On-line Event Reconstruction in the CBM Experiment

V. Akishina$^{1,2,3,4}$, I. Kulakov$^3$, I. Kisel$^{1,2,3}$, I. Vassiliev$^3$ and M. Zyzak$^{3,1,2}$

for the CBM Collaboration

1 – Goethe-Universität Frankfurt, Frankfurt am Main, Germany
2 – Frankfurt Institute for Advanced Studies, Frankfurt am Main, Germany
3 – GSI Helmholtzzentrum für Schwerionenforschung GmbH, Darmstadt, Germany
4 – Joint Institute for Nuclear Research, Dubna, Russia

International Conference on Science and Technology for FAIR in Europe 2014
Worms, Germany
16.10.2014
The CBM Experiment

• **CBM** - future fixed-target heavy-ion experiment at FAIR, Darmstadt, Germany.
• $10^5$-$10^7$ collisions per second.
• Up to 1000 charged particles/collision.
• Free streaming data.
• No hardware triggers.
• **On-line event reconstruction and selection** is required in the first trigger level.

• **On-line** reconstruction on the 60000 CPU equivalent cores farm.
• High **speed** and **efficiency** of the reconstruction algorithms are required.
• The algorithms have to be highly **parallelised** and **scalable**.
• CBM event reconstruction: Kalman Filter and Cellular Automaton.

Simulated central Au-Au collision at 25 AGeV
The First Level Event Selection (FLES) package for the on-line reconstruction consist of:

- Cellular Automaton based track finder for track search in the Silicon Tracking System;
- Kalman filter based track fitter for track parameters reconstruction;
- event building based on the obtained set of tracks;
- KF Particle Finder for short-lived particles reconstruction and physics analysis;
- the module for a quality check.
Cellular Automaton Track Finder

0. Hits

1. Segments

2. Counters

3. Track Candidates

4. Tracks

Cellular Automaton:
1. Build short track segments.
2. Connect according to the track model, estimate a possible position on a track.
3. Tree structures appear, collect segments into track candidates.
4. Select the best track candidates.

Cellular Automaton:
• local w.r.t. data
• intrinsically parallel
• extremely simple
• very fast
Perfect for many-core CPU/GPU!

Track finder:
1. Kalman filter for track segments fit
2. The code is optimised with respect to both efficiency and time
3. The code is parallelised
   • Data level (SIMD instructions, 4 single-precision floating point calculations in parallel)
   • Task level (ITBB, parallelisation between cores)

Useful for complicated event topologies with large combinatorics and for parallel hardware
Significant multiple scattering of low-momentum tracks in the therefore their reconstruction efficiency is lower — 81.2%.

Decays of particles, e.g. in decays of tracks that is equal to 97.1%. The high-momentum secondary therefore, similar to the efficiency of high-momentum primary close to the collision point. Their reconstruction efficiency is, tum higher than 1 GeV/c originating from the region very charmonium, light vector mesons) are particles with moment-

The tests have been performed of tracks and ratios of clones (double found) and ghost (wrong) Au UrQMD collisions at 25 collisions at 25

Fig. 5. Efficiency of the track reconstruction for minimum bias Au-Au

Efficiency of the track reconstruction for minimum bias Au-

Cellular Automaton Track Finder: Efficiency

E -plane, YZ -plane,

The majority of signal tracks (decay products of D-mesons, by a factor 10000 with respect to the original scalar version algorithm down to 1 increased the processing speed of the SIMD KF track fit to different CPU/GPU architectures. All these changes have been used to keep flexibility of the algorithm with respect track fit quality as the standard fourth order Runge-Kutta expansion [6]. The analytic formula allows to obtain the same track propagation in the non-homogeneous magnetic field is done by a analytic formula, which is based on the Taylor treatment. The

R -plane, XZ -plane,

The residuals and the pulls for all track parameters are

Residuals of the track parameters are determined as a

The residuals and the pulls for all track parameters are

The CBM experiment is an experiment with a forward

Efficient and stable event reconstruction

<table>
<thead>
<tr>
<th>Efficiency, %</th>
<th>mbiass</th>
<th>central</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary high- ( p ) tracks</td>
<td>97.1</td>
<td>96.2</td>
</tr>
<tr>
<td>Primary low- ( p ) tracks</td>
<td>90.4</td>
<td>90.7</td>
</tr>
<tr>
<td>Secondary high- ( p ) tracks</td>
<td>81.2</td>
<td>81.4</td>
</tr>
<tr>
<td>Secondary low- ( p ) tracks</td>
<td>51.1</td>
<td>50.6</td>
</tr>
<tr>
<td>All tracks</td>
<td>88.5</td>
<td>88.3</td>
</tr>
<tr>
<td>Clone level</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>Ghost level</td>
<td>0.7</td>
<td>1.5</td>
</tr>
<tr>
<td>Reconstructed tracks/event</td>
<td>120</td>
<td>591</td>
</tr>
<tr>
<td>Time/event/core</td>
<td>8.2 ms</td>
<td>57 ms</td>
</tr>
</tbody>
</table>
4D Event Building

100 AuAu mbias events at 25 AGeV at $10^7$ Hz

- The SIS100/300 beam will have no bunch structure, but continuous.
- Free streaming data.
- Measurements in this case will be 4D ($x,y,z,t$).
- Reconstruction of time slices rather than events will be needed.

Reconstructed tracks clearly represent groups, which correspond to the original events.
4D Track Reconstruction with the CA Track Finder

<table>
<thead>
<tr>
<th>Algorithm Step</th>
<th>% of total execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialisation</td>
<td>8 %</td>
</tr>
<tr>
<td>Triplets construction</td>
<td>64 %</td>
</tr>
<tr>
<td>Tracks construction</td>
<td>15 %</td>
</tr>
<tr>
<td>Final stage</td>
<td>13 %</td>
</tr>
</tbody>
</table>

Parallelism inside a time-slice, 100 mbias events in a time-slice

- CA Track Finder
- Initialisation
- Triplets Construction
- Tracks Construction
- Final Stage

Intel E7-4860 @ 2.27 GHz with Hyper-Threading technology

Total time - 84 ms

Total time - 849 ms

4D track reconstruction is scalable with the speed-up factor of 10.1 (out of 13)
Track fit: Estimation of the track parameters at one or more hits along the track – Kalman Filter (KF)

KF Block-diagram

KF as a recursive least squares method

Kalman filter:
1. Start with an arbitrary initialisation.
2. Add one hit after another.
3. Improve the state vector.
4. Get the optimal parameters after the last hit.

Nowadays the Kalman Filter is used in almost all HEP experiments
Portability of the KF Track Fitter

- CPU-like approach.
- Steps due to the hyper-threading.

- GPU-like approach.
- Jumps because some of the streaming multiprocessors are loaded partially.

Full portability of the Kalman filter fitter on Intel/AMD CPUs, Intel Xeon Phi, Nvidia GPUs, AMD GPUs.
Concept of KF Particle

Concept:

- Mother and daughter particles have the same state vector and are treated in the same way
- Geometry independent
- Kalman filter based

Functionality of the package:

- Construction of the particles from tracks or another particles
- Decay chains reconstruction
- Transport of the particles
- Simple access to the particle parameters and their errors
- Calculation of the distance to point
- KF Particle Finder for short-lived particles reconstruction

State vector
Position, momentum and energy

\[ r = \{ x, y, z, p_x, p_y, p_z, E \} \]
KF Particle Finder for On-line Event Selection

Tracks: $e^\pm$, $\mu^\pm$, $\pi^\pm$, $K^\pm$, $p^\pm$
secondary and primary

Strange particles:
$K^0_s \rightarrow \pi^+ \pi^-$
$\Lambda \rightarrow p \pi^-$
$\bar{\Lambda} \rightarrow \pi^+ p^-$

Strange and multi-strange resonances:
$\Sigma^{*+} \rightarrow \Lambda \pi^+$
$\Sigma^{*-} \rightarrow \Lambda \pi^-$
$\Sigma^{*0} \rightarrow \Lambda \pi^0$
$K^{*-} \rightarrow K^0_s \pi^-$
$K^{*+} \rightarrow K^0_s \pi^+$
$\Xi^{*-} \rightarrow \Lambda K^-$
$\Xi^{*0} \rightarrow \Lambda K^0$
$\Xi^{*+} \rightarrow \bar{\Lambda} K^+$

Gamma-decays
$\pi^0 \rightarrow \gamma \gamma$
$\eta \rightarrow \gamma \gamma$
$\Sigma^0 \rightarrow \Lambda \gamma$
$\Sigma^0 \rightarrow \bar{\Lambda} \gamma$

Open-charm:
$D^0 \rightarrow \pi^+ K^-$
$D^0 \rightarrow \pi^+ \pi^+ \pi^- K^-$
$\bar{D}^0 \rightarrow \pi^- K^+$
$D^0 \rightarrow \pi^- \pi^+ K^+$
$D^+ \rightarrow \pi^+ \pi^+ K^-$
$D^- \rightarrow \pi^- \pi^+ K^+$
$D^+_s \rightarrow \pi^+ K^+ K^-$
$D^-_s \rightarrow \pi^- K^+ K^-$
$\Lambda_c \rightarrow \pi^+ K^- p$

Multi-strange resonances:
$\Xi^{-} \rightarrow \Lambda \pi^-$
$\Xi^{+} \rightarrow \Lambda \pi^+$
$\Omega^{-} \rightarrow \Lambda K^-$
$\Omega^{+} \rightarrow \bar{\Lambda} K^+$

Light vector mesons:
$\rho \rightarrow e^- e^+$
$\rho \rightarrow \mu^- \mu^+$
$\omega \rightarrow e^- e^+$
$\omega \rightarrow \mu^- \mu^+$
$\phi \rightarrow e^- e^+$
$\phi \rightarrow \mu^- \mu^+$
$\phi \rightarrow K^- K^+$

Strange resonances
$K^{*0} \rightarrow K^+ \pi^-$
$\bar{K}^{*0} \rightarrow \pi^+ K^-$
$\Lambda^* \rightarrow p K^-$
$\bar{\Lambda}^* \rightarrow p^* K^+$

Charmonium:
$J/\Psi \rightarrow e^- e^+$
$J/\Psi \rightarrow \mu^- \mu^+$

Multi-strange hyperons:
$\Xi^{-} \rightarrow \Lambda \pi^-$
$\Xi^{+} \rightarrow \Lambda \pi^+$
$\Omega^{-} \rightarrow \Lambda K^-$
$\Omega^{+} \rightarrow \bar{\Lambda} K^+$

Gamma-decays
$\pi^0 \rightarrow \gamma \gamma$
$\eta \rightarrow \gamma \gamma$
$\Sigma^0 \rightarrow \Lambda \gamma$
$\Sigma^0 \rightarrow \bar{\Lambda} \gamma$
Search for Strange Particles, AuAu, SIS 100

- 5M central AuAu UrQMD events at 10 AGeV with realistic ToF PID with 80 ps resolution.
- All particles - UrQMD output.
- Event by event 3D analysis.
Scalability of KF Particle Finder on Many-core System

- The KF Particle Finder has been vectorised and parallelised.
- The KF Particle Finder shows linear scalability on many-core machines (the scalability on a computer with 40 physical, 80 logical cores is shown).

The speed of the package:
- mbias AuAu collisions at 25 AGeV – 1.5 ms/event/core
- central AuAu collisions at 25 AGeV – 10.5 ms/event/core
The first version of the FLES package is
- vectorised and parallelised;
- is portable;
- shows scalable behaviour on nodes with different types and number of CPUs.

The packages currently starts with STS space points. Hits reconstruction is to be added.

Towards CBM FLES farm

Available prototype clusters:
- LOEWE CSC (FIAS, Frankfurt)
- Mini Cube (GSI, Darmstadt)
- FAIR-Russia HPC Cluster (ITEP, Moscow)
Summary and Plans

Summary

• The first version of the FLES package for the on-line reconstruction includes stages for:
  - track reconstruction and fitting;
  - event building;
  - short-lived particles reconstruction and physics selection.

• All methods demonstrate high reconstruction quality.

• The package for on-line reconstruction shows high speed and strong scalability including HEP many-core clusters.

Plans for FLES

• Implement 4D physics selection with KF Particle Finder.

• Add global tracks reconstruction.

• Add reconstruction in PID detectors.

• Add reconstruction of hits.

• Port the whole FLES to graphic cards.