Studies of FPGA accelerated track reconstruction for the ATLAS Event Filter

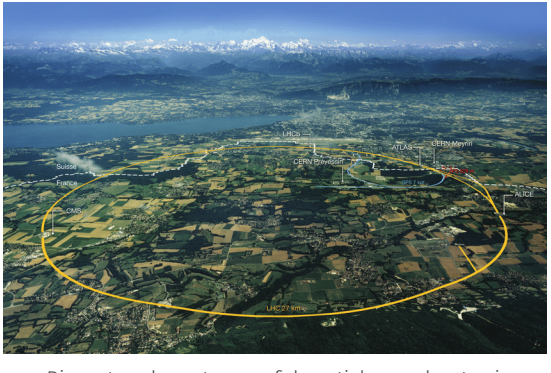
Kevin Sedlaczek on behalf of the ATLAS collaboration

IEEE Realtime 2026 | May 28, 2026

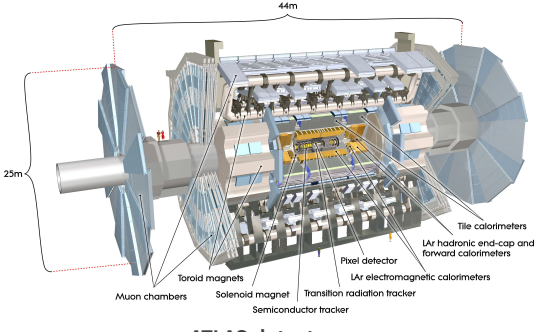
kevin.sedlaczek@cern.ch

Northern Illinois University

The ATLAS Detector at the Large Hadron Collider



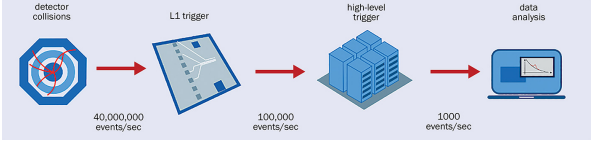
- Biggest and most powerful particle accelerator in the world (27km ring)
- Collision of protons and heavy ions



ATLAS detector:

- one of four large experiments at the LHC
- multiple systems to measure the products of proton collisions

Real-Time Filtering of the Collision Data

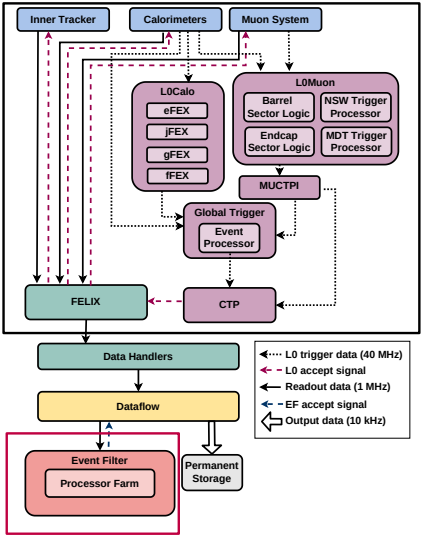


- Collisions at 40 MHz (every 25 ns) with event sizes at 1 MB: **data output of ≈ 320 TB/s**
- A **trigger system** rejects most events
- ATLAS is currently developing a **completely new trigger system** for the coming data taking periods

Hardware trigger (L0): custom read-out boards using FPGAs with 1 MHz output

Software trigger (Event Filter): commodity servers with final accept decision at 10 kHz (ATLAS recently decided for GPU+CPU)

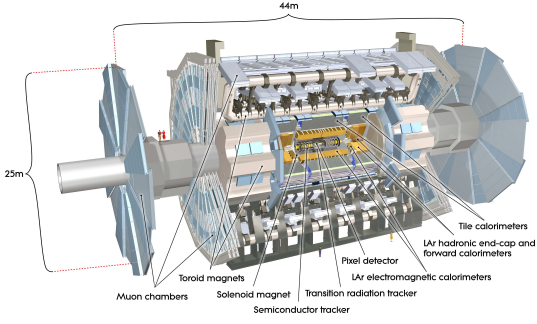
Tracking is the biggest part of the Event Filter



Technical Design Report. Addendum to the TDR

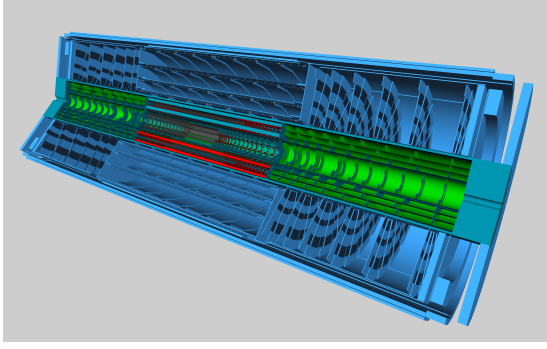
The Challenge: Reconstructing Particle Trajectories at very low Latency

- Reconstructing the particle trajectories is **essential** for the reconstruction of the physics event
- ATLAS will introduce a sophisticated **silicon-based detector (ITk)** to reconstruct particle trajectories as point clouds of spatial coordinates (hits)
- The upcoming data taking periods will be defined by **unprecedented collision rates**
- Aiming for up to **200 interactions per collision**, creating around **10M hits** in the tracking detector, originating from **~10.000 real particle tracks**
- The maximum latency for this is $\mathcal{O}$(ms)



The Challenge: Reconstructing Particle Trajectories at very low Latency

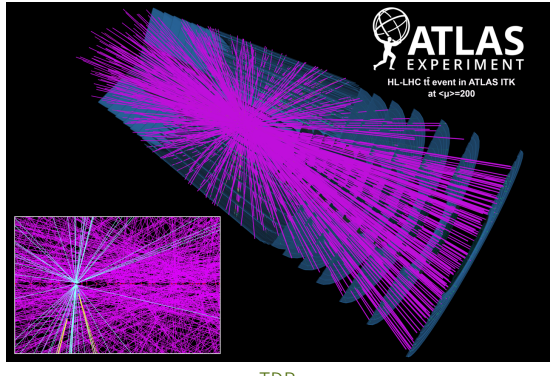
- Reconstructing the particle trajectories is **essential** for the reconstruction of the physics event
- ATLAS will introduce a sophisticated **silicon-based detector (ITk)** to reconstruct particle trajectories as point clouds of spatial coordinates (hits)
- The upcoming data taking periods will be defined by **unprecedented collision rates**
- Aiming for up to **200 interactions per collision**, creating around **10M hits** in the tracking detector, originating from **~10.000 real particle tracks**
- The maximum latency for this is $\mathcal{O}$(ms)



TDR

The Challenge: Reconstructing Particle Trajectories at very low Latency

- Reconstructing the particle trajectories is **essential** for the reconstruction of the physics event
- ATLAS will introduce a sophisticated **silicon-based detector (ITk)** to reconstruct particle trajectories as point clouds of spatial coordinates (hits)
- The upcoming data taking periods will be defined by **unprecedented collision rates**
- Aiming for up to **200 interactions per collision**, creating around **10M hits** in the tracking detector, originating from **~10.000 real particle tracks**
- The maximum latency for this is $\mathcal{O}$(ms)



TDR

How does Tracking work?

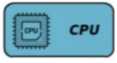
Track Reconstruction in the Event Filter

- **Energy deposits** in the silicon sensors are measured and used to reconstruct hits of **charged particle trajectories**
- Clusters are used to **create seed tracks**: combinations of 4-5 hits in the inner most detector region (closest to collision)
- Seed tracks are then **extended to the outward layers** of the detector
- **Duplicates** and likely **false track candidates** are removed
- The remaining tracks are run through a **high-quality track reconstruction algorithm**, which applies another quality selection and reconstructs the **track parameters** of interest

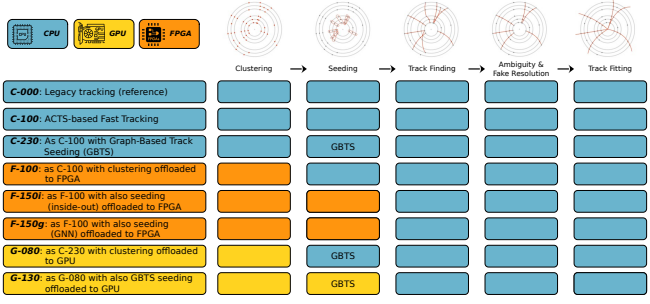
The different tracking steps can either be run on a CPU or offloaded to an accelerator card

Choosing the best Heterogeneous Computing Solution

- The ATLAS collaboration has spent the last 4 years building demonstrators for three technologies:



- For each technology multiple **demonstrator pipelines** for tracking in the Event Filter have been developed and tested
- The goal: **find the setup that performs best under the given power, latency and cost constraints**
- This is the **first direct comparison** of CPUs, GPUs and FPGAs for a trigger system at the LHC



Three FPGA-based tracking solutions in this talk: F-100, F-150i and F-150g

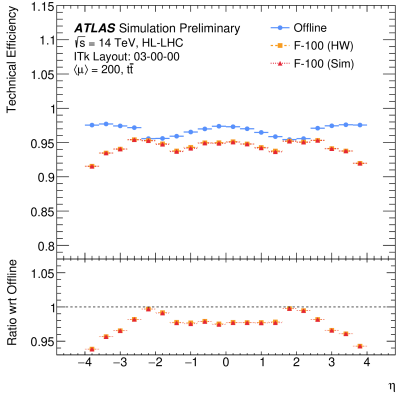
Simulating and testing the Behavior and Performance of the Tracking Algorithms

- **Algorithmic C++ simulation** of the FPGA tracking pipelines as part of the **public**, **version-controlled** ATLAS software framework `athena`, alongside firmware kernels using RTL/HLS
- Allows for fast development and testing of new algorithms using **accurate simulation of the physics and detector response**
- Combination of **software** (C++) and **firmware** (HLS/RTL) simulations:
  - Easy development, optimization and **fast testing and comparing of new algorithms** without the initial need for a full firmware implementation
  - **Pragmatic accuracy model**: not bitwise simulation, but relevant floating-point quantities are truncated to the target data-format precision, which yields **near perfect agreement**
  - Ultimately the performance is **tested on the hardware** using proper firmware kernels in representative testbed machines
- Workflow:
  - **Develop and optimize algorithms in simulation** and validate their performance
  - Test **power consumption** and **latency** on the hardware

Alveo U250 and U55c FPGAs

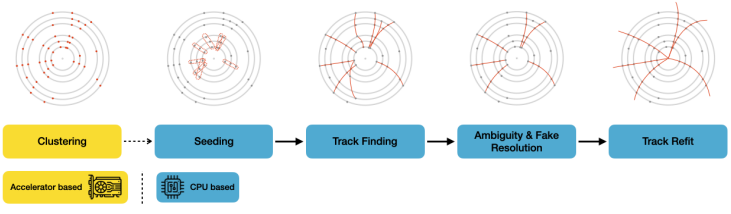
Simulating and testing the Behavior and Performance of the Tracking Algorithms

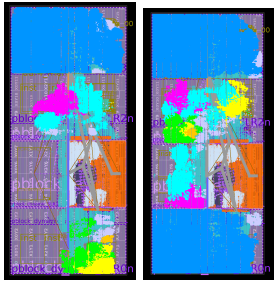
- **Algorithmic C++ simulation** of the FPGA tracking pipelines as part of the **public**, **version-controlled** ATLAS software framework `athena`, alongside firmware kernels using RTL/HLS
- Allows for fast development and testing of new algorithms using **accurate simulation of the physics and detector response**
- Combination of **software** (C++) and **firmware** (HLS/RTL) simulations:
  - Easy development, optimization and **fast testing and comparing of new algorithms** without the initial need for a full firmware implementation
  - **Pragmatic accuracy model**: not bitwise simulation, but relevant floating-point quantities are truncated to the target data-format precision, which yields **near perfect agreement**
  - Ultimately the performance is **tested on the hardware** using proper firmware kernels in representative testbed machines
- Workflow:
  - **Develop and optimize algorithms in simulation** and validate their performance
  - Test **power consumption** and **latency** on the hardware



ATL-DAQ-PUB-2025-002

F-100: A minimal, fully validated FPGA-based tracking pipeline





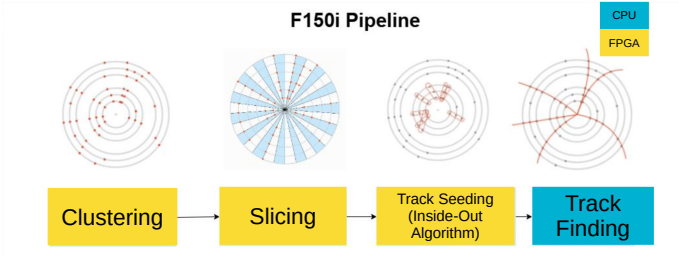
- The **hit clustering** and **coordinate transforms** are off-loaded to the FPGA: connected-component analysis of the ITk bytestream data on the accelerator card
- This offloads ≈**15% of the load to the FPGA**; seeding, CKF and ambiguity resolution remain on CPU

First successful implementation of FPGA based tracking in the EF trigger system

- off-loading
- validation of simulation accuracy (firmware/software)
- Integration of communication loop between firmware and ATLAS software (`athena`)
- performance and latency (within budget and competitive with other technologies)

→ **Laying the foundations for all other FPGA tracking pipelines.**

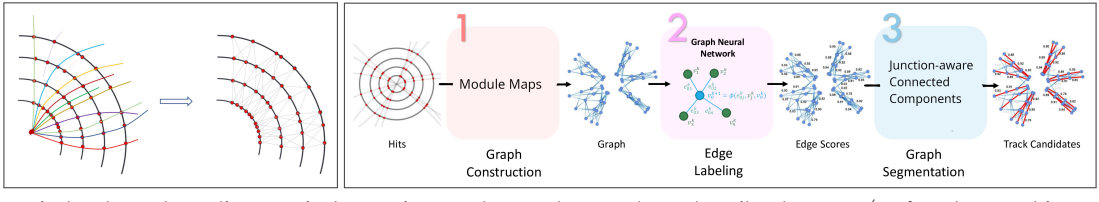
F-150i: Building upon F-100 with a novel track seeding algorithm



Checkout P. Sundararajan's poster from Tuesday for more details on this!

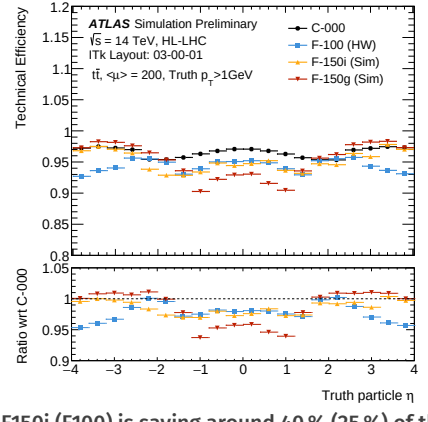
- Extension to the F-100 pipeline: **additional off-loading of pixel track seeding** to the FPGA (natural potential for parallelisation)
- Off-loading ≈**50 % of the load to the FPGA card**; CKF and ambiguity resolution on CPU
- **Specific algorithm** developed for this approach: the **Pixel-only Inside-Out track seeding**
  1. the **detector is split into 1280 regions**, in each of which seed tracks are formed using the inner-most layers of the detector
  2. clusters are classified based on a **binning in relevant track parameters** to find likely combinations
  3. track **candidates are constructed iteratively** starting from two clusters by adding consistent hits to form track candidates
  4. analytic expression is used to **estimate the quality** of the tracks and **bad quality tracks are removed**
- outputs track seeds with 4-5 hits, which are then extended using the CPU tracking algorithms

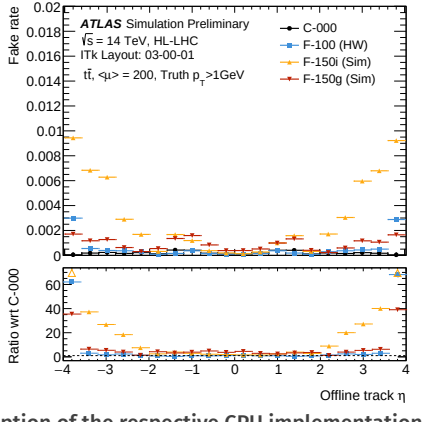
F-150g: Extending F-100 using graph neural networks for track seeding



- Pixel-only track seeding step is done using Graph Neural Networks to describe the event (**nodes**: detector hits, **edges**: connections between hits)
- Initially developed as part of the **GNN4ITk project**
- **First successful implementation of GNNs on FPGAs in EF tracking** (Alveo U250): significant work to **reduce the hardware footprint** of the models
  - No fully connected graphs: numbers of edges in reconstruction significantly reduced by using module maps, describing the detector geometry
- Reconstructed edges are assigned probability scores to **select high-quality reconstructed tracks**
- Create tracks from high-scoring edges, if multiple partially overlapping candidates: junction aware connected components analysis

Physics Performance, Power consumption

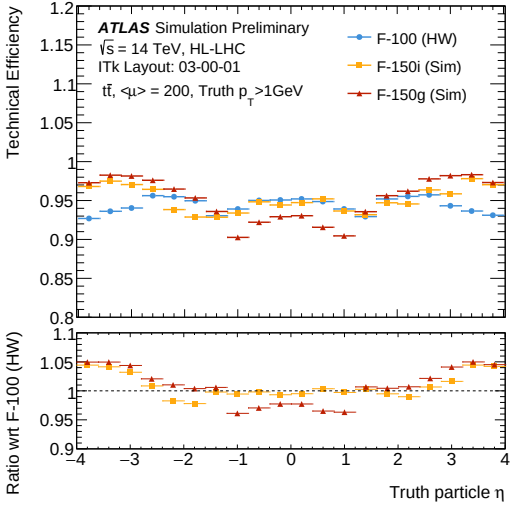


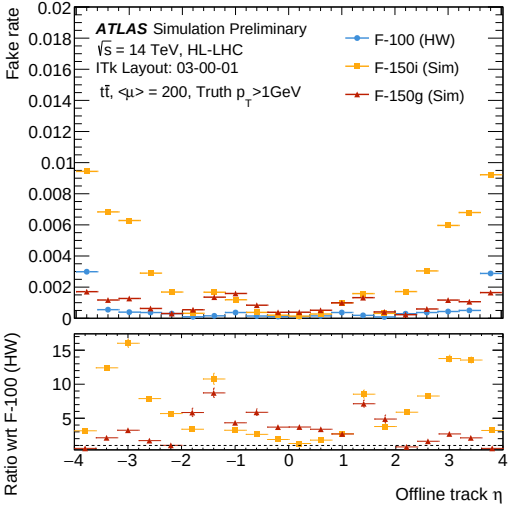


F150i (F100) is saving around 40 % (25 %) of the power consumption of the respective CPU implementation

More details on the performance comparison across technologies in M. Aparo's poster

Physics Performance, Power consumption





Conclusions

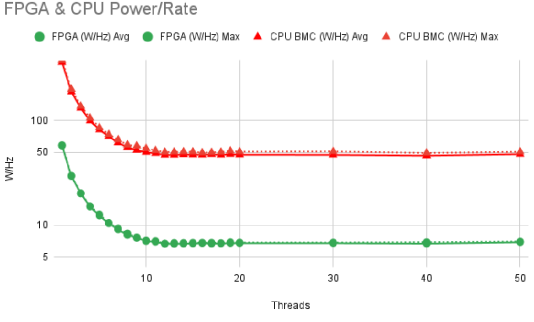
- The ATLAS collaboration has just concluded a **multi-year process** of testing **heterogeneous computing solutions** for the Event Filter trigger system
- This process has shown that off-loading various parts of the tracking chain to FPGAs is:
  - **Satisfying performance requirements**
  - **Working within the latency budget of the Phase-II trigger**
  - **An effective way to reduce the power footprint of the server farm**
- This is the **first time** a large LHC experiment has undergone a direct **quantitative comparison of CPU, GPU and FPGA** based heterogeneous computing solutions

FPGA Resource consumption

Computing farm constraints

- Power: 1.8 MW
- Rack space: 101 units

- FPGA power consumption when running the F-100 pipeline: ≈ 40 W
- Full server power consumption around 280 W



FPGA occupancy

Clustering (pixels), Local to global (pixels), EDM Prep (pixels), Clustering (strips), Local to global (strips), EDM Prep (pixels)

Name	LUT	REG	BRAM	URAM	DSP
Total FPGA resources	1,728k	3,456k	2,688	1280	12,288
Used by Static Platform	162k	257k	287	0	13
% of FPGA	9.39 %	7.45 %	10.68 %	0 %	0.11 %
Available User Budget	1,564k	3,195k	2,401	1,280	12,275
% of FPGA	91.61 %	92.55 %	89.32 %	100 %	99.89 %
Used Resources	**309k**	**310k**	**532**	**171**	**742**
% of User Budget	**19.77 %**	**9.71 %**	**22.16 %**	**13.36 %**	**6.04 %**
Clustering - Pixel	212	198k	480	96	448
% of User Budget	13.53 %	6.18 %	19.99 %	7.50 %	3.65 %
Clustering - Strip	2.5k	2.2k	0	0	0
% of User Budget	0.16 %	0.07 %	0 %	0 %	0 %
L2G - Pixel	15k	17k	0	27	95
% of User Budget	0.99 %	0.53 %	0 %	2.11 %	0.77 %
L2G - Strip	19k	19k	13	48	85
% of User Budget	1.20 %	0.60 %	0.54 %	3.75 %	0.69 %
EDM Prep - Pixel	33k	36k	8	0	66
% of User Budget	2.11 %	1.12 %	0.33 %	0 %	0.54 %
EDM Prep - Strip	19k	24k	8	0	48
% of User Budget	1.18 %	0.74 %	0.33 %	0 %	0.39 %
Helper Kernels	9k	15k	23	0	0
% of User Budget	0.59 %	0.47 %	0.96 %	0 %	0 %