Tracking for Triggering Purposes in ATLAS

John Baines on behalf of the ATLAS Collaboration

Connecting the Dots Workshop: Vienna 22-24 Feb 2016
Overview

- The ATLAS Detector
  - The ATLAS Tracker & Trigger
- Trigger Tracking Challenges
- Solutions Current & Future:
  - Custom Fast algorithms
  - Hardware-based Tracking: Fast TracKer (FTK)
  - Use of co-processors/accelerators
- ATLAS Trigger Tracking for HL-LHC
- Summary
The ATLAS Detector

Si Tracker:
- 12 Si detector planes
- 4 Pixel layers (3 in Run-1)
- 4 double SCT layers
- ~100M channels
Muon:
• Match ID and Muon track segments

Electron:
• Match track and calorimeter cluster

Lepton isolation:
• Look for reconstructed tracks within cone around lepton

Hadronic Tau lepton decays:
• Distinguish hadronic Tau decay from jet background
• Based on distribution of tracks in Core and Isolation Cones
Tracking in the Trigger (2)

Jets originating from $b$-quark ($b$-jets):
- Identify primary and secondary vertex

B-Physics Triggers:
- Invariant mass cuts e.g.
- Identify $J/\psi(\mu,\mu)$ or $B(\mu,\mu)$

Pile-up suppression:
- Identify number of interactions (no. vertices)
- Identify which particles in jet come from primary vertex
LHC & High Luminosity LHC

- Addition of Inner pixel layer (IBL)
- Trigger Upgrades
- Addition of FTK
- Further trigger upgrades
  - Concurrent processing on many-core cpu (+ accelerators)
- Upgraded ATLAS Detector incl.
  - Replacement of Inner Tracker
  - Upgraded Trigger

FTK: Fast TracKer: hardware based tracking

First W→eV 7 TeV collision data.
Trigger Requirements, Challenges & Solutions

Requirements:
- High Efficiency; Low fake rate
- Excellent track parameter resolution

Challenges:
- **Event complexity**: many superimposed collisions
  - 45 (Run 1) to 69 (Run 3) to 200 (HL-LHC)
- **High rate**:
  - 100 kHz Run 2 & 3 to 400 kHz (1MHz) HL-LHC
- **Short Time**:
  - finite HLT farm size => ~300ms/event for ALL Reco.
  - ~factor 50 faster than offline
- **Huge number of hit combinations**
  for current luminosities (~30 interactions):

  - $10^5$ Space-points
  - $10^9$ Triplets
  - $10^4$ Seeds
  - $10^3$ Tracks

  Space-point = Pixel cluster or SCT cluster-pair (φ+stereo)

Solutions for the Trigger:
- **Reconstruction in Regions of Interest**
  - Reduced detector volume reconstructed
  - Knowledge of L1 trigger type enables optimised reconstruction
- **Two stage tracking**:
  - **Fast Tracking**: Initial loose trigger selection using reduced resolution tracks
  - **Precision Tracking**: full precision tracks for final trigger selection
- **Limit hit combinations**:
  - Geometrical constraints using RoI, beamspot and possibly primary vertex info
- **Hardware Trigger: FTK (Run 2 & 3)**
  - Pattern matching using custom ASIC
- **Acceleration (Future)**
  - Exploring use of GPGPUs
Trigger in LHC Run 2 & 3

**Level 1**
- Hardware-based (FPGA)
- Calorimeter & Muon
  (+ specialized triggers)

**Fast TracKer:**
- Hardware-based tracking
- Commissioned during 2016
- Full-event track information

**High Level Trigger:**
- Software-based running on CPU farm
- Precision Calo and Muon info.
- Precision Tracking
  - mainly in RoI
  - improved resolution
- Time budget:
  - \(< t \sim 300 \text{ ms / event}\)

Note: In Run-1 HLT was split into Level-2 and Event Filter
Run-1 Trigger Tracking

Two-step Tracking: Fast-Tracking followed by Precision Tracking

1) Fast-Tracking in Run-1 used 2 strategies:

**Histogramming method**: 1

- **Zfinder**: Identify z-positions of vertices
  - Histogram z of pairs/triplets of hits
- **HitFilter**: Identify groups of nearby hits
  - Histogram in $\eta - \phi$ (uses z of vertex for $\eta$)
- **Group Cleaner**: Identify hits on track
  - Histogram in $\phi_0, 1/p_T$
- **Clone Removal & Track Fitting**
  - Remove duplicate tracks sharing hits
  - Determine track parameters
- **Extension to Transition Radiation Tracker**

**Combinatorial method**: 3

- Also used for GPU prototype (see later). Similar to algorithm used in Run2 (next slide).

2) Precision Tracking uses Online configuration of Offline tools

<table>
<thead>
<tr>
<th>Histogramming Method:</th>
<th>Combinatorial Method:</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Time scales linearly with occupancy</td>
<td>+ Does not rely on primary vertex for efficiency</td>
</tr>
<tr>
<td>- potential loss of efficiency if vertex not found</td>
<td>(but vertex info. can increase speed)</td>
</tr>
<tr>
<td></td>
<td>+ More similar to offline algorithm</td>
</tr>
<tr>
<td></td>
<td>- More rapid scaling of execution time with occupancy</td>
</tr>
</tbody>
</table>

1: Sutton, doi:10.1016/j.nima.2007.10.005
Run 2 HLT Tracking

- **Fast Tracking**\(^1\):
  - **Seeding:**
    - Form Triplets of Hits using bins in r and \(\phi\)
    - Conformal mapping to estimate triplet parameters
    - Combinations limited using RoI geometry
  - **Track Following:** extend to all Si layers
  - **Fast Kalman Track Fit**

- **Precision Tracking**\(^2\):
  - Online configuration of Offline Tools\(^3\)
  - Resolve ambiguous hit assignments
  - Remove duplicate tracks
  - Precision track fit (Global \(\chi^2\) algorithm\(^3\))

---

\(^1\) Martin-Haugh 2015 J. Phys.: Conf. Ser. 664 082029
Fast TracKer FTK

FTK:
- Gets SCT & Pixel data direct from detector front-end
- Performs pattern matching & track fitting at up to full Level 1 rate (100 kHz)
- Sends tracks to the HLT: perigee parameters and hits-on-track information (cluster centres)

HLT has access to:
- FTK Tracks in whole event
- HLT Refit of FTK tracks
- HLT tracks in RoI

FTK Tracks
- Primary vertex reconstruction
- Initial track-based selections in whole event or large RoI incl.
  - hadronic Tau-lepton decays
  - B decays including hadrons
  - Lepton isolation; Track-based jets

HLT Refit of FTK tracks in RoI
- Selections requiring higher parameter precision incl.
  - Primary & secondary vertex for b-jet tagging
  - Impact parameter and invariant mass for B-physics

HLT tracking in RoI
- Single leptons: e, µ, leptonic τ decays
- Final selections requiring ultimate efficiency & precision

Custom pixel clustering algorithm on FPGAs
Geometrically grouped data distributed to processing units

Precise fitting with good tracks being extrapolated to missing layers
Track fit with full-resolution info for track candidates

Transformation to coarse-resolution hits
Comparison to reference track patterns at coarse resolution
FTK System

- **High throughput (40M tracks/s)**
- **Low latency (100 µs)**
- Achieved by:
  - **Parallelism:** 64 independent towers (4 in $\eta$ x 16 in $\phi$)
  - **Hardware Implementation:** custom ASICs and FPGAs
  - **Two stages:**
    1. **Pattern matching with 8 detector layers**
       - Uses Associative Memory (AM): 1 billion patterns
       - Reduced granularity: Pixels/Strips grouped to super strips
    2. **Extension to 12 layers**
       - **Track parameters extracted using Principle Component Analysis** => Sum rather than fit
       - Uses full granularity

**Implementation:**
- 8192 Associative Memory (AM) chips
- each with 128k Patterns (8 words each)
- Total Memory 18 Mbits
- 2000 FPGAs

**AM chip:**
- Special Content-Addressable Memory chip
- AM06 implemented in 65 nm technology
- Large chip: $168mm^2$
- $10^{14}$ parallel 16-bit comparisons per second
FTK Patterns & Constants

- Determined from large sample of simulated single muons (~1.5 billion tracks)
- Pattern bank size determined by required:
  - minimum $p_T$ (1 GeV), $d_0$ & $z_0$ range
  - Super-strip size
- Larger super-strips => smaller pattern bank but more fakes => more work for later steps
- Variable resolution patterns:
  - Effectively multiply the Super Strip size by 2 (4 or 8) in a layer of a given pattern
  - Employs “Don’t Care” feature when bit-matching Super-Strip address

Maximizing efficiency:
- Allow 1 missing hit at each of stage 1 and stage 2
- Additional Wild Card option: treat layer as if all Super Strips had a hit

Variable Resolution Patterns:
Allow optimum combination of:
- large bins to limit bank size
- Small bins to minimise fake rate

doi:10.1109/ANIMMA.2011.6172856
FTK First Stage

- Clusters formed from SCT and Pixel input
- Transformed to coarse resolution Super Strips
- AM board performs Pattern Matching
  - Uses 3 Pixel + 5 SCT layers
    - (6+5 measurements)
- 1st Stage Track Fitting
  - Input: addresses of patterns with hits
  - Uses full-resolution hit information
  - Uses Principle Component Analysis to estimate $\chi^2$
  - Allows 1 missing hit
- 1st Stage Hit Warrior: Duplicate track removal
  - Compare tracks within same road
  - If tracks share >$N_{\text{max}}$ hits, keep higher quality track (based on $N_{\text{hits}}$ and $\chi^2$)

\[
\chi^2 = \sum_{i=1}^{6} \left( \sum_{j=1}^{11} S_{ij} x_j + h_i \right)^2
\]

$i$: components of $\chi^2$

$j$: measured coordinates

$S_{ij}$ & $h_i$ are pre-calculated constants

$x_j$ are hit coordinates
FTK Second Stage

- Use 8-layer track parameters to find 3 SCT stereo hits + IBL hit
- 12-layer track fit: parameters & $\chi^2$
- All track input to Hit Warrior to remove Duplicates
- Fit 1 track in 1 ns. (1 track every 5ps for full system)

$$\tilde{p}_i = \sum_{l=1}^{N} C_{il}x_l + q_i$$

$p_i$: track parameters
$C_{il}$, $q_i$: constants
$x_l$: hit coordinate

Data Formatter
Extrapolator
Track Fitter
Processor Unit
Auxiliary Board
Hit Warrior
FTK Status & Schedule

- **Status:**
  - Production of Chips & Boards is underway
  - First AM06 chips received – functioning well
  - Integration with Data Acquisition in progress
  - Dataflow to FTK system exercised at end of 2015 data-taking
  - Triggers being developed & tested with Simulated Data
    => expected performance measurements

- **Schedule:**
  - Full Barrel Commissioned Mid 2016
  - Full Coverage (for $<\mu> = 40$) : for start of 2017 data-taking
  - Full Spec. system (for $<\mu> = 69$): 2018 (luminosity driven)
GPU Prototype

- GPU provide high levels of parallelisation =>
- Re-writing of code required to exploit this parallelisation
- GPU prototype has been developed to explore potential for GPGPU in the ATLAS Trigger
- Goal is to assess the potential benefit in terms of throughput per unit cost.
- Integration with the Atlas software uses a client-server technology:

<table>
<thead>
<tr>
<th></th>
<th>Nvidia C2050</th>
<th>Nvidia K80</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cores</td>
<td>448</td>
<td>2x2496</td>
</tr>
<tr>
<td>Multiprocessors</td>
<td>14</td>
<td>2x13</td>
</tr>
<tr>
<td>Cores/SMX</td>
<td>32</td>
<td>192</td>
</tr>
</tbody>
</table>
Clustering on GPU

- Pixel Clustering implemented using a Cellular Automaton Algorithm
  - Each hit is assigned an initial tag
  - Tags are replaced for adjacent clusters
  - Ends when no more adjacent hits

- Obtain Factor 26 Speed-up running on GPU compared to 1 CPU core

D Emeliyanov and J Howard 2012 J. Phys.: Conf. Ser. 396 012018
Combinatorial Track-finding on GPU

- **Doublet formation:**
  - Combines hits from 2 layers to form doublets
  - 2D thread block; Uses local buffer

- **Triplet Making:**
  - Combines doublets with hits in 3rd layer
  - 2D thread block; Uses local buffer

- **Track Extension:**
  - Extends triplets to other layers
  - 1D thread block

- **Clone Removal:**
  - Identify tracks with different seeds sharing extension hits
  - Processes \((N \times (N-1))/2\) track pairs.
  - Each GPU thread processes a number of tracks.
GPU Performance & Outlook

Code Speed-up:
- Average factor 12 speedup for Tracking on GPU c.f. 1 CPU core

System Performance:
- Question: What increase in throughput comparing CPU system with CPU+GPU
- Measurements in progress:
  - Updated hardware: K80 GPU, Intel E5-2695V3 CPU
  - Updated software: Run-2 Tracking algorithm
- Initial measurements with Seed-Maker suggest factor of two increase in system throughput could be obtained by adding GPU.
- Work in progress to add GPU track following
- Prototype also includes Calorimeter & Muon reconstruction
Tracking in the Phase II Trigger

- \( L = 7.5 \times 10^{34} \text{ cm}^{-2} \text{s}^{-1} \), 200 pile-up events
  - \( \sim 10 \) times current highest luminosity
- Upgraded Detector including:
  - upgraded Trigger
  - replacement Tracker (ITK)
- Tracking in the Phase II Trigger:
  - Hardware Tracking in RoI at up to 1 MHz
    - Used to refine L0 decision
  - Hardware Full Event Tracking at \( \sim 100 \) kHz – selected events
    - Provides primary vertex information for pile-up suppression
  - Precision Tracking at the Event Filter - Mainly in RoI
    - Near offline quality tracks to maximize selection efficiency

Schematic of possible ITK layout

- Silicon Strips
- Pixels
- Extended coverage to \( |\eta| < 4 \)
The **Trigger** is a challenging environment for tracking
- High Rates, Short times, requirement for high efficiency and low fake rates

**Current solutions** include:
- Partial event reconstruction: use of Regions of Interest
- Limiting combinations: using beam constraint and RoI information
- Multi-step reconstruction and selection: Fast lower resolution, Slower full resolution
- Hardware-based Fast-Tracking based on Associative Memory and FPGA

**Future Evolution:**
- Evolution of software framework to exploit **concurrent execution** of algorithms
- Evolution of algorithms to exploit **internal parallelisation**
- Possible use of **accelerators** such as GPGPU – encouraging results obtained from prototype

**Phase-II is even more challenging**, will exploit:
- Hardware based tracking: Full-event and Regional
- CPU-based Precision Tracking in RoI (use of accelerators)

**Solutions used in the trigger could also be useful offline**
- Holy Grail: Full offline-quality tracking at L0 output rates (400 kHz – 1 MHz)
- Looking for new ideas to bring us closer to this ideal.
Additional Material
**The AM06 Chip**

AM06 is a **large** silicon chip in TSMC CMOS 65 nm technology:
- 168 mm²
- 421 M transistors
- 64 blocks, each of them containing 2 kbit of Associative Memory (AM), designed with full-custom approach and employing a new AM cell (XORAM) specifically optimized for FTK
- SERDES (serializer/deserializer) IP blocks from Silicon Creations, for serial I/O at 2 Gbit/s
- Standard cell ‘glue’ logic, JTAG interface, and BIST (CRC)

Status of testing:
- received first samples of AM06 (slow corner: -7.5% saturation current)
- find 42 out of 50 tested without production defects
- defects are localized errors at pattern level for few patterns
- tested intensively a few samples-they are running fine at full speed 100 MHz
The AM chip history

- 90’s Full custom VLSI chip – 0.7 mm AMS (INFN-Pisa) 128 patterns, 6x12 bit words each (F. Morsani et al., The AM chip: a Full-custom MOS VLSI Associative memory for Pattern Recognition, IEEE Trans. on Nucl. Sci., vol. 39, pp. 795-797 (1992).) 25 MHz clock


- 1999 first standard cell project presented at LHCC

- 2006 AMChip03 Standard Cell UMC 0.18 mm, 5k patterns in 100 mm² for CDF SVT upgrade total: AM patterns (L. Sartori, A. Annovi et al., A VLSI Processor for Fast Track Finding Based on Content Addressable Memories, IEEE TNS, Vol 53, Issue 4, Part 2, Aug. 2006). 50 MHz clock

- 2012 AMchip04 (Full custom/Std cell) TSMC 65 nm LP technology, 8k patterns in 14mm² Pattern density x12. First variable resolution implementation. (F. Alberti et al, 2013 JINST & C01040, doi:10.1088/1748-0221/8/01/C01040) 100 MHz

- 2013 AMchip05, 4k patterns in 12 mm² a further step towards final AMchip version. Serialized I/O buses at 2 Gbs, further power reduction approach. BGA 23x23 package.

- End 2015 AMchip06: 128k patterns in 180 mm². Final version of the AMchip for the ATLAS experiment.
**FTK = (very complex) custom parallel supercomputer**

- **FTK is a big system**
  - 8 full 9U VME crates
  - 5 ATCA shelves

- **Many different boards**
  - 2 types of ATCA boards
    - Data Formatter (DF), FTK-to-Level-2 Interface Crate (FLIC)
    - mezzanine for clustering (input mezzanine)
  - 9U Auxiliary board (AUX)
    - data organizer, track fitting and fake reduction functions
  - 2 types of 9U VME boards
    - Associative Memory Board (AMB), Second Stage Board (SSB)
    - 9U/4 mezzanine for Associative Memory chip (AMchip06)

~2000 FPGAs and ~8000 custom AM chips
FTK System and Boards

Input Mezzanine card (IM) + Data Formatter (DF)

Auxiliary card (AUX) + Associative Memory Board (AM)

Second Stage Board (SSB)

FTK to Level2 Interface Crate (FLIC)

100 kHz Event Rate

Raw Data ROBs

FTK ROBs

FLIC

DO AM DO AM
TF Proc. unit HW Proc. unit

45°+10° in φ
8 η-φ towers 2 PU/tower

Second Stage Fit (4 brds)

Data Format

Cluster Finding

Dual HOLA card

Pixels & SCT

RODs
**Example of complete L2 ID chain implemented on GPU (Dmitry Emeliyanov)**

<table>
<thead>
<tr>
<th></th>
<th>Time (ms)</th>
<th>Tau RoI 0.6x0.6</th>
<th>tt events $2 \times 10^{34}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C++ on 2.4 GHz CPU</td>
<td>CUDA on Tesla C2050</td>
<td>Speedup CPU/GPU</td>
</tr>
<tr>
<td>Data Prep.</td>
<td>27</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>Seeding</td>
<td>8.3</td>
<td>1.6</td>
<td>5</td>
</tr>
<tr>
<td>Seed ext.</td>
<td>156</td>
<td>7.8</td>
<td>20</td>
</tr>
<tr>
<td>Triplet merging</td>
<td>7.4</td>
<td>3.4</td>
<td>2</td>
</tr>
<tr>
<td>Clone removal</td>
<td>70</td>
<td>6.2</td>
<td>11</td>
</tr>
<tr>
<td>CPU GPU xfer</td>
<td>n/a</td>
<td>0.1</td>
<td>n/a</td>
</tr>
<tr>
<td>Total</td>
<td>268</td>
<td>22</td>
<td><strong>12</strong></td>
</tr>
</tbody>
</table>

Max. speed-up: x26
Overall speed-up $t(GPU)/t(CPU) = 12$

**Hardware:** GPU server at RAL –
- dual quad-core Westmere 2.4 GHz, Nvidia Tesla C2050
- GPU (448 cores @ 1.15 GHz, 3GB on-card memory, PCIe 2.0)

**Software:**
- Nvidia CUDA 4.0 for GPU programming
- SLC5 + Linux GPU driver from NVidia, AtlasHLT, 17.1.0, opt, 32
Expected Occupancies

- Occupancy studied at high expected pileup ($\mu=200$)
- High occupancies in layers facing interaction point and with long strip length
- Challenging to clustering and pattern recognition

```
ttbar event, $\mu=140$
```
Hardware Tracking at Phase II

**AM chip:**
- AM chip projected to be about 500 k patterns with a 28 nm ASIC using multi-layer masks (MLM).

**Regional Tracking:**
- Reconstruction in Regions of Interest at up to 1 MHz
- The estimated number of patterns required to cover the entire ITk up to |η| < 4.0, for track p_T > 4 GeV, is about 1.5x10^9. (c.f. 1 GeV for FTK)
- Duplicate patterns to allow 2 events to be processed in parallel.
- Total 3x10^9 patterns => 6,000 AM chips

**Full Event Tracking:**
- uses all ITk layers out to |η| < 4.0 and finds tracks down to p_T of 1 GeV at 100 kHz.
- ~ 10 Billion patterns => 20,000 AM chips