ATLAS GPU pipelines for tracking: An overview

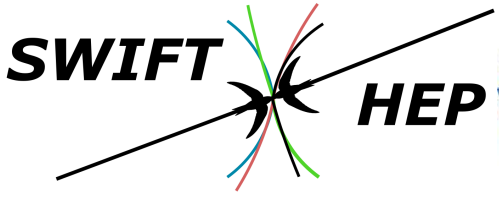
Jyoti Prakash Biswal

Rutherford Appleton Laboratory

GridPP53 & SWIFT-HEP09

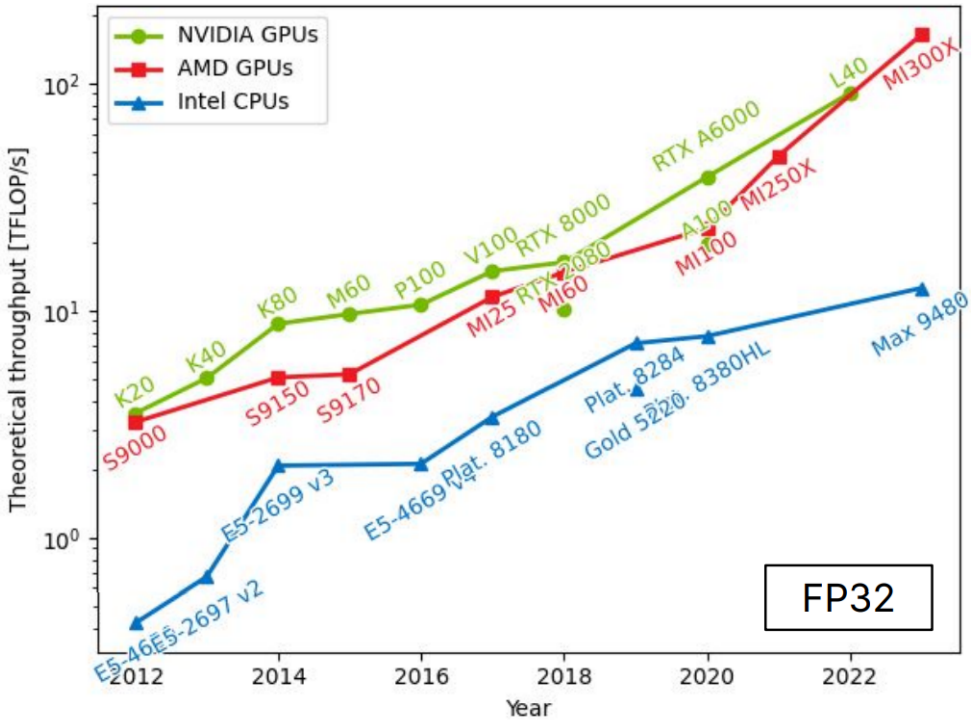
University of Manchester

11 April 2025



Motivation

- HEP experiments encounter significant challenges in data analysis because of the vast amount of data involved.
- GPUs have the potential to revolutionise data processing by significantly surpassing CPUs in handling parallelisable algorithms with vast datasets.
- Utilising GPUs for track reconstruction is ideal due to their computational intensity and potential for parallel processing.
- This idea is driven by ACTS through *traccc* library (see next slide).



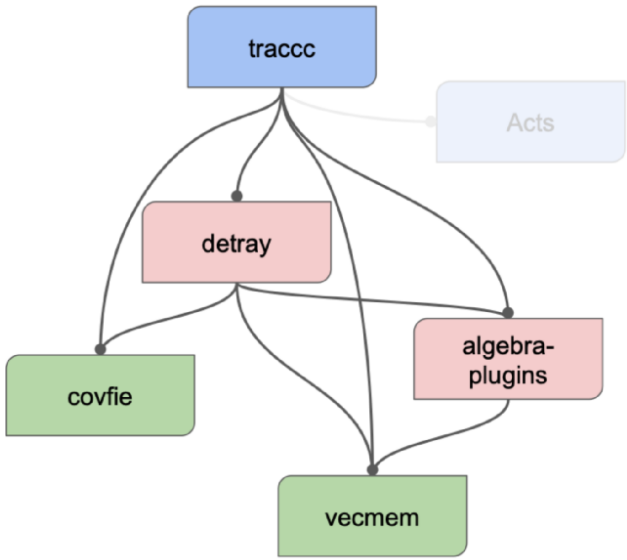
- FP32: Single-precision floating-point format (32 bits).
- FLOPS: Floating-point operations per second.
- TFLOPS: Tera ($= 10^{12}$) FLOPS.
- $$\text{Theoretical throughput} = (\text{number of cores}) \times \frac{\text{FLOPS}}{\text{cycle}} \times \frac{\text{cycles}}{\text{second}}$$

Recap of *traccc* project

- The ACTS (A Common Tracking Software) project aims to modernise tracking software for current systems with sustainable code maintenance.
  - A bridge to general detector libraries (DD4Hep and GeoModel) for *traccc*.
- *traccc*, developed independently but with ATLAS involvement, is a GPU tracking implementation evolving from ACTS with re-integration plans.
  - It comprises multiple projects, each targeting specific tasks.

- detray: GPU-optimised detector description library [Silicon-based detectors; gaseous detectors with wire measurements].
- covfie: library for representing vector fields.
- algebra-plugins: small matrix linear algebra abstractions.
- vecmem: memory management library for heterogeneous computing.

traccc: main repository, primarily algorithmic code.



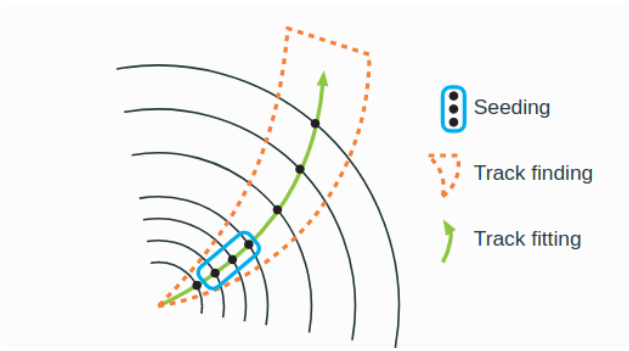
The goal is to maintain GPU track reconstruction with full physics quality.

traccc, similar to SWIFT-HEP, emphasizes GPU development.

EF tracking

- The high pile-up conditions planned for the High Luminosity LHC (HL-LHC) give rise to specific challenges for the **trigger** algorithms.
- Event Filter (EF) tracking will run on a farm of commodity CPUs with/without GPUs/FPGAs.
  - Technology choice later this year!
- EF track reconstruction chain:
  1. An initial data preparation stage – Inner Tracker (ITk) data preparation.
  2. Seed formation and pattern recognition step (track Seeding and pattern recognition).
  3. Track extension, fitting and ambiguity resolution.
- Robust against pile-up: capable of reconstructing vertices and tracks, ensuring high tracking efficiency like offline.
- **Athena** is the ATLAS software framework that manages almost all ATLAS production workflows: event generation, simulation, reconstruction and derivation production.
- Full track reconstruction chain in Athena:

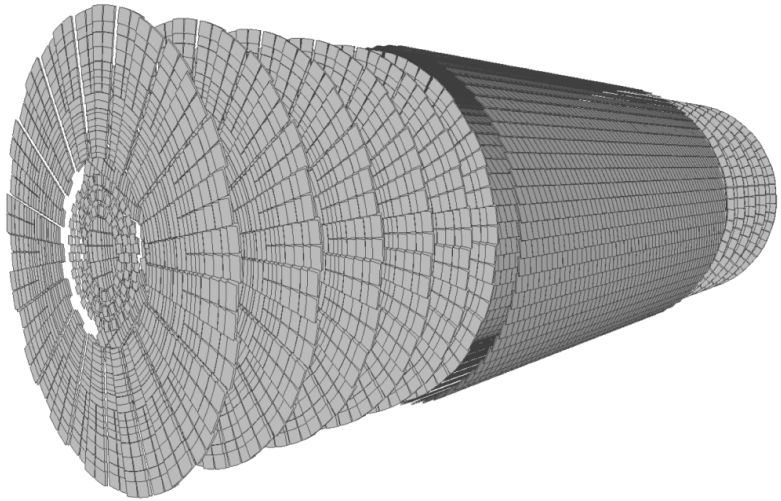
Clusters → Seeds → Track finding →
Track fitting → Ambiguity resolution



Track reconstruction with GPUs (I)

- Running track reconstruction on GPUs demands a special description of **the detector**, **the material in the detector**, **magnetic field**, and **module design**.

  - The Inner Tracker (ITk) geometry is translated from GeoModel (a detector description toolkit for HEP experiments) to detray using ACTS.
  - The process needs updates for production readiness with ACTS dependencies and version in Athena externals.
  - Future translations will be done instantly without saving or loading detray JSON files.



ITk from detray viewpoint.

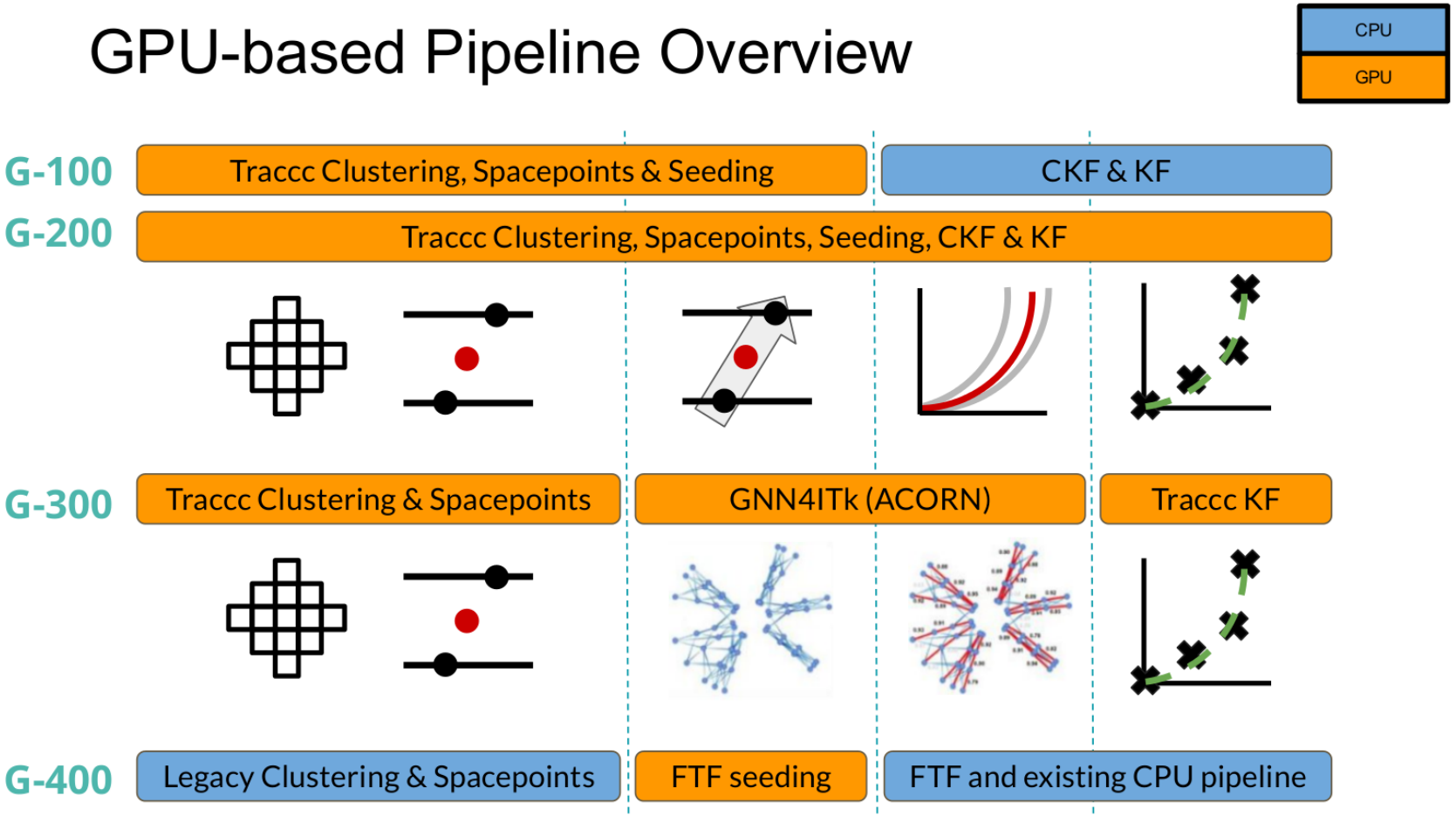
- Work on creating the material maps is well advanced – an agreement between Geant4, ACTS and detray can be demonstrated.
- At the moment, a constant magnetic field approximation is used – magnetic field description is ready to be imported to covfie (work-in-progress).
  - Only $\sim$ a 10% increase in processing time with a non-constant magnetic field description is expected (link).

Track reconstruction with GPUs (II)

- *traccc* currently reads **module design** from the JSON file using the ACTS ID of the surface.
  - It does not yet handle complex module shapes.
  - It also does not yet handle clustering on modules with non-uniform pitch.
  - Strip clustering is treated as 1-dimensional pixel clustering.

- The Event Data Model (EDM) improves commonality across the detector subsystems and subgroups.
  - It includes a common track class for combined reconstruction in inner tracking and muon detectors.
  - GPUs require the following EDM conversions:
    - Hit information → *traccc* cell container.
    - InDet clusters → *traccc* measurements and spacepoints.
    - *traccc* tracks → ACTS tracks.
    - *traccc* clusters → xAOD (= ROOT-readable format) clusters.

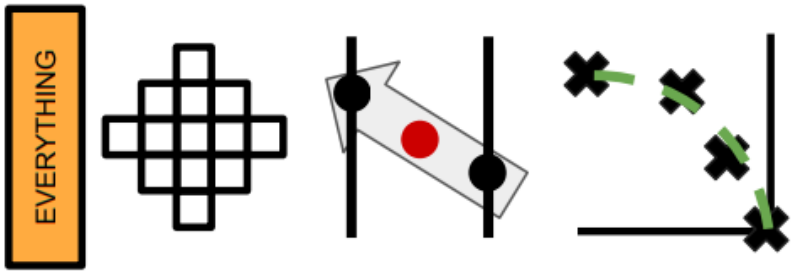
GPU-based Pipeline Overview

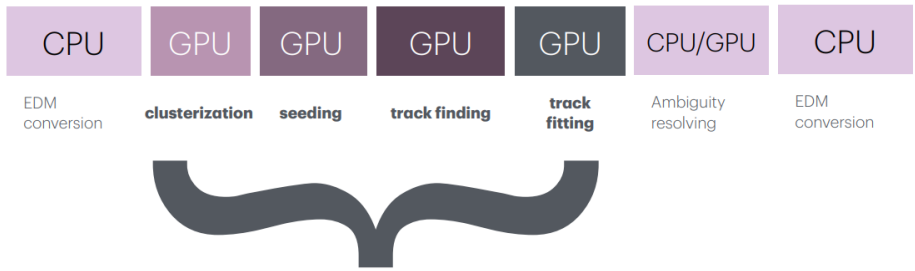


- **G-200**: full end-to-end GPU pipeline using *traccc*.
- **G-100**: the same approach as **G-200**, but GPU returns seeds for subsequent processing on CPU.
- **G-300**: replaces *traccc* components with the Graph Neural Network (GNN).
- **G-400**: uses legacy Athena framework with new GPU algorithms.

G-200 (I)

- Full *traccc* GPU pipeline – every component of the track reconstruction chain executed on the GPU utilises *traccc*.





- It has showcased *traccc* using CUDA and SYCL on both Nvidia and AMD GPUs.

- G-200 can be run as part of Athena as well as part of the trigger chain.

- It has also been successfully linked to IDTPM, and conducts Continuous Integration (CI) and ART tests.
  - IDTPM: **InDetTrackPerfMon** generates user-defined tracking validation histograms for offline and Inner Detector trigger tracking.
  - ART: The **ATLAS Release Tester** system tests software on the Grid post nightly builds.

G-200 (II)

Physics performance studies

- The studies were performed with (a) the single muon samples with various $\mathrm{p_T}$ profiles [1 GeV, 10 GeV and 100 GeV] and (b) $t\bar{t}$ samples with/without pile-up 200.
- Caveats: (1) constant magnetic field approximation, and (2) no material map.

Status of computing performance studies

- The numbers are obtained by running on the EF tracking GPU testbed, lxplus-gpu, RAL GPU farm, and others.
- Seeding and track finding are the most "expensive".
- Caveats: (a) *traccc* has no track quality cuts optimised for ITk and (b) massive overtime (due to inclusive reconstruction).

Plans

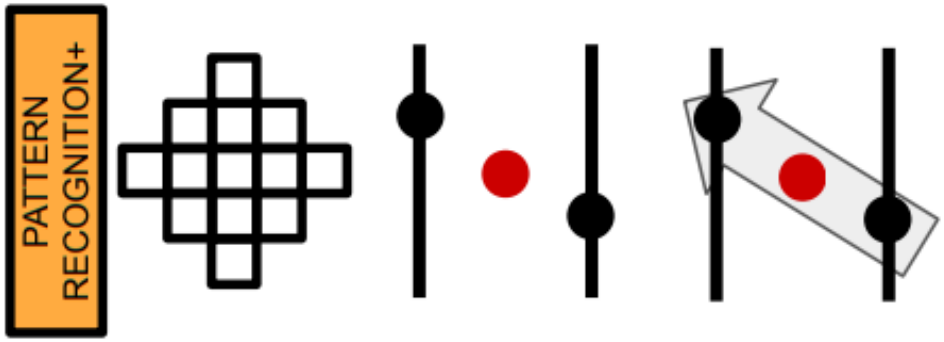
- Inclusion of the material map and non-constant magnetic field description.
- Understanding of the duplication of measurements in strip modules for clustering.
- Implementation of a different description for modules (with annulus bounds).
- Performance optimisation – seeding configuration for ITk, triplet fit algorithm on GPU, ...

RAL involvement [in progress]:

- Comparison of the output from G-200 with C-000 using IDTPM. || Debugging of the G-200 pipeline.

G-100 (I)

Data preparation pipeline.



Algorithms

- **Input:** Hits.

1. *traccc* pixel clusterisation.
2. *traccc* pixel spacepoint creation.
3. *traccc* pixel seeding.

- This pipeline runs as a "proof of principle" with GPU seed-finding only.
- It follows G-200 development.
- It is expanding *traccc* GPU algorithm usage back down the pipeline towards the initial hits.
- **Comparison with different inputs (clusters):**
  - Shows robustness of *traccc* → xAOD converter code.
  - Comparable results with G-200 (the ones that start from clusters).

G-100 (II)

Performance status

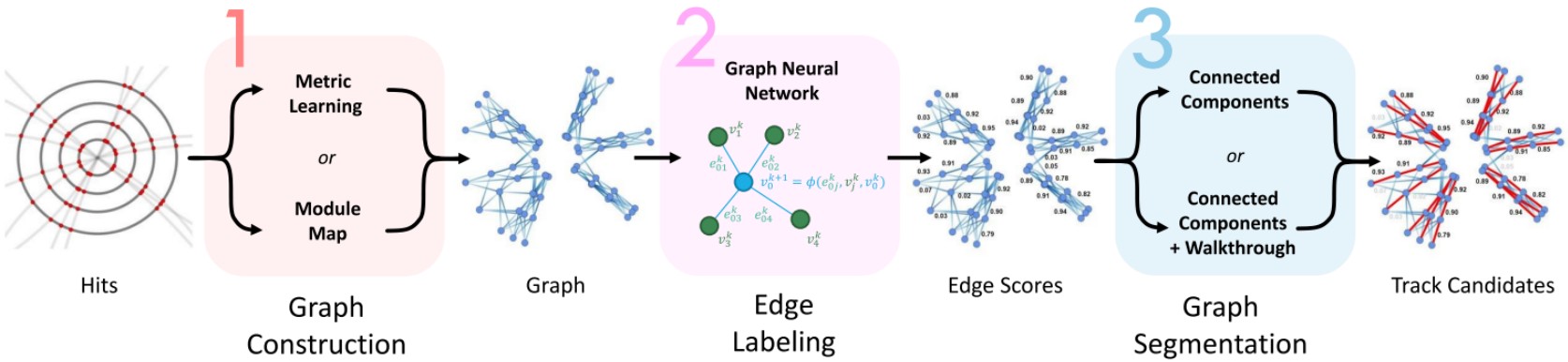
- Poor efficiency at barrel region – single muon samples.
- Areas under investigation:
  - *traccc* to Athena geometry mismatch (major).
  - Lack of strip hits.
  - Inclusion of pixel analogue calibration.
- There is also detray to ITk granularity difference.
  - The pixel pitch is not uniform in some layers of the endcap and barrel.
- Seeding efficiency ($\Rightarrow$ expected track efficiency coverage) was analysed with $t\bar{t}$+ pile-up 200 sample.
  - Without the inclusion of strip hits.

Work-in-progress

- Optimisation of *traccc* GPU seed-finding.
- Integration of *traccc* GPU seed-finding with Athena workflow.
- Enhancement of EF tracking performance.

G-300 (I)

GNN on GPU pipeline. || Goal: Full GPU reconstruction pipeline based on GNNs and *traccc*.



Implementation in ACTS standalone (✓)

- Allows flexible testing with third-party libraries.
- Fast turnaround due to more lightweight software – comes with a price: ITk geometry outside Athena is non-trivial!
- Clusters and spacepoints imported from Athena – dumped as ROOT files.

Throughput measurements ($t\bar{t}$+ pile-up 200)

- A buffered data reader has been implemented in ACTS to work around the I/O bottleneck.
- Use TensorRT backend (most performant, some memory leak in libtorch).
- Timing mode (no truth information, metric computations).
- Only GNN part/pattern recognition is included.

G-300 (II)

- Optimisation of module map CUDA kernels helped achieve a speed-up of $\sim 3\times$.
- **Static quantisation**: the precision of parts of the network after the training is changeable, *e.g.*, 32-bit float $\rightarrow$ 16-bit float.
- Baseline hardware for GNN4ITk results: Nvidia A100.
- Has recently been tested on RTX 5000 Ada Generation (testbed).

  $\Rightarrow$ Throughput: substantial differences between testbed and A100, $\sim 2\times$.
- Single-threaded timings were also obtained for (1) graph construction, (2) GNN, and (3) track building.
  - Hardware: A100 – float32 & quantised; testbed – float32 & quantised.

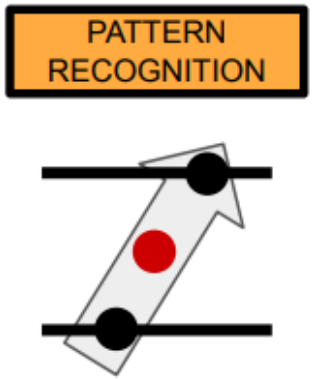
Athena integration (ongoing)

- Rely on CPU clustering and spacepoint formation.
- Rely on CPU KalmanFilter (optionally *traccc*, but not a priority).
- Expose one function in ACTS that captures the workflow.
- For now, it will require a custom build of ACTS (due to third-party dependencies).

G-400 (I)

FastTrackFinder (FTF) GPU seeding pipeline. || FTF is an HLT algorithm for reconstructing tracks in the detector (steps below).

- Search for groups of spacepoints (track seeds) in regions of interest (RoIs) from L1.
- Combinatorial Kalman filter extends seeds in search roads and builds track candidates.
- Another step is to remove duplicate tracks sharing more than a certain number of hits.



- Refactoring of the GPU seeding code (✓).
- Interface GPU seed-finding code with EFTrack demonstrator (✓).

RAL involvement: Implementation of Graph-Based Track Seeding (GBTS) on GPU [**ongoing**].

What is GBTS?

- Pair up space points in linked η-bins, form edges, and create a graph of connected edges with similar track parameters.
- Use Kalman filter-based walkthrough to extract seeds from graph, starting from the **best edges** identified by Connected Component Analysis (CCA).
- **A key goal** is to accelerate track following by providing pure seeds (75-95%).

G-400 (II)

- GBTS **part-1** offloads the graph node preparation and graph building, but the rest is on the CPU (✓).
  - Node preparation is parallelised over ∼0.2 million space points.
  - Graph building: Edge linking is parallelised by η-bin → one sub-bin bin/pair per thread block.
  - **Preliminary timings** (on A100 with $t\bar{t}$+ pile-up 200):
    - Graph building accelerated by **28**×.
    - A factor of **4.6**× for overall seeding, mainly because of data export needed for CPU-based parts of the GBTS algorithm.
    - The number of seeds is 10-15% more than the CPU version, resulting in ∼10% more tracks.
  - It is already merged into Athena.

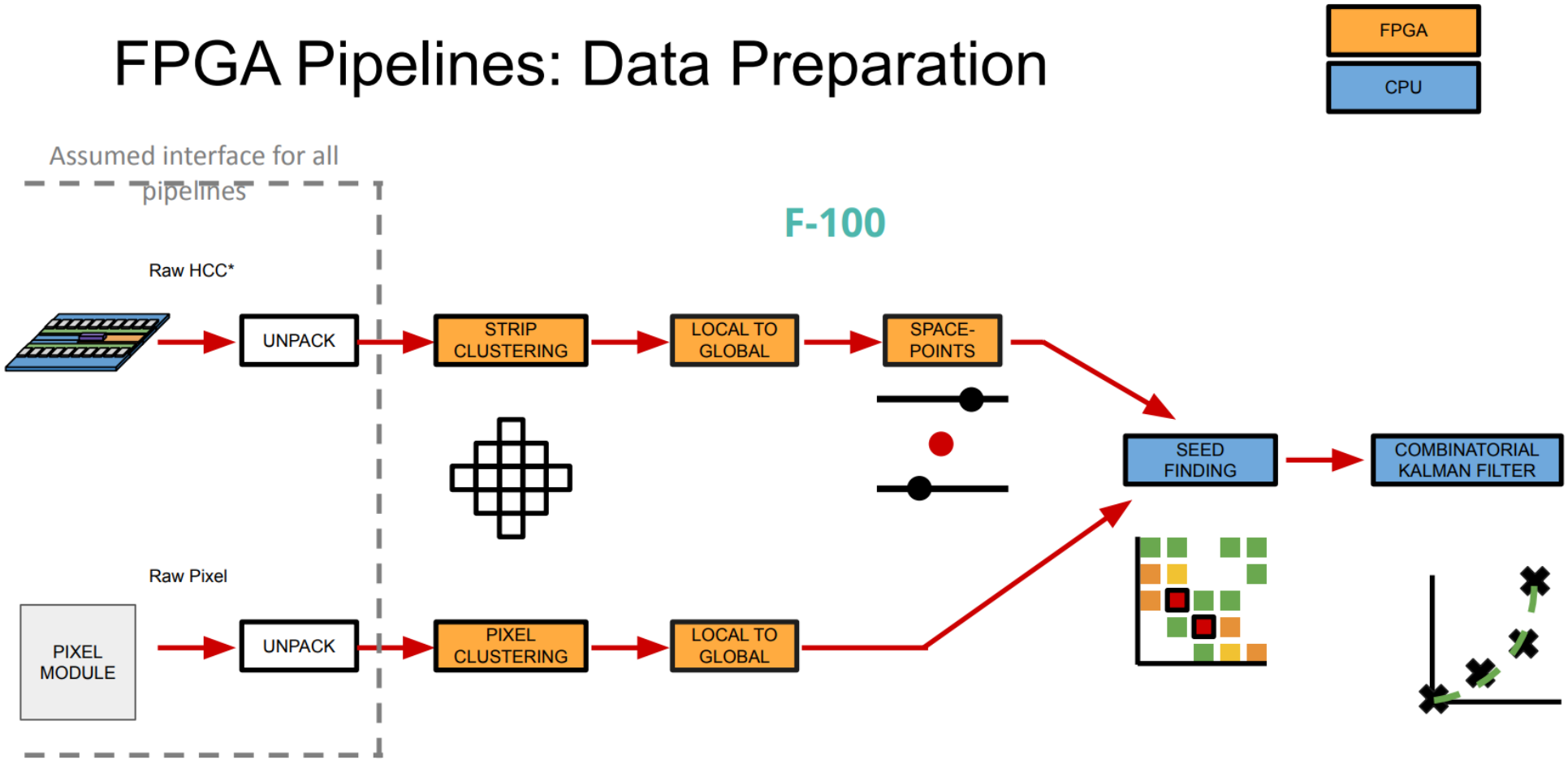
- GBTS **part-2**: GPU seed extraction [**work in progress**].
  - Athena integration after more tests.
  - Benchmarking on a broader range of GPUs.

Outlook

- Individual pipelines have made significant progress in the past six months or so.
  - Updates are regularly presented at the fortnightly GPU roundtable and EF tracking weekly meetings.
- Mid-cycle reviews are in progress for G-200, G-100 and G-300 – by the end of April 2025.
  - The optimisation cycle reviews are due by the end of July 2025.
- G-400 has impressive preliminary results, and its development is advanced.
- ATLAS is developing GPU pipelines to be compared against CPU-only and FPGA-accelerated alternatives to inform the technology choice.
- RAL is actively participating in the effort.

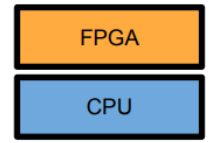
Backup

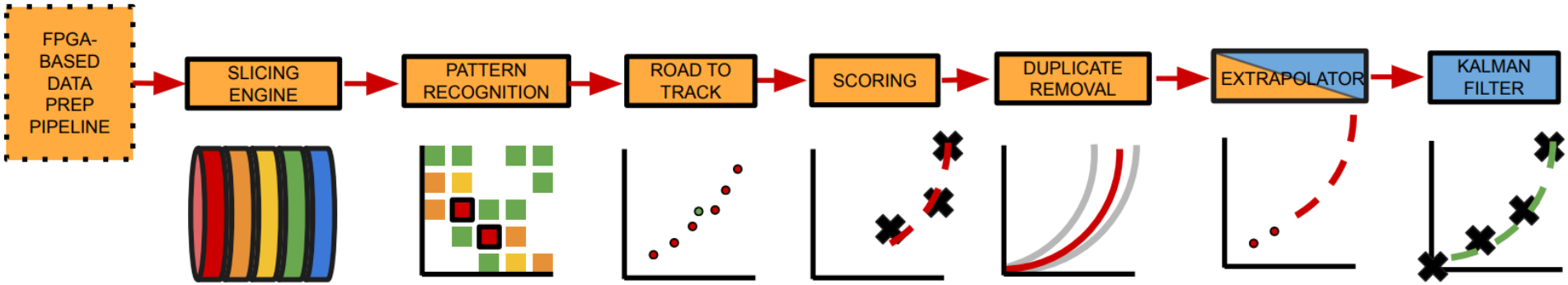
FPGA Pipelines: Data Preparation



FPGA Pipelines: Tracking

F-2xx through F-5xx





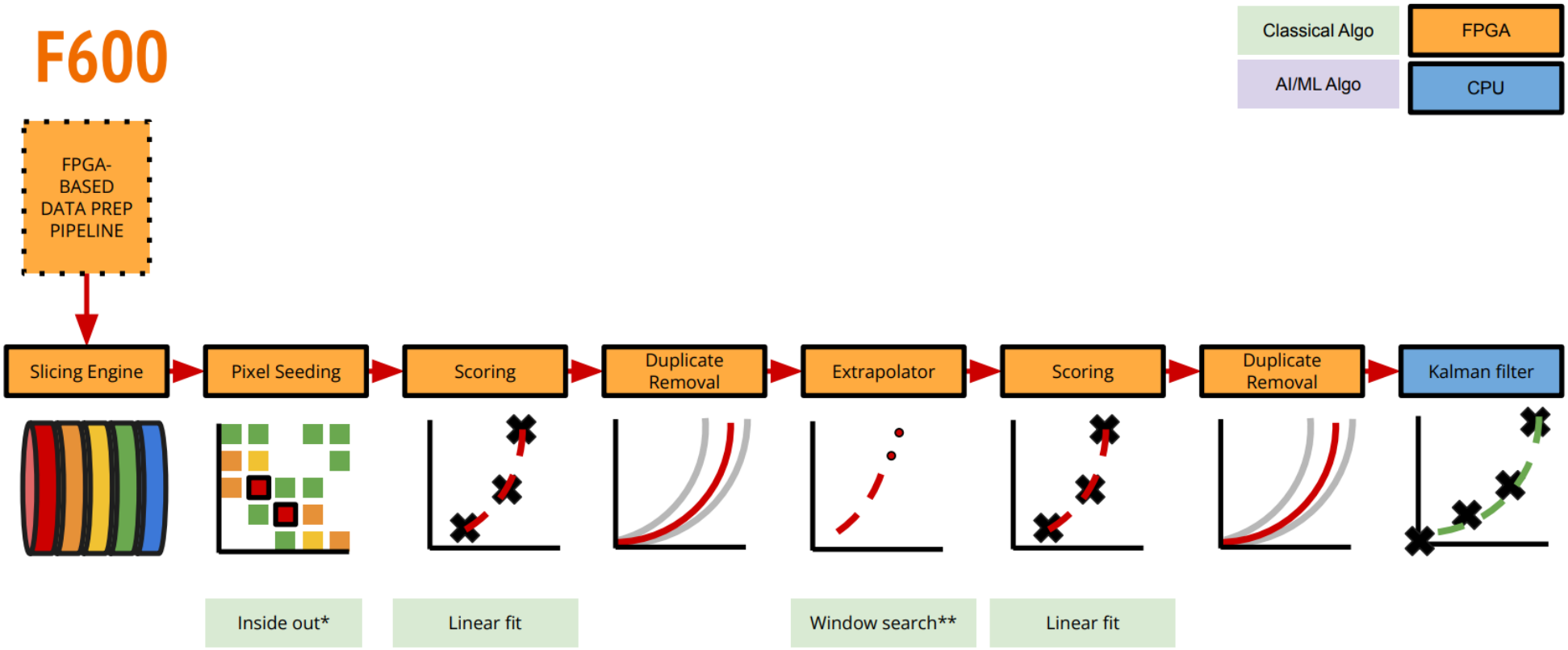
Options:

F-2xx 1-D Bit Shift
F-3xx 2-D Hough Transform
F-4xx GNN
F-5xx 1-D Bit Shift + NN Pathfinder

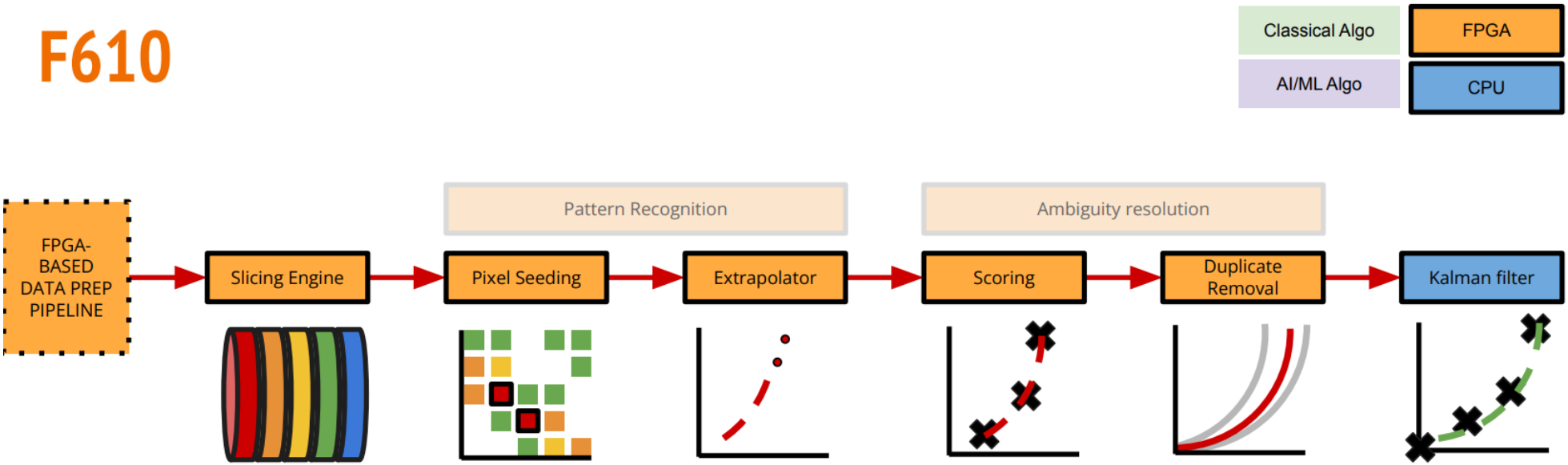
Options:

F-x0x Linear Fit
F-x1x NN

F600


Classical Algo
FPGA
AI/ML Algo
CPU
FPGA-BASED DATA PREP PIPELINE
Slicing Engine
Pixel Seeding
Scoring
Duplicate Removal
Extrapolator
Scoring
Duplicate Removal
Kalman filter
Inside out*
Linear fit
Window search**
Linear fit

F610



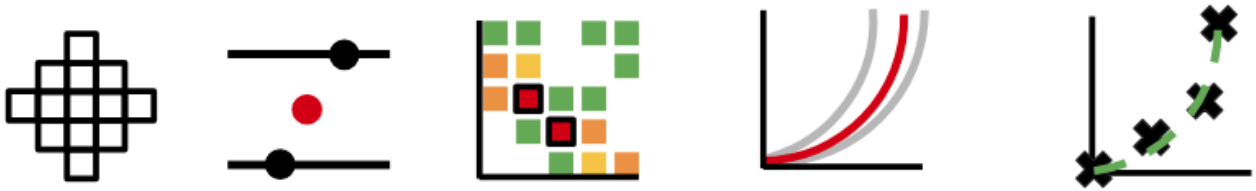
Inside out* | NN Pathfinder | NN Scoring

* it is unclear if we need a scoring/duplicate removal step after seeding. From Barrel results, it seems we don't need one

CPU-only Pipelines

CPU

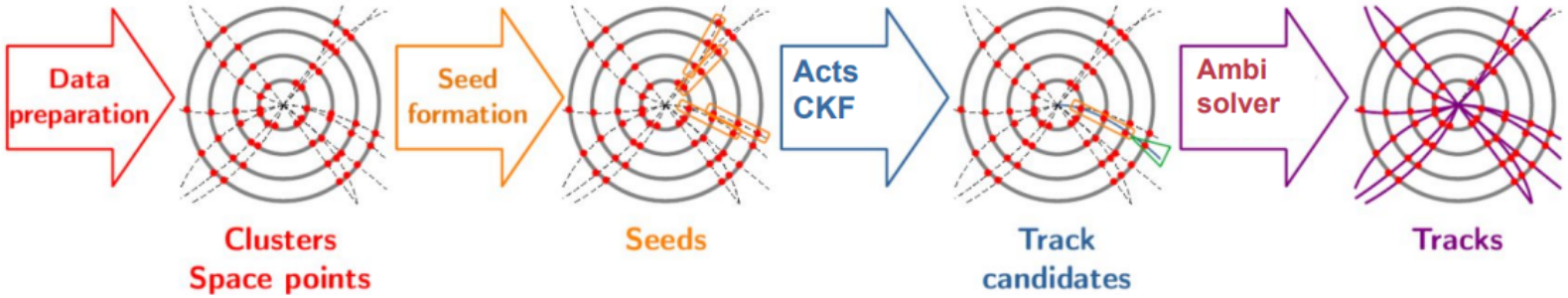
C-000 Offline Athena-ACTS full pipeline (reference)



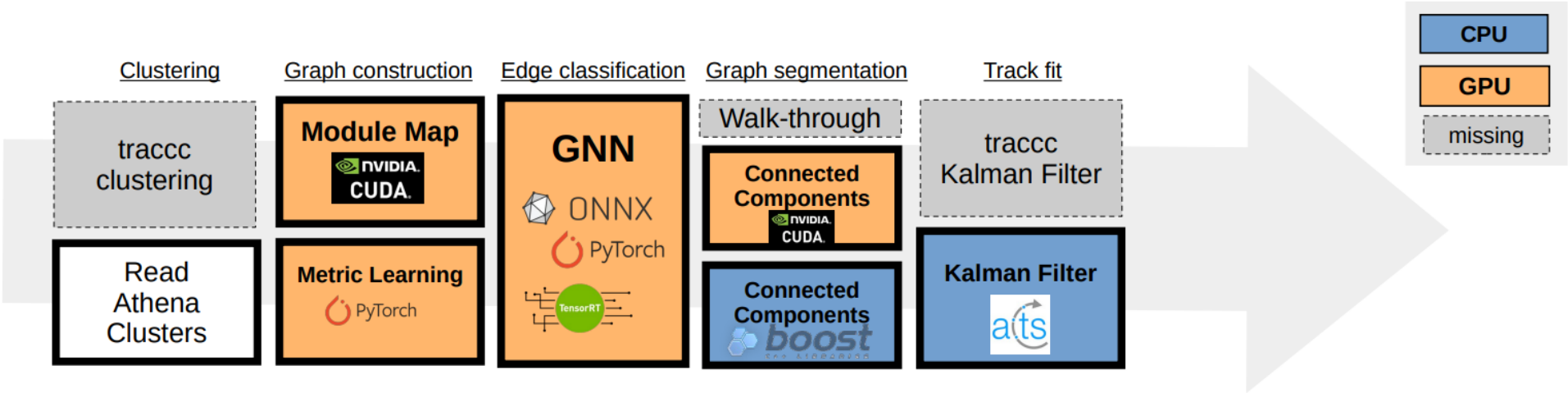
C-100 (Optimized) Offline Athena-ACTS without AR (“Fast Tracking”)

C-110 FastTrackFinder (FTF) using ITk and integrated into Athena/ACTS

C100 / C230 Pipeline overview



CPU - Based Pipeline

C100	ACTS-based Fast Tracking (FT) CPU Chain		
C230	ACTS Clustering / SPs	Graph-based (FTF) Seeding (ACTS)	ACTS CKF

G-300 overview