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
**From Run 2 to Phase 2**

**Run 2:**
- Pile-up $<\mu>$ reached values up to $\approx 70$

**Phase 2:**
- $<\mu>$ between 140 and 200 expected

High-$\mu$ run performed with Run 2 detector & SW

Reconstruction time showed exponential growth

$\rightarrow$ HL-LHC will become a challenge for tracking

**But:**
- Run 2 detector not made for this scenario, target $<\mu>$ was $\lesssim 23$
- Applied tracking settings unchanged, not appropriate for high pile-up scenario

$\Rightarrow$ Motivates tracking algorithms R&D
Strategy to address the problem

High-\(\mu\) run was performed in 2017

**Strategy & outline**

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

6-7 years remaining for preparation

COVID-19

LS1
splice consolidation
button collimators
R2E project

13 TeV


75% nominal Lumi
30 fb\(^{-1}\)

7 TeV

EYETS
cryocooler interaction regions

14 TeV

LS2
Diodes Consolidation
LIU installation
11 T dipole coil
Civil Eng. PIP

13 - 14 TeV

2019 2020 2021 2022 2023 2024

2 x nominal Lumi

ATLAS - CMS
upgrade phase

350 fb\(^{-1}\)

6-7 years

ATLAS - CMS
HL upgrade

14 TeV

HL-LHC installation
5 to 7.5 x nominal Lumi

3000 fb\(^{-1}\)
4000 (ultimate)

integrated luminosity

2 x nominal Lumi

Run 1 Run 2 Run 3 Run 4 • 5...
Fast tracking for the HL-LHC ATLAS detector

Fabian Klimpel

ATLAS Phase 2 detector upgrade: ITk

- Radiation damage of detector reduces performance
- Run-2 detector operates at bandwidth limit
  - Detector was designed for $<\mu> \lesssim 23$
  - Transition Radiation Tracker (TRT)
    - has 35-50% occupancy at $<\mu> = 70$

  → Will be replaced by the Inner Tracker (ITk)

- ITk is designed for
  - $<\mu> \approx 200$ (= expected upper limit of HL-LHC)
  - 5 layer pixel coverage of $|\eta| < 4$
  - 4 layer strip coverage of $|\eta| < 2.7$
    - Inner Detector (ID) covers $|\eta| < 2.5$
  - High granularity Silicon (pixel & strip) tracking only
    - No TRT

- ITk design provides
  - $\approx$ const minimum #hits vs $\eta$
  - High tracking performance at all $\eta$
  - While minimising pixel surface and bandwidth

- Optimised for physics and CPU tracking performance at $<\mu> = 200!$ [c.f. here]
Run 2 tracking software adaption to ITk

- High #hits vs $\eta$ in ITk allows tighter cuts

| Requirement       | $|\eta| < 2.0$ | $2.0 < |\eta| < 2.6$ | $2.6 < |\eta| < 4.0$ |
|-------------------|--------------|-----------------|-----------------|
| Pixel & Strip hits | $\geq 9$     | $\geq 8$       | $\geq 7$       |
| Pixel hits        | $\geq 1$     | $\geq 1$       | $\geq 1$       |
| Holes             | $\leq 2$     | $\leq 2$       | $\leq 2$       |
| $p_T$ [MeV]       | $> 900$      | $> 400$        | $> 400$        |
| $|d_0|$ [mm]        | $\leq 2$     | $\leq 2$       | $\leq 10$      |
| $|z_0|$ [cm]        | $\leq 20$    | $\leq 20$      | $\leq 20$      |

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

<table>
<thead>
<tr>
<th>Detector</th>
<th>Pile-up</th>
<th>Cluster Finding</th>
<th>Space Points</th>
<th>Si Track Finding</th>
<th>Ambiguity Resolution</th>
<th>TRT+Back Tracking</th>
<th>Primary Vertex</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITk</td>
<td>200</td>
<td>22</td>
<td>6.5</td>
<td>78</td>
<td>97</td>
<td>-</td>
<td>6</td>
</tr>
<tr>
<td>ID</td>
<td>20</td>
<td>1.5</td>
<td>0.7</td>
<td>23</td>
<td>15</td>
<td>19</td>
<td>0.5</td>
</tr>
</tbody>
</table>

(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

- Significant improvement achieved
- Phase 2 Reconstruction would still require \( \approx 45\% \) of estimated CPU needs from 2018
- Still above flat budget model

**ATLAS** Preliminary
CPU resource needs

- **2017 Computing model**
- **2018 estimates:**
  - MC fast calo sim + standard reco
  - MC fast calo sim + fast reco
  - Generators speed up x2

- Flat budget model
  (+20%/year)

**Year**

- **Run 2**
- **Run 3**
- **Run 4**
- **Run 5**

**ATLAS Phase 2 Computing CDR** will update the prediction in 2 weeks
Extrapolation from Run 2

Update based on ITk and current reconstruction software

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

How can this gap be closed?

- Optimise current tracking software
- R&D
  - Reduce CPU by developing something new

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

Fabian Klimpel

Fast tracking for the HL-LHC ATLAS detector
Run 2 tracking software workflow

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

**Vertexing**

**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

**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**

**Clustering**

**Space Point formation**

**Seeding**

**Track finding**

**Track fitting**

Not further discussed here
Fast track reconstruction: Overview

- Motivated by software oriented trigger tracking
- Starting point:
  - Classical workflow
  - ITk geometry
  - $\mu = 140$ 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:

<table>
<thead>
<tr>
<th>$\mu$</th>
<th>Cluster Finding</th>
<th>Space points</th>
<th>Si Track Finding</th>
<th>Ambiguity resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>140</td>
<td>17.1</td>
<td>6.0</td>
<td>41.1</td>
<td>58.2</td>
</tr>
<tr>
<td>200</td>
<td>26.3</td>
<td>8.6</td>
<td>85.8</td>
<td>92.0</td>
</tr>
</tbody>
</table>

(Numbers in HS06s)
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:

<table>
<thead>
<tr>
<th>Requirement</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0(</td>
<td>\eta</td>
<td>&lt; 2.0)</td>
</tr>
<tr>
<td>Pixel &amp; Strip hits</td>
<td>≥ 9 (7)</td>
<td>≥ 8 (7)</td>
<td>≥ 7 (7)</td>
</tr>
<tr>
<td>Unique hits</td>
<td>≥ 7 (1)</td>
<td>≥ 6 (1)</td>
<td>≥ 5 (1)</td>
</tr>
<tr>
<td>Shared Hits</td>
<td>≤ 2 (no cut)</td>
<td>≤ 2 (no cut)</td>
<td>≤ 2 (no cut)</td>
</tr>
<tr>
<td>(p_T) [MeV]</td>
<td>&gt; 1000 (900)</td>
<td>&gt; 400 (400)</td>
<td>&gt; 400 (400)</td>
</tr>
<tr>
<td>(</td>
<td>z_0</td>
<td>) [cm]</td>
<td>≤ 15 (20)</td>
</tr>
</tbody>
</table>

- 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
  - Pixel seeds
  - 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

Fabian Klimpel
Fast tracking for the HL-LHC ATLAS detector
Fast track reconstruction: Strategy

- Seeding in Run 2 took about **20%** of CPU time
- For ITk@\(\langle \mu \rangle = 200\) it rises to **50%**

![Atlas Simulation Graph](image)

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

**Combinatorial track finder**
- Iterative workflow
  - Pixel seeds
  - 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 |\(\eta| < 4\) range
- Tuning to improve **pixel seed purity** while **dropping** 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 @ $<\mu>$=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 $\eta/p_T$
- 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 insignificantly 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

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

(* 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 a**ts.**
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:
• 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. $\mu$, e/$\gamma$, 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, $<\mu> = 200$ vs. ID, $<\mu> = 60$
    - 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**