Event Filter Tracking in ATLAS for the HL-LHC

Ben Rosser

University of Chicago

May 15, 2024

THE UNIVERSITY OF CHICAGO

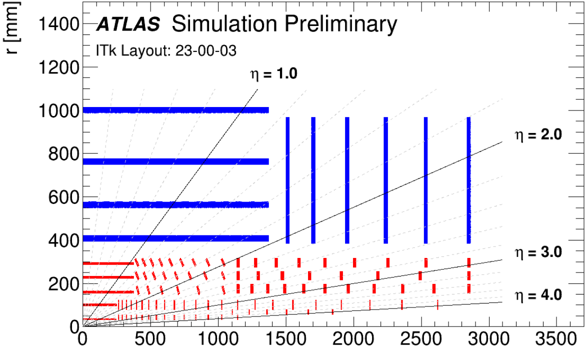
ATLAS EXPERIMENT

Introduction

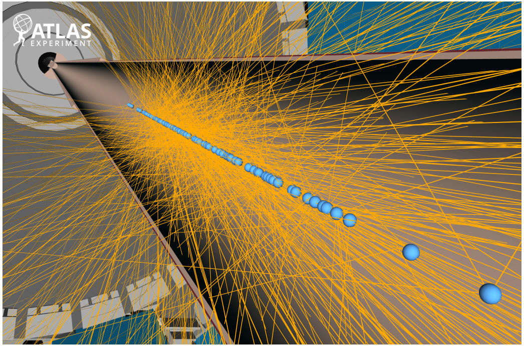
- Event Filter Tracking: ATLAS track trigger upgrade for **HL-LHC**:
  - Increase in luminosity by **4x** to $\mu = 200$ (**10x** greater than original design).
  - ATLAS detector and readout electronics upgrades needed.
  - This project: design dedicated **tracking** coprocessor for Event Filter trigger.
- This talk:
  - Overview of EF Tracking and how it fits into ATLAS HL-LHC upgrade program.
  - Overview of possible FPGA-based solution for EF Tracking.
  - Discussion of **pattern recognition** and **ambiguity resolution** algorithms under study.
  - Outlook and path forward towards building this system.

ATLAS Inner Tracker Upgrade

- For the HL-LHC: building new, all-silicon **Inner Tracker** (ITk):
  - Comprised of 2D pixels and 1D strips.
  - Extends tracking from $\eta = 2.4$ to $\eta = 4.0$; challenging high-pileup environment!



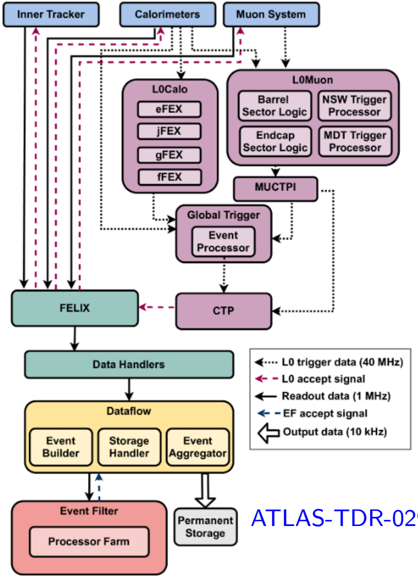
ATL-ITK-PUB-2022-001



UNSG-2022-89

ATLAS Trigger Upgrade Plans

- Two-stage HL-LHC trigger:
  - $40\,\mathrm{MHz} \rightarrow 1\,\mathrm{MHz}$ **Level 0** (hardware)
  - $1\,\mathrm{MHz} \rightarrow 10\,\mathrm{kHz}$ **Event Filter** (CPU)
  - **10x** increase in readout rate.
- ITk data **only** used in the Event Filter:
  - $1\,\mathrm{MHz}$ tracking in regions of interest.
  - Reduced $150\,\mathrm{KHz}$ rate for "full scan" tracking.
- **Offline** algorithms could meet latency requirements:
  - CPU-only system estimated to need 1.9-$2.3\,\mathrm{MW}$.
  - **Entire** datacenter power budget: $2.5\,\mathrm{MW}$!
  - Motivates compute **accelerators**: GPU, FPGA.

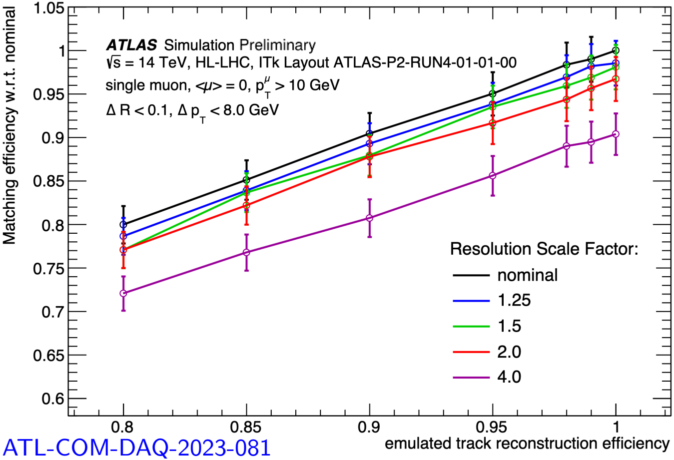


ATLAS-TDR-029

Event Filter Tracking

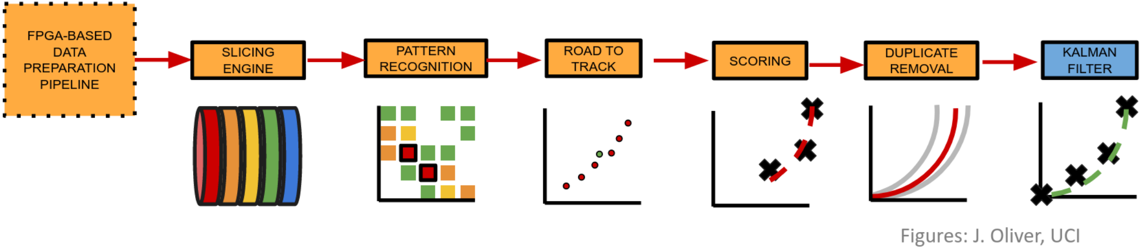
- EF Tracking project: new R&D effort started in 2021:
  - Studying wide range of tracking options on **commodity** hardware: CPUs, GPUs, and FPGAs.

- R&D will continue until **2025**.
- Studies underway to determine performance requirements:
  - Example: tracking performance to reach **98%** muon trigger efficiency.
  - Other metrics: power, bandwidth, latency, maintainability, etc.
- **Technology choice** next year!



ATL-COM-DAQ-2023-081

FPGA Tracking Pipeline

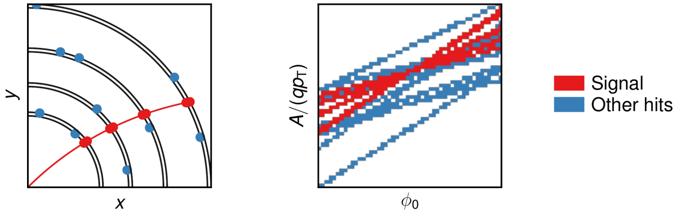


- Current status: designing complete **pipelines**. Example FPGA pipeline:
  - Unpack raw data, perform pattern recognition, ambiguity resolution on board.
  - Send tracks passing ambiguity resolution to CPU for high quality refit.
  - Targeting Xilinx, initial estimates using Alveo U250: 0.6-$0.7\,\mathrm{MW}$ (vs $1.8\,\mathrm{MW}$ for CPU-only!)

Pattern Recognition: Hough Transforms

$$\frac{qA}{p_\mathrm{T}} = \frac{\sin(\phi - \phi_h)}{r_h} \approx \frac{1}{r_h}\phi - \frac{\phi_h}{r_h}$$

- Fast image processing transform.
- Map **hits** (r_h, ϕ_h) into **lines** in track $(\phi, q/p_\mathrm{T})$, find intersections.
- **Coarse** estimate of track params.
- Combine **2D slices** to cover full detector volume.

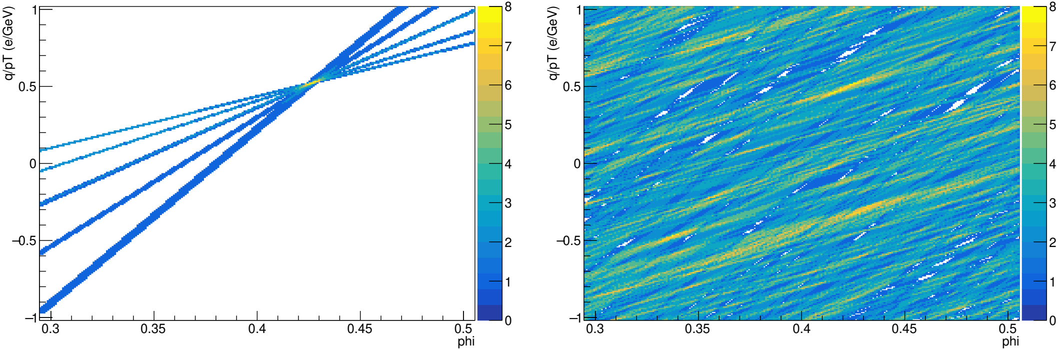


ATL-DAQ-PROC-2016-034

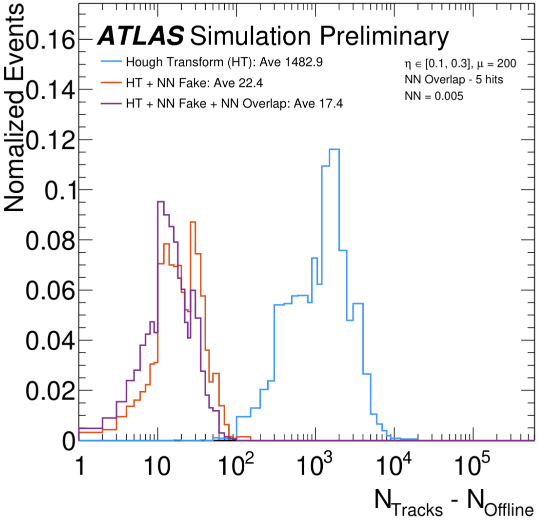
- Multiple versions of transform under study:
  - Baseline version uses four double-sided **strip layers** plus **outermost pixel layer**.
  - Track candidates must have at least 7/9 hits.

Need for Ambiguity Resolution

- Many fake tracks from Hough transform at $\mu = 200$:
  - Single muon track + **O(1000)** fakes in one $0.2 \times 0.2\ \eta \times \phi$ slice!
  - **O(100k)** total; far too much data to pass to CPU for offline track fit.



ATLAS-TDR-029-ADD-1



ATL-COM-DAQ-2022-052

- Score tracks with **fast linear fit** (χ^2) or **neural network**: use to reject duplicates.
- Initial results: algorithms **comparable**:
  - **Two** orders of magnitude rejection.
  - Linear fit also reduces down to **15.1**$\sim$**32.1** tracks per region.
  - Further improvements possible with extra pattern filtering: can get as low as **3.9**.
  - Methods can also estimate **track params** for extension to inner layers.
- Optimization still ongoing!

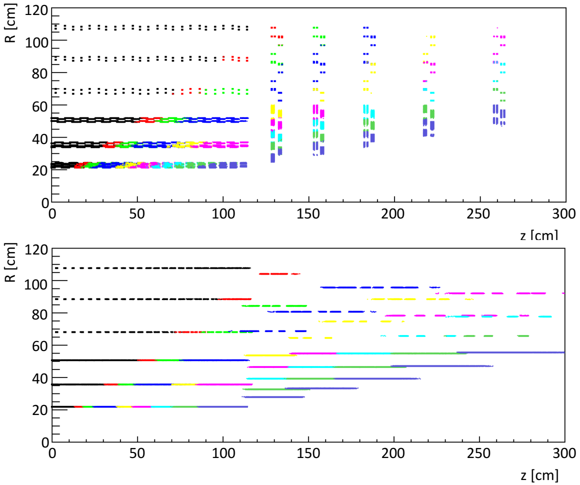
Conclusion and Outlook

- Looking towards EF tracking technology choice next year:
  - Prototype firmware for these algorithms exist, simulation studies continuing.
  - Integration of firmware to create complete pipelines in progress.
  - Will compare different FPGA pipelines to each other and to GPU and CPU based solutions.
  - Lots of great track trigger R&D work even if these options not ultimately selected.
- Thanks for your attention!

Backup

Linear Fitting Challenges

- NN can learn ITk geometry; fit cannot.
- Many different fit constants needed:
  - Cover nonlinearities due to variations in detector geometry.
  - Up to $O(40\text{k})$ in **one** 0.2×0.2 region.
- Solution: **project** physical hit positions onto idealized **fixed-radius** cylinders:
  - Idea from CMS, smooths nonlinearities.
  - Requires track q/p_{T}: take from Hough.
  - Perform fit using transformed coordinates (z', ϕ').
  - Uses **one** set of constants per region.



arXiv:1809.01467

Preliminary FPGA Resource Estimates

- Resource usage and processing time estimates for Alveo U250 implementation.
- Two different versions of Hough algorithm with NN ambiguity resolution.
- Preliminary results, subject to change in complete FPGA pipeline!

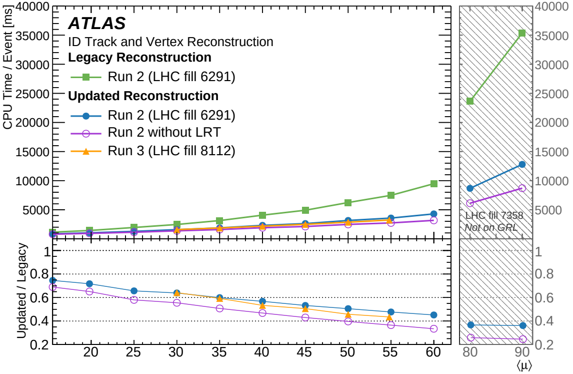
Firmware Block	LUT (%)	flip-flop (FF) (%)	BRAM/ URAM (%)	DSP (%)
	Logic Functions			
PCIe	0.6	0.6	0.3	–
Clustering	1–4	0.14–0.51	1.3–5.4	–
Stub-Finding	0.2	0.05	0.1	–
Slicing Engine	0.1	0.07	13	–
Hough (2D, 0.2×0.2)	39-59	10-30	1-5	1.8-21
Hough (1D, 0.2×0.8)	12	7	27	1
Fake Rejection (NN)	8	1	0.02	29
Duplicate Removal	1	1	–	–
Track Fitting	∼ 10		–	∼ 10
Monitoring (IPbus)	∼ 1		–	–
2nd-Stage Fitting	∼ 10		∼ 30	∼ 15
Total	**44–94**	**32–55**	**33–41**	**55–75**

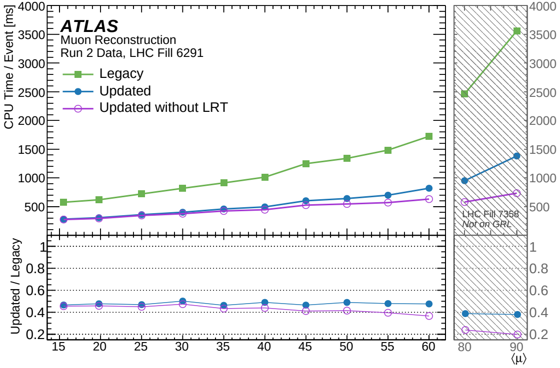
Per Event	Firmware Implementation & Scenario Hough (2D)	Hough (1D)
Loading Time (ms)	1.9-2.8	0.7
Readout Time (ms)	2.7-3.4	1.3
Total Time (ms)	max(loading, readout) 2.7-3.4	1.3
Processing Rate (Hz)	294-534	741
N_{accel}	374-680	270

ATLAS-TDR-029-ADD-1 (Tables 2.8, 2.8)

Offline Tracking Performance

- Latest ATLAS track reconstruction time as function of pileup.
- Significant improvements from adoption of ACTS Common Tracking Software.





Comput Softw Big Sci 8, 9 (2024) (Tables 2.8, 2.8)