026, indico.cern.ch/event/1471803/contributions/6967025