Fast tracking for the HL-LHC ATLAS detector

Fabian Klimpel
on behalf of the ATLAS Collaboration
CERN, TU Munich
fklimpel@cern.ch
21.04.2020

Technische Universität München

From Run 2 to Phase 2

Run 2:

Pile-up <μ> reached values up to ≈ 70

Phase 2:

<μ> between 140 and 200 expected

High-μ run performed with Run 2 detector & SW

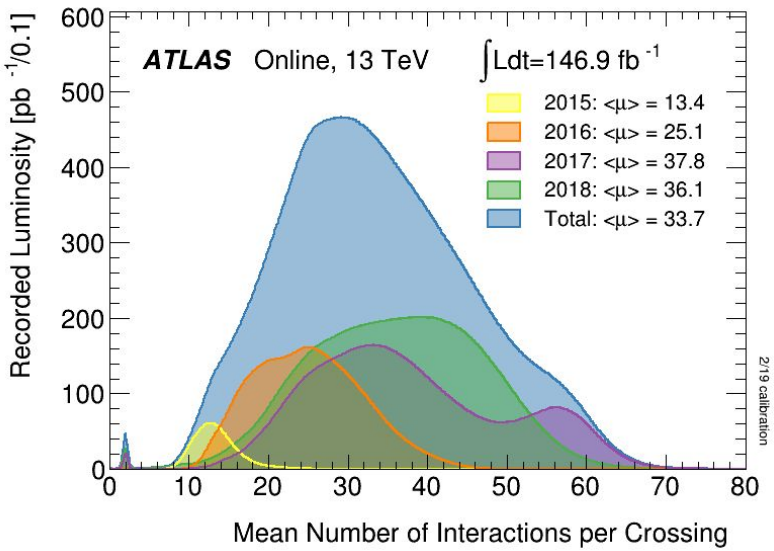
Reconstruction time showed exponential growth

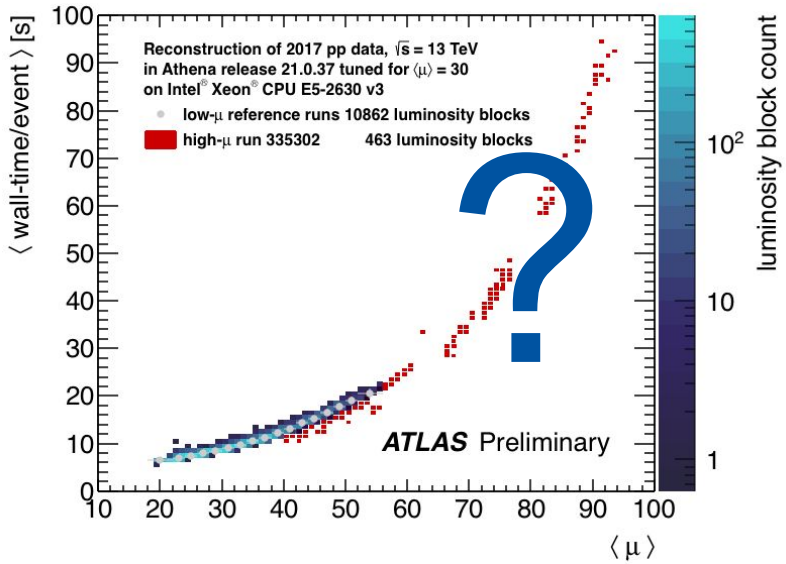
→ HL-LHC will become a challenge for tracking

But:

- Run 2 detector not made for this scenario, target <μ> was ≲ 23
- Applied tracking settings unchanged, not appropriate for high pile-up scenario

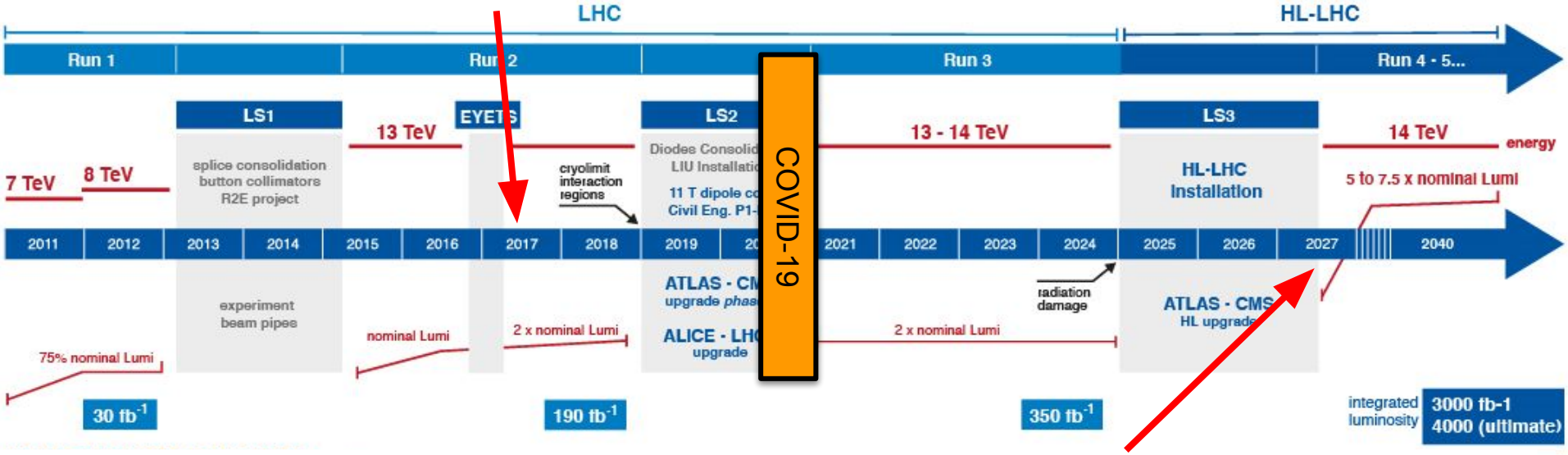
➔ Motivates tracking algorithms R&D





Strategy to address the problem

High-μ run was performed in 2017



6-7 years remaining for preparation

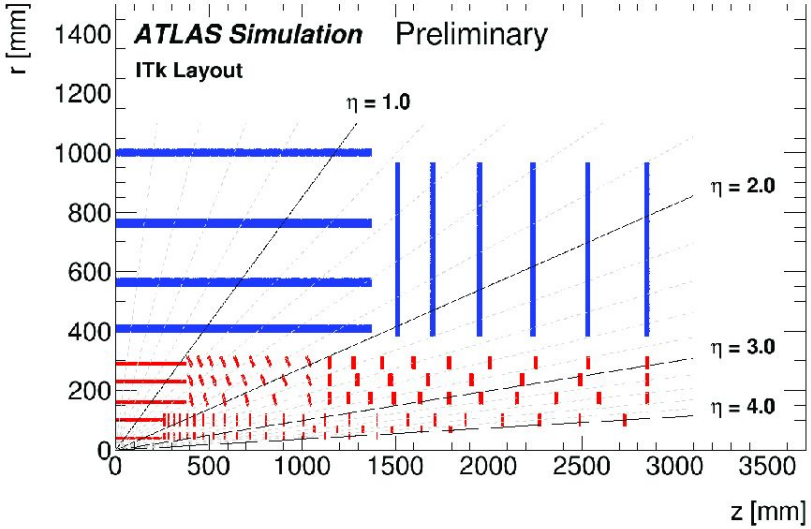
Strategy & outline

1. **Replace** the detector
   → 2. **Adapt** default tracking accordingly
   → 3. **Speed-up** track reconstruction
   → 4. **R&D** to new approaches

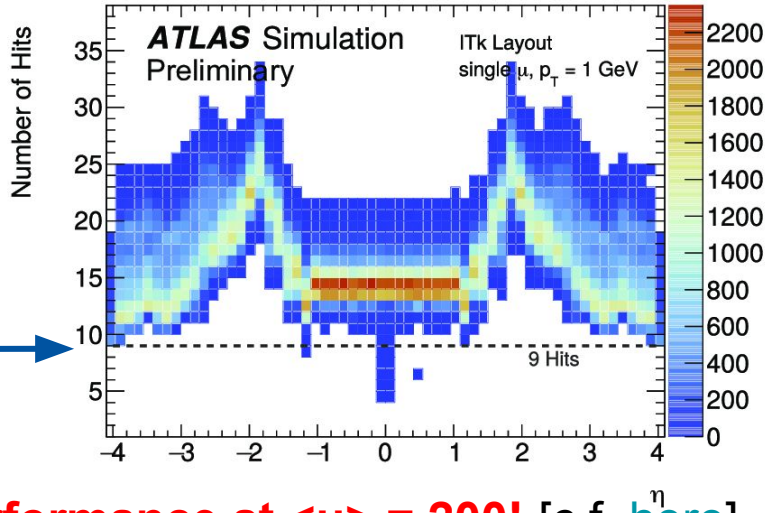
ATLAS Phase 2 detector upgrade: ITk

- Radiation damage of detector reduces performance
- Run-2 detector operates at bandwidth limit
  - Detector was designed for <μ> ≲ 23
  - **T**ransition **R**adiation **T**racker (TRT) has 35-50% occupancy at <μ> = 70

→ Will be replaced by the **I**nner **T**rac**k**er (ITk)



- ITk is designed for
  - **<μ> ≈ 200** (= expected upper limit of HL-LHC)
  - 5 layer pixel coverage of **|η| < 4**
  - 4 layer strip coverage of **|η| < 2.7**
    - **I**nner **D**etector (ID) covers **|η| < 2.5**
  - High granularity **Silicon** (pixel & strip) tracking only
    - No TRT



- ITk design provides
  - ≈ const minimum #hits vs η
  - High tracking performance at all η
  - While minimising pixel surface and bandwidth

- **Optimised for physics and CPU tracking performance at <μ> = 200!** [c.f. here]

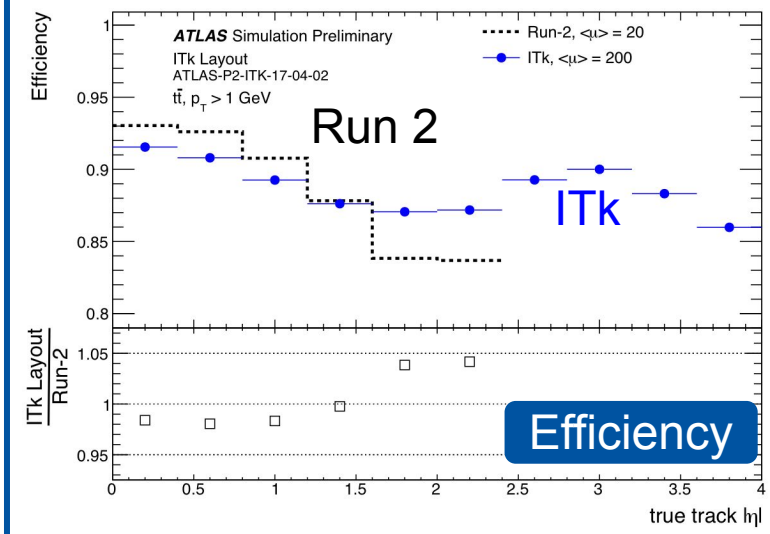
Run 2 tracking software adaption to ITk

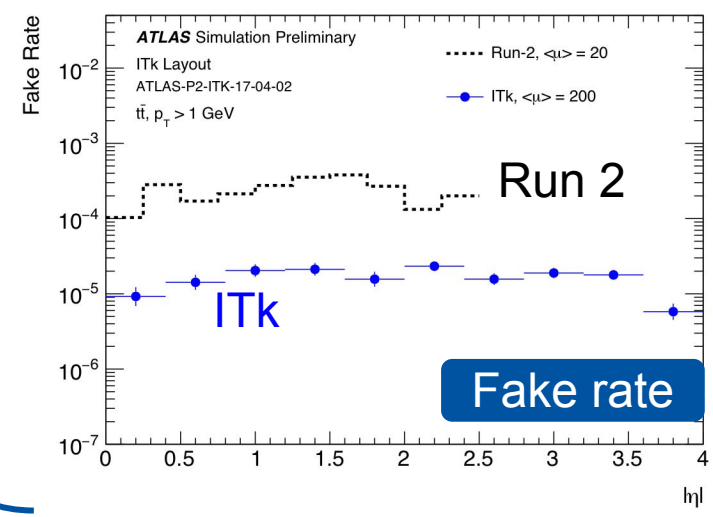
- High #hits vs η in ITk allows tighter cuts

Requirement	$\lvert\eta\rvert < 2.0$	$2.0 < \lvert\eta\rvert < 2.6$	$2.6 < \lvert\eta\rvert < 4.0$
Pixel & Strip hits	≥ 9	≥ 8	≥ 7
Pixel hits	≥ 1	≥ 1	≥ 1
Holes	≤ 2	≤ 2	≤ 2
p_T [MeV]	> 900	> 400	> 400
$\lvert d_0\rvert$ [mm]	≤ 2	≤ 2	≤ 10
$\lvert z_0\rvert$ [cm]	≤ 20	≤ 20	≤ 20

- Similar performance vs η
  - **Stable, high efficiency** vs η expected
  - Stable tracking efficiency up to **η = 4**
- Improved purity of working point
  - **Reduced fake rate** expected
- Results demonstrate excellent tracking performance of the ITk

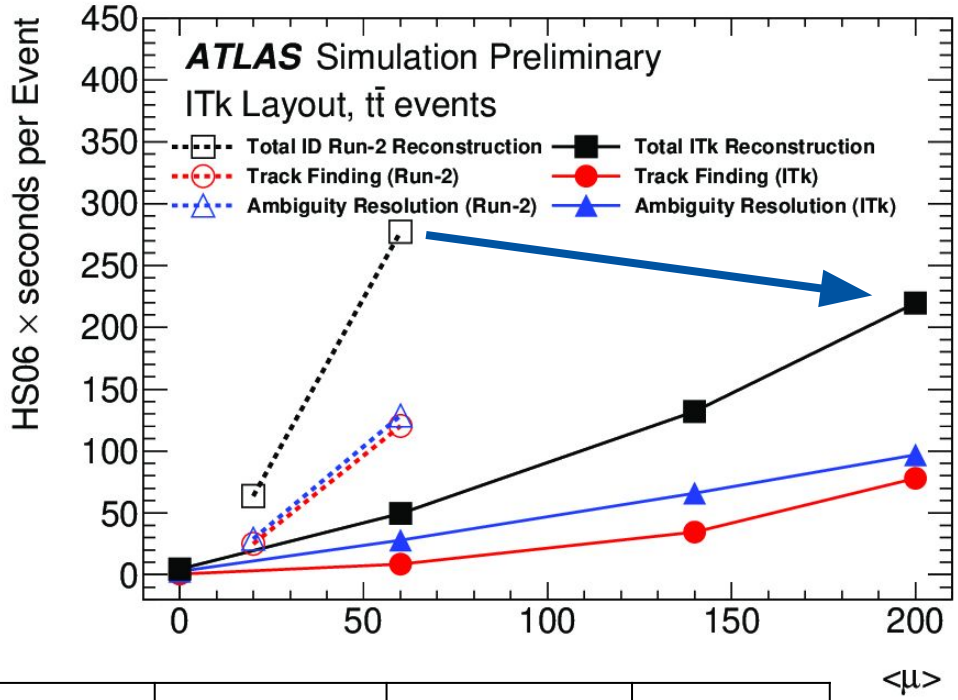
Run 2,<μ>=20 vs. ITk,<μ>=200





CPU requirements for ITk tracking

- ITk tracking using Run 2 SW
- Biggest contributions:
  - (Silicon) Track Finding
  - Ambiguity Resolution
- Major contribution in Run 2 by TRT
- Tracking code usually tuned for physics performance



Detector	Pile-up	Cluster Finding	Space Points	Si Track Finding	Ambiguity Resolution	TRT+Back Tracking	Primary Vertex
ITk	200	22	6.5	78	97	-	6
ID	20	1.5	0.7	23	15	19	0.5

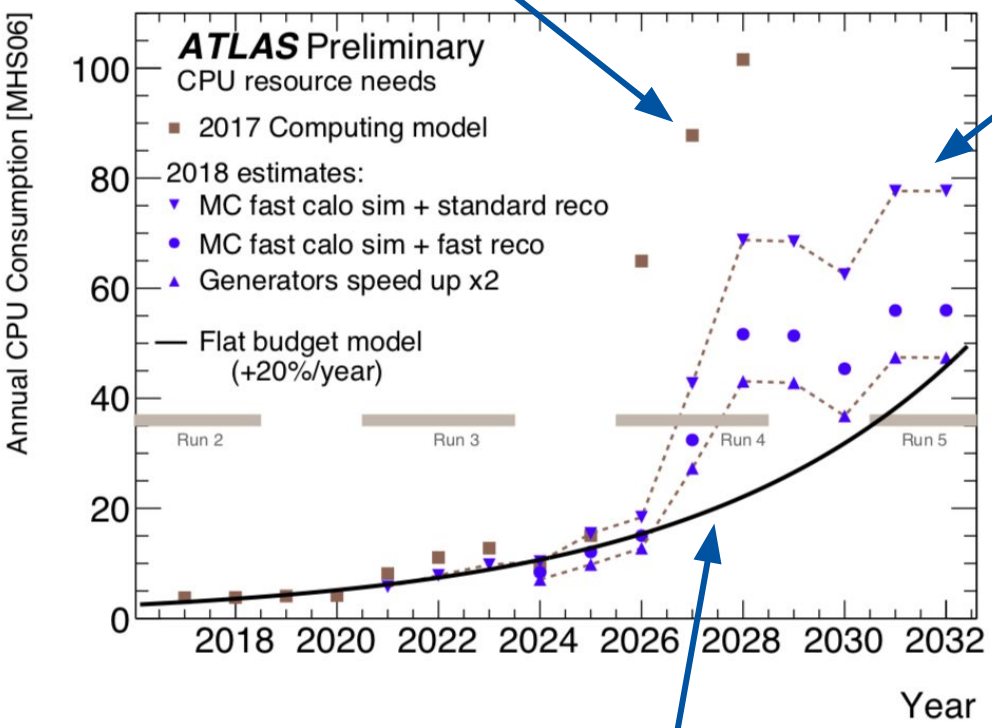
(Numbers in HS06s)

- HL-LHC CPU performance with ITk **exceeds** Run 2 CPU performance with ID
  - Difference arises from optimised detector and adapted track selection

Phase 2 computing

Extrapolation from **Run 2**

Update based on **ITk** and **current** reconstruction software



CPU extrapolation assuming constant budget

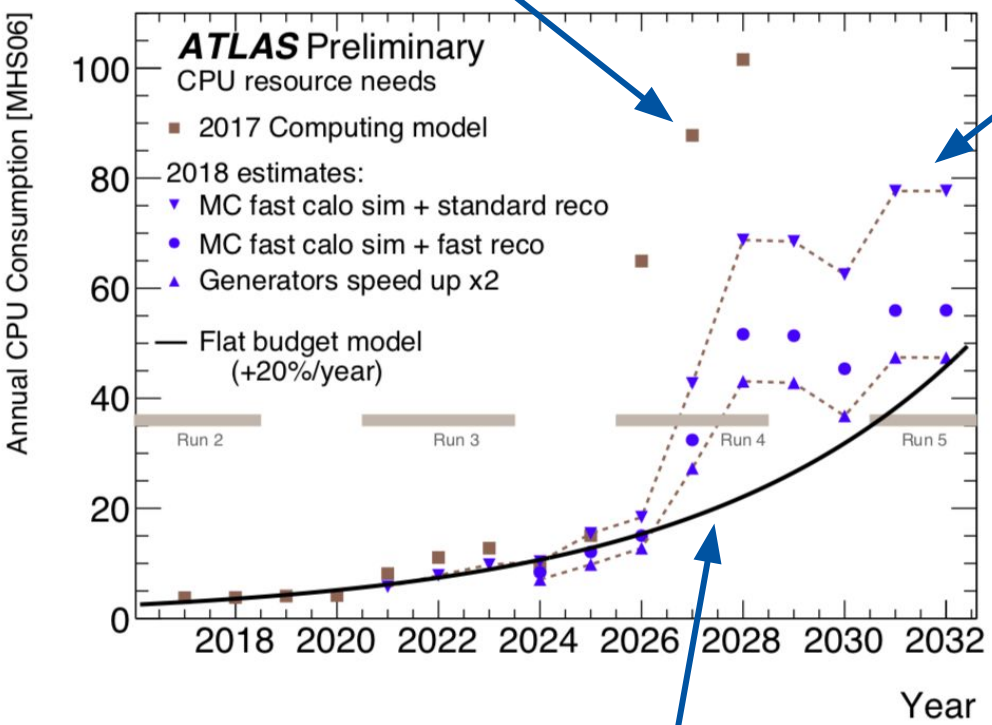
- Significant improvement achieved
- Phase 2 Reconstruction would still require **≈ 45%** of estimated CPU needs from 2018
- Still above flat budget model

ATLAS Phase 2 Computing CDR will update the prediction in 2 weeks

Phase 2 computing

Extrapolation from **Run 2**

Update based on **ITk** and **current** reconstruction software



CPU extrapolation assuming constant budget

- Significant improvement achieved
- Phase 2 Reconstruction would still require **≈ 45%** of estimated CPU needs from 2018
- Still above flat budget model

<u>How can this gap be closed?</u>

Optimise current tracking software

Reduce CPU consumption of **existing** workflow

R&D

Reduce CPU by developing something **new**

ATLAS Phase 2 Computing CDR will update the prediction in 2 weeks

Run 2 tracking software workflow

Pre-processing
- Pixel/SCT clustering
- TRT drit circle formation
- Space point formation

Combinatorial track finder
- Iterative workflow
  a. Pixel seeds
  b. Pixel & SCT seeds
  c. SCT seeds
- Combinatorial Kalman Filter
- Brem.recovery in EM ROI

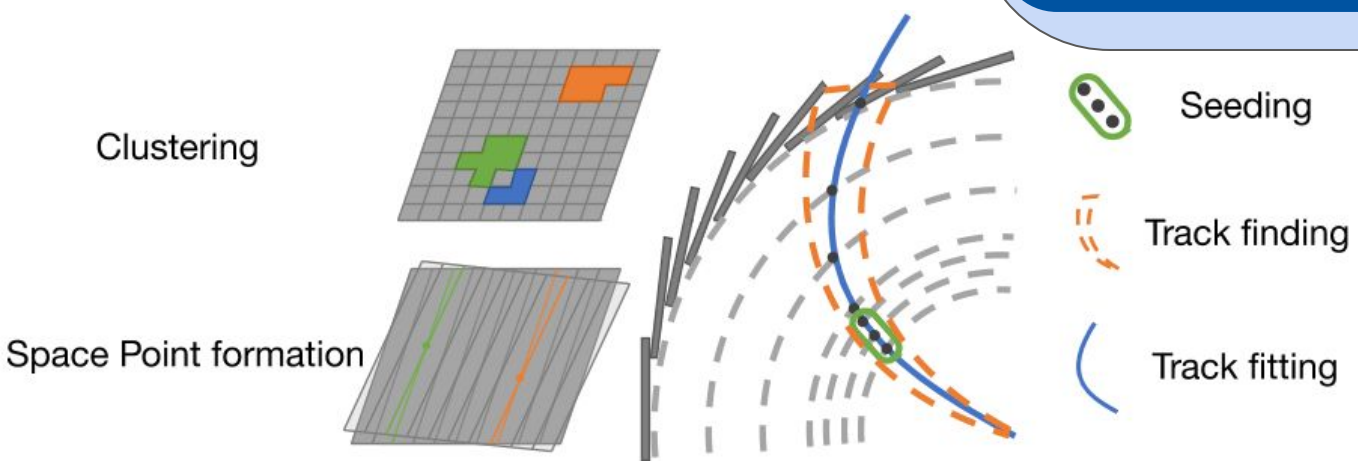
Tracking in pixel & SCT

Ambiguity resolution
- Hole search
- Track scoring according to quality
- NN cluster splitting in jets
- Precise least-squares fit with Brem.recovery
- Apply final selection cuts

Not further discussed here

Vertexing

TRT related tracking



Fast track reconstruction: Overview

- Motivated by software oriented trigger tracking
- Starting point:
  - Classical workflow
  - ITk geometry
  - <μ> = 140 and 200
- Target:
  - Reduce CPU as far as possible
  - With minimal physics performance losses
- Workflow consists of:
  - Pre-processing
  - Combinatorial track finder
  - Ambiguity resolution
- CPU timing baseline:

<μ>	Cluster Finding	Space points	Si Track Finding	Ambiguity resolution
140	17.1	6.0	41.1	58.2
200	26.3	8.6	85.8	92.0

(Numbers in HS06s)

Pre-processing
- Pixel/Strip clustering
- Space point formation

↓

Combinatorial track finder
- Iterative workflow
  a. Pixel seeds
  b. Strip seeds
- Combinatorial Kalman filter
- Brem.recovery in EM ROI

↓

Ambiguity resolution
- Hole search
- Track scoring according to quality
- NN cluster splitting in jets
- Precise least-squares fit with Brem.recovery
- Apply final selection cuts

Fast track reconstruction: Strategy

- Disable Ambiguity resolution
  - Most expensive contribution
  - No precise track fit → Loss of resolution
  - No NN cluster splitting → Loss of b-tagging Performance
- Offload to track finder
  - Suppress duplicates and apply final cuts:

Requirement	$\|\eta\| < 2.0$	$2.0 < \|\eta\| < 2.6$	$2.6 < \|\eta\| < 4.0$
Pixel & Strip hits	**≥ 9** (7)	**≥ 8** (7)	**≥ 7** (7)
Unique hits	**≥ 7** (1)	**≥ 6** (1)	**≥ 5** (1)
Shared Hits	**≤ 2** (no cut)	**≤ 2** (no cut)	**≤ 2** (no cut)
p_T [MeV]	**> 1000** (900)	**> 400** (400)	**> 400** (400)
$\|z_0\|$ [cm]	**≤ 15** (20)	**≤ 15** (20)	**≤ 15** (20)

- Cuts tightened compared to default (brackets)
  - (… which also speeds up track finding!)
- Material model & cluster corrections approximated but full cluster calibration used
- Combinatorial Kalman filter estimates final track parameters

Pre-processing
- Pixel/Strip clustering
- Space point formation

↓

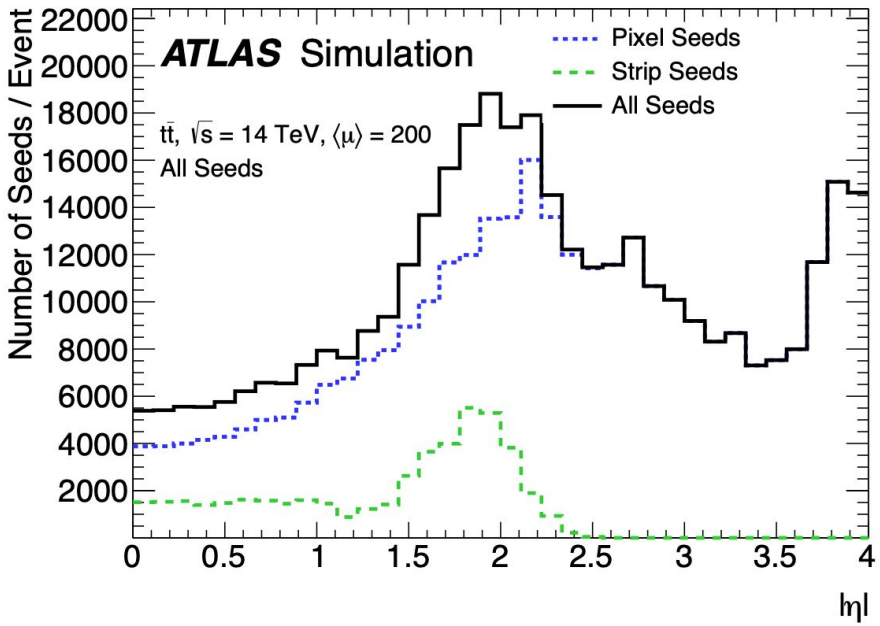
Combinatorial track finder
- Iterative workflow
  a. Pixel seeds
  b. Strip seeds
- Combinatorial Kalman filter
- Brem.recovery in EM ROI

↓

~~**Ambiguity resolution**~~
- ~~Hole search~~
- ~~Track scoring according to quality~~
- ~~NN cluster splitting in jets~~
- ~~Precise least-squares fit with Brem.recovery~~
- ~~Apply final selection cuts~~

Fast track reconstruction: Strategy

- Seeding in Run 2 took about **20%** of CPU time
- For ITk@<μ>=200 it rises to **50%**



Pre-processing
- Pixel/Strip clustering
- Space point formation

Combinatorial track finder
- Iterative workflow
  a. Pixel seeds
  b. ~~Strip seeds~~
- Combinatorial Kalman filter
- ~~Brem recovery in EM ROI~~

(Brem recovery temporarily turned off)

- ITk consists of **5 pixel layers**
- Most seeds produced by ITk pixel detector
- Pixel detector covers the full |η| < 4 range
- Tuning to improve <u>pixel seed purity</u> while <u>dropping</u> strip seeding
  - Purity reduces time in road building and track finding

Fast track reconstruction: Strategy

- Code optimisation for
  - Pixel Clustering and Space Point formation
  - Strip Clustering
- Strip Space Points
  - Expensive to calculate
  - Only required for seeding
    → Can be dropped

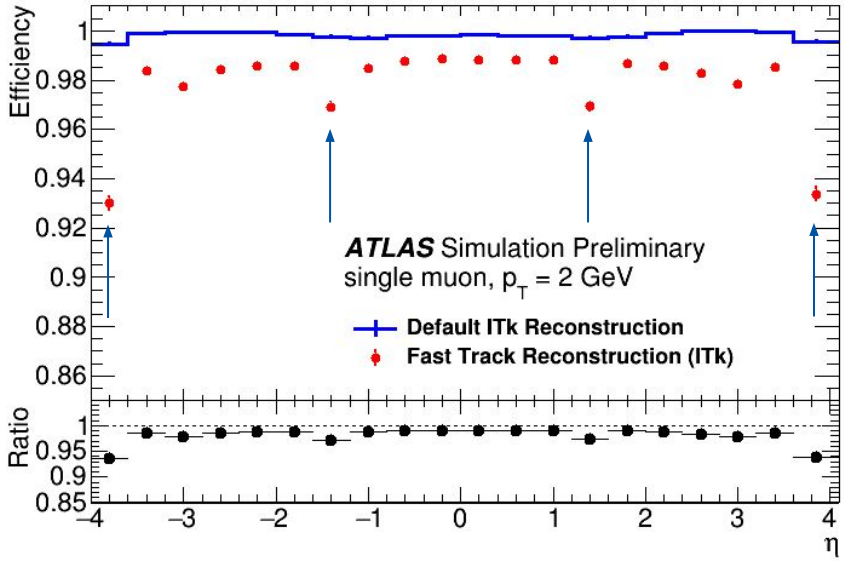
- RD53B frontend chip
  - Reads-out multiple pixel in tree-like way
    - Allows reading of pixel clusters
  - Allows for significant data compression
  - ITk @ <μ>=200 event size:
    - 580 kB for pixels
    - 470 kB for strips
  - But no decoding software yet
    - Run 2 raw data decoding times scaled to ITk sizes
    - ROOT file decoding used for MC

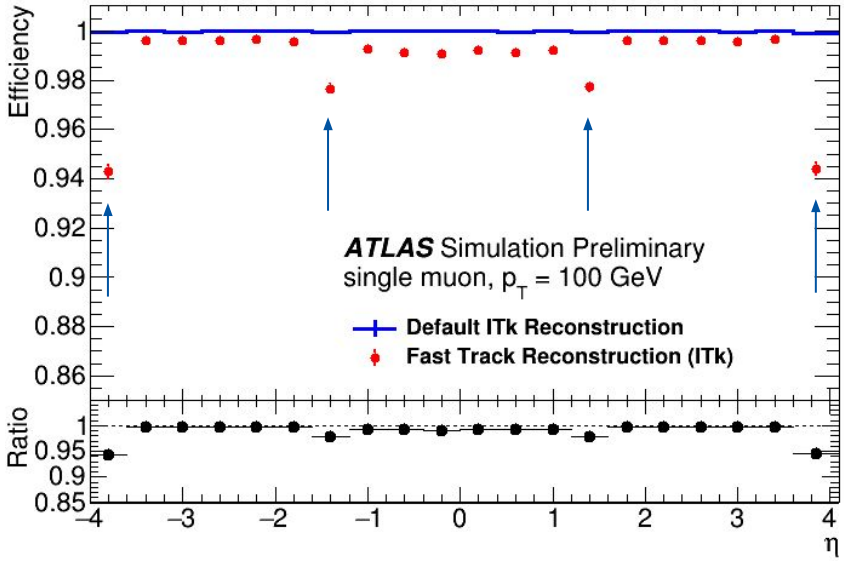
Pre-processing
- Pixel/Strip clustering
- Space point formation

↓

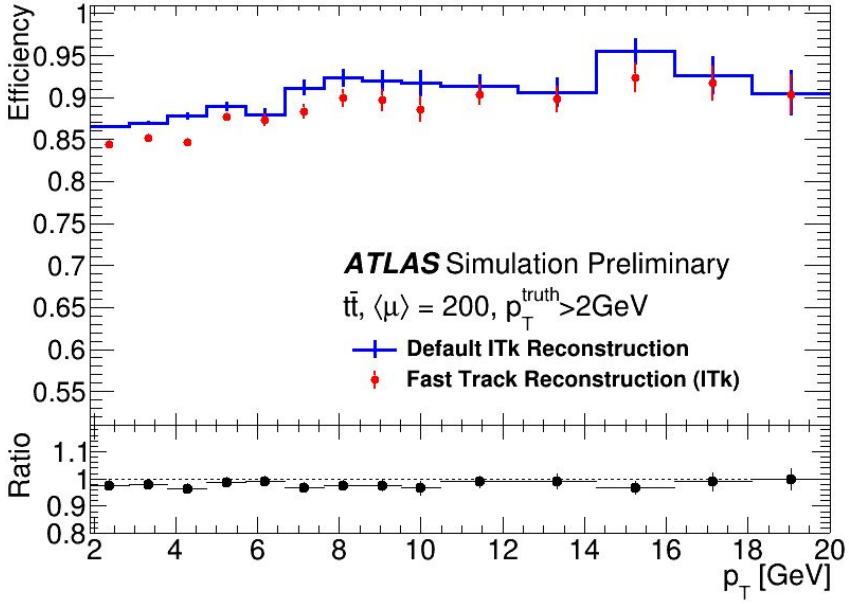
Combinatorial track finder
- Iterative workflow
  a. Pixel seeds
- Combinatorial Kalman filter

Fast track reconstruction: Performance

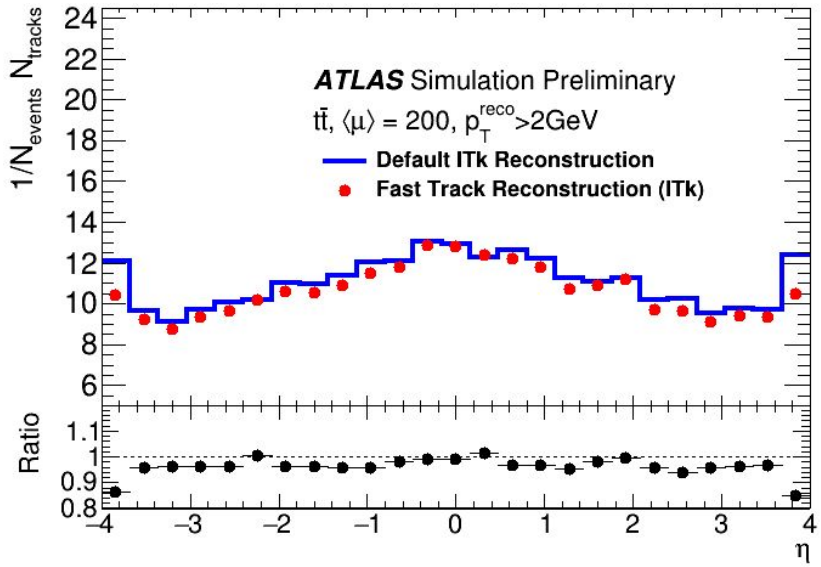


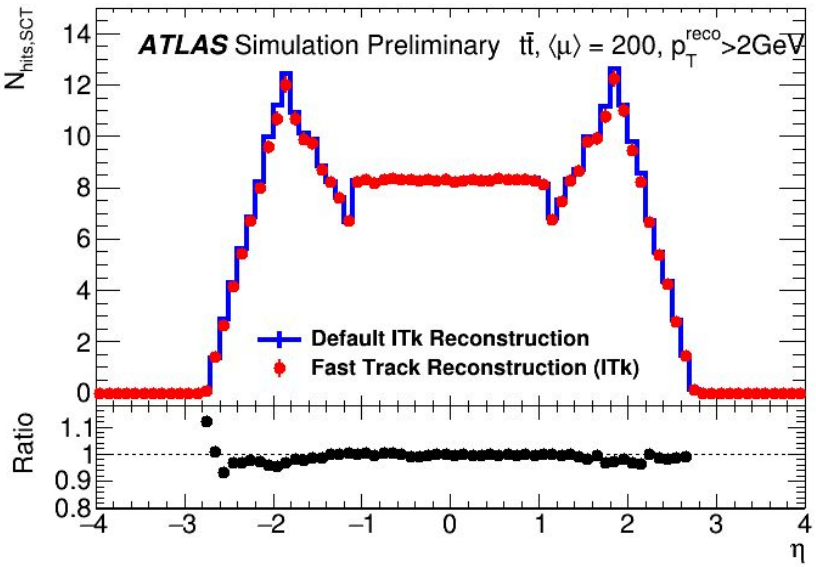


- Small efficiency losses for all η/p_T
- More significant:
  - Barrel/End-cap transition
  - $|\eta| \approx 4$
  - Low-p_T
- Larger losses at low-p_T due to approximations in accounting for material interaction

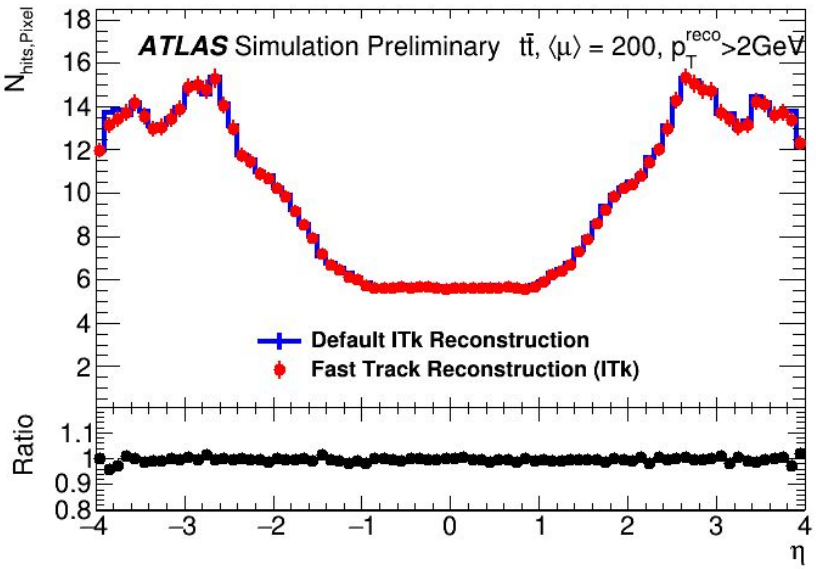


Fast track reconstruction: Performance



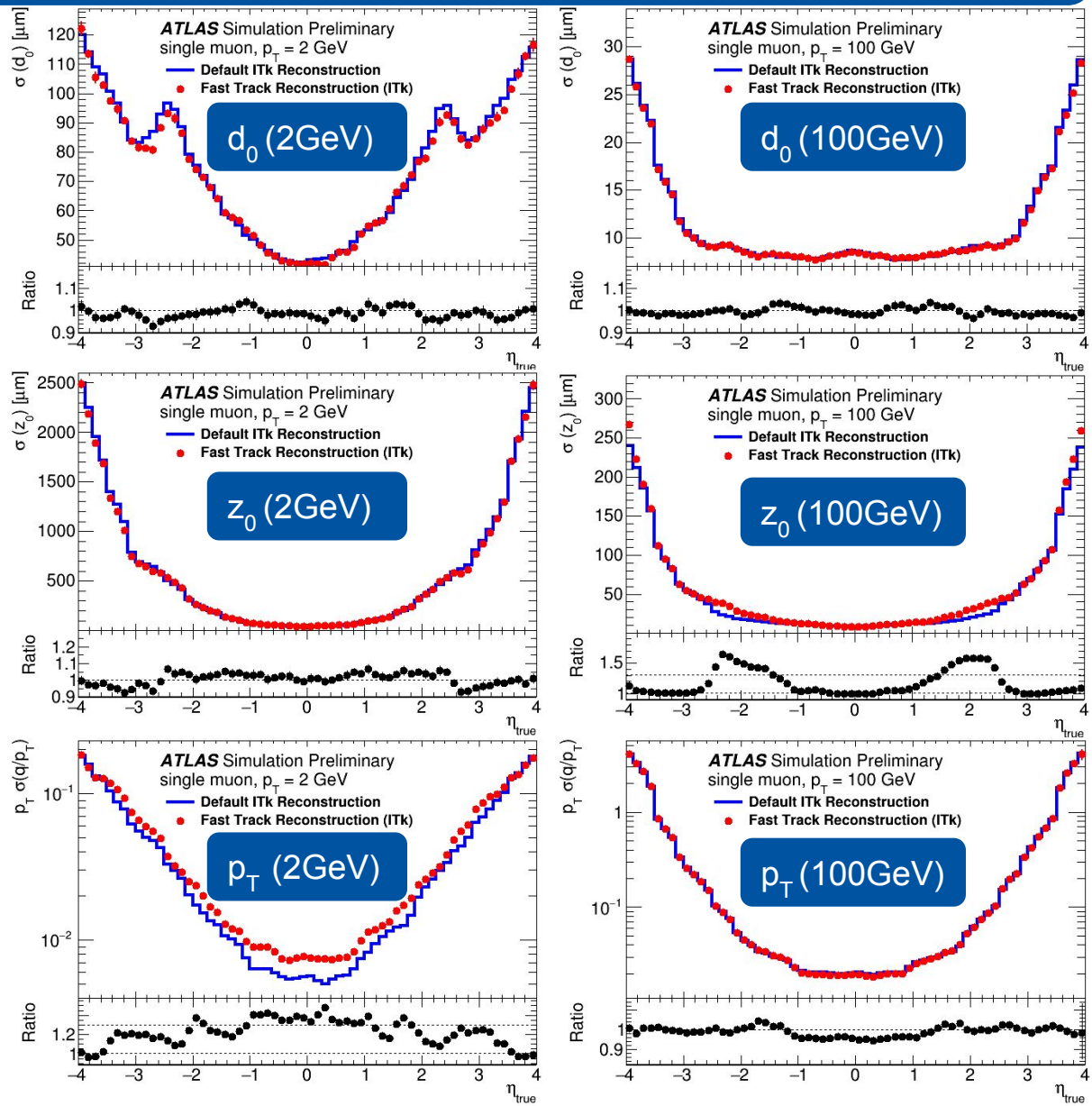


- Slightly fewer tracks found
  - Effect of lower efficiency
  - Fake-rate <u>insignificantly</u> changed
- Hit association for pixel & strip detector **almost identical** to default reconstruction



Fast track reconstruction: Performance

- Resolution parameters:
  - Single muons provide insight
- For 2 GeV:
  - 20% loss in p_T resolution
  - Multiple scattering is major effect
  - Issue of material model
- For 100 GeV:
  - 50% loss in z_0 around $|\eta| \approx 2$
  - Result of cluster correction
- Sources are understood!



Fast track reconstruction: CPU Performance

(Numbers in HS06s)

<μ>	Tracking	Byte Stream Decoding	Cluster Finding	Space Points	Si Track Finding	Ambiguity resolution	Total ITk
140	default	1.2(*)	17.1	6.0	41.1	58.2	123.6
	fast		4.5	0.9	12.4	-	19.0
200	default	1.6(*)	26.3	8.6	85.8	92.0	214.3
	fast		6.3	1.2	22.6	-	31.7

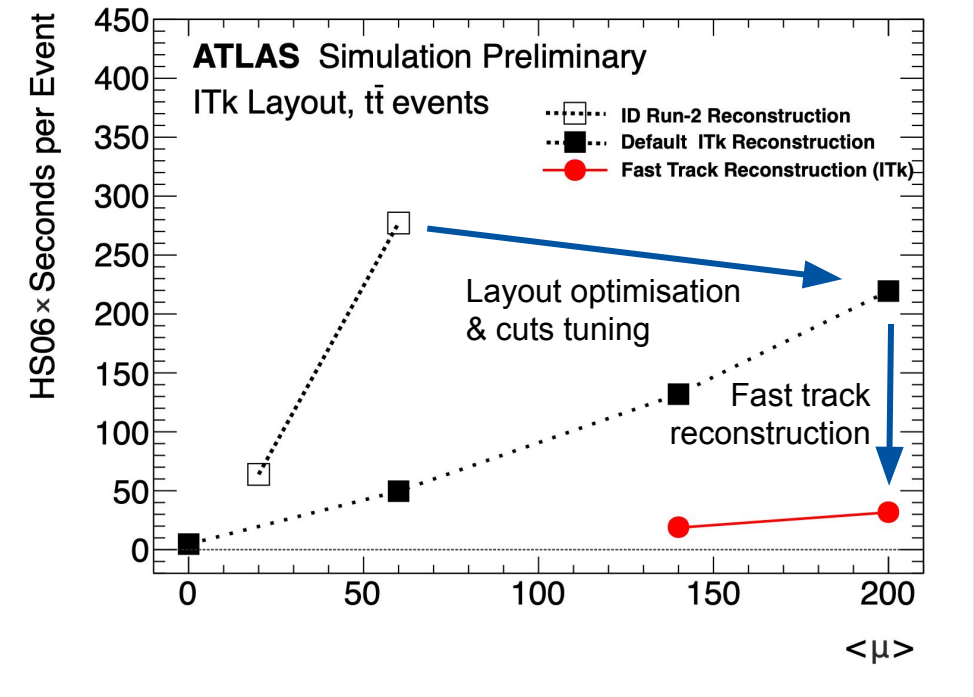
(* Scaled from Run 2 events)

Outstanding speed-up:

- **6-7x faster** for <μ> = 140/200
- Valid for p_T > 1 (0.4) GeV
- Reconstruction of the full detector

And still:

- **Result is based on Run 2 software**
  - No benefits from acts

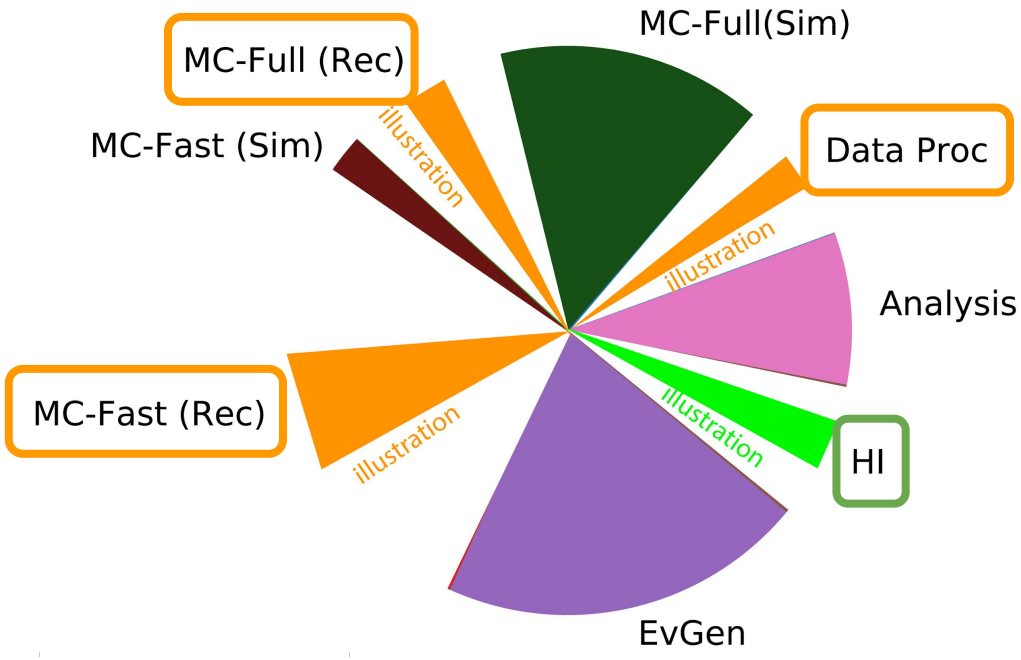


Back to Phase 2 computing

Lessons from fast tracking study allow to significantly reduce ATLAS CPU needs for Phase 2

- Affects requirements for **MC reconstruction** and **Data Processing**
- Reconstruction of **Heavy Ions** will also benefit
- CPU for **event generators** and **Geant4** will dominate
- Tracking is **not** dominating the CPU consumption

- Naive Illustration for the effect of Fast Track Reconstruction on the computing model:



R&D

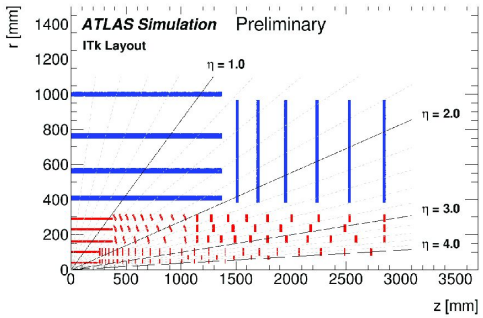
- Fast Track Reconstruction is sufficiently fast for Phase 2 tracking
  - But physics performance needs further improvements
    - A different software system will be used to achieve it → **Acts**
- **Acts:**
  - Open source tracking project from ATLAS/Belle-II/FCC… contributes [c.f. here]
  - Provides modular tracking tools/algorithms with **simple interfaces for R&D**
  - Important to **improve all reconstructions** (e.g. μ, e/γ, b-tagging,..)

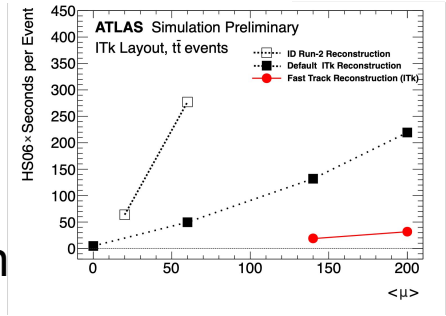
- Towards new harbors: Machine learning
- **TrackML:**
  - Public tracking performance competition
- Adaption to new hardware: **FPGA, GPU**
- Testing new ideas and inspirations

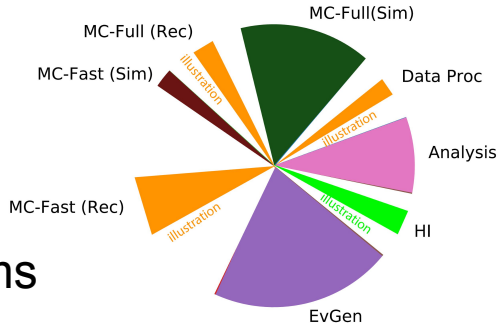
→ Importance of the tracking community & workshops

Summary

- HL-LHC is a challenge for track reconstruction
- Appropriate hardware
  - **ITk** is optimised for tracking at 200 pile-up
- **CPU** requirements are very strict
  - Prototype study for fast reconstruction done
- **Fast Track Reconstruction**
  - CPU oriented optimisation of classical tracking
  - **8x** faster for ITk,<μ>=200 vs. ID,<μ>=60
    - Tracking is **not** dominating Phase 2 CPU consumption
  - Achieved physics performance:
    - Exceeds trigger requirements
    - Improvements required for offline reconstruction
- **Acts** will be essential for Phase 2 physics performance
  - Provides R&D platform for track reconstruction algorithms
  - Beneficial to **all reconstructions**







- ATLAS is actively investing in researching the development of new concepts and adaption to new hardware architectures