FLES – First Level Event Selection Package for the CBM Experiment

Valentina Akishina[1,2], *Ivan Kisel*[1,2,3], *Igor Kulakov*[1,3], *Maksym Zyzak*[1,3]

[1]Goethe University, Grueneburgplatz 1, 60323 Frankfurt, Germany
[2]FIAS Frankfurt Institute for Advanced Studies, Ruth-Moufang-Str. 1, 60438 Frankfurt, Germany
[3]GSI Helmholtz Center for Heavy Ion Research, Planckstr. 1, 64291 Darmstadt, Germany

DOI: http://dx.doi.org/10.3204/DESY-PROC-2014-05/4

The CBM (Compressed Baryonic Matter) experiment is a future heavy-ion experiment at the future Facility for Anti-Proton and Ion Research (FAIR, Darmstadt, Germany). First Level Event Selection (FLES) in the CBM experiment will be performed on-line on a dedicated processor farm. An overview of the on-line FLES processor farm concept, different levels of parallel data processing down to the vectorization, implementation of the algorithms in single precision, memory optimization, scalability with respect to number of cores, efficiency, precision and speed of the FLES algorithms are presented and discussed.

1 Introduction

The CBM (Compressed Baryonic Matter) experiment [1] is an experiment being prepared to operate at the future Facility for Anti-Proton and Ion Research (FAIR, Darmstadt, Germany). Its main focus is the measurement of very rare probes, that requires interaction rates of up to 10 MHz. Together with the high multiplicity of charged particles produced in heavy-ion collisions, this leads to huge data rates of up to 1 TB/s. Most trigger signatures are complex (short-lived particles, e.g. open charm decays) and require information from several detector sub-systems.

The First Level Event Selection (FLES) package [2] of the CBM experiment is intended to reconstruct the full event topology including tracks of charged particles and short-lived particles. The FLES package consists of several modules: track finder, track fitter, particle finder and physics selection. As an input the FLES package receives a simplified geometry of the tracking detectors and the hits, which are created by the charged particles crossing the detectors. Tracks of the charged particles are reconstructed by the Cellular Automaton (CA) track finder [3] using to the registered hits. The Kalman filter (KF) based track fit [4] is used for precise estimation of the track parameters. The short-lived particles, which decay before the tracking detectors, can be reconstructed via their decay products only. The KF Particle Finder, which is based on the KFParticle package [2], is used in order to find and reconstruct the parameters of short-lived particles by combining already found tracks of the long-lived charged particles. The KF Particle Finder also selects particle-candidates from a large number of random combinations. In addition, a module for quality assurance is implemented, that allows to control the quality of the reconstruction at all stages. It produces an output in a simple ASCII format, that can be interpreted later as efficiencies and histograms using the ROOT framework. The FLES package is platform and operating system independent.

The FLES package in the CBM experiment will be performed on-line on a dedicated many-core CPU/GPU cluster. The FLES algorithms have to be therefore intrinsically local and parallel and thus require a fundamental redesign of the traditional approaches to event data processing in order to use the full potential of modern and future many-core CPU/GPU architectures. Massive hardware parallelization has to be adequately reflected in mathematical and computational optimization of the algorithms.

2 Kalman Filter (KF) track fit library

Searching for rare interesting physics events, most of modern high energy physics experiments have to work under conditions of still growing input rates and regularly increasing track multiplicities and densities. High precision of the track parameters and their covariance matrices is a prerequisite for finding rare signal events among hundreds of thousands of background events. Such high precision is usually obtained by using the estimation algorithms based on the Kalman filter (KF) method. In our particular case, the KF method is a linear recursive method for finding the optimum estimation of the track parameters, grouped as components into the so-called state vector, and their covariance matrix according to the detector measurements.

The Kalman filter based library for track fitting includes following tracking algorithms: track fit based on the conventional Kalman filter; track fit based on the square root Kalman filter; track fit based on the Kalman filter with UD factorization of the covariance matrix; track smoother based on the listed above approaches and deterministic annealing filter based on the listed above track smoothers.

High speed of the reconstruction algorithms on modern many-core computer architectures can be accomplished by: optimizing with respect to the computer memory, in particular declaring all variables in single precision, vectorizing in order to use the SIMD (Single Instruction, Multiple Data) instruction set and parallelizing between cores within a compute node.

Several formulations of the Kalman filter method, such as the square root KF and the UD KF, increase its numerical stability in single precision. All algorithms, therefore, can be used either in double or in single precision.

The vectorization and parallelization of the algorithms are done by using of: header files, Vc vector classes, Intel TBB, OpenMP, Intel ArBB and OpenCL.

The KF library has been developed and tested within the simulation and reconstruction framework of the CBM experiment, where precision and speed of the reconstruction algorithms are extremely important.

When running on CPU the scalability with respect to the number of cores is one of the most important parameters of the algorithm. Figure 1 shows the scalability of the vectorized KF algorithm. The strong linear behavior shows, that with further increase of the number of cores on newer CPUs the performance of the algorithm will not degrade and the maximum speed will be reached. The stair-like dependence appears because of the Intel Hyper-Threading technology, which allows to run two threads per core and gives about 30% of performance advantage. The scalability on the Intel Xeon Phi coprocessor is similar to CPU with four threads per core running simultaneously.

In case of the graphic cards the set of tasks is divided into working groups and distributed among compute units (or streaming multiprocessors) by OpenCL and the load of each compute

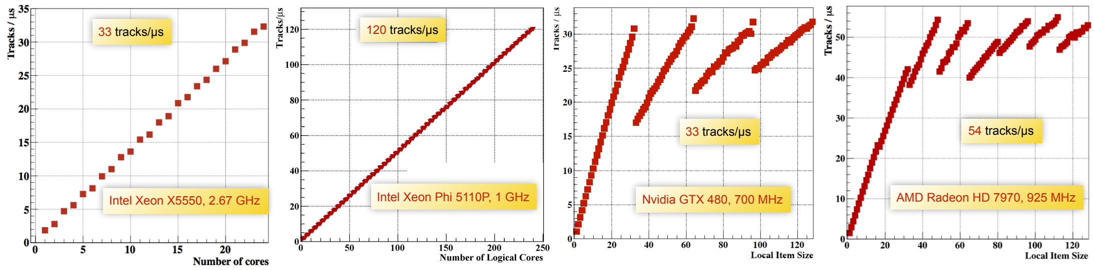

Figure 1: Portability of the Kalman filter track fit library on different many-core CPU/Phi/GPU architectures.

unit is of the particular importance. Each working group is assigned to one compute unit and should scale within it with respect to the number of tasks in the group. Figure 1 shows that the algorithm scales linearly on the graphic cards up to the number of cores in one compute unit (for Nvidia GTX480 — 32, for AMD Radeon HD 7970 — 16). Then the drop appears, because when first 32 (for Nvidia) or 16 (for AMD) tasks are processed, only one task is left and all other cores of the compute unit are idle. Increasing the number of tasks in the group further the speed reaches the maximum with the number of tasks dividable by the number of cores in the compute unit. Due to the overhead in tasks distribution the maximum performance is reached when the number of tasks in the group is two-three times more than the number of cores.

3 Cellular Automaton (CA) track finder

Every track finder must handle a very specific and complicated combinatorial optimization process, grouping together one- or two-dimensional measurements into five-dimensional tracks.

In the Cellular Automaton (CA) method first (1) short track segments, so-called cells, are created. After that the method does not work with the hits any more but instead with the created track segments. It puts neighbor relations between the segments according to the track model here and then (2) one estimates for each segment its possible position on a track, introducing in such a way position counters for all segments. After this process a set of tree connections of possible track candidates appears. Then one starts with the segments with the largest position counters (3) and follows the continuous connection tree of neighbors to collect the track segments into track candidates. In the last step (4) one sorts the track candidates according to their length and χ^2-values and then selects among them the best tracks.

The majority of signal tracks (decay products of D-mesons, charmonium, light vector mesons) are particles with momentum higher than 1 GeV/c originating from the region very close to the collision point. Their reconstruction efficiency is, therefore, similar to the efficiency of high-momentum primary tracks that is equal to 97.1%. The high-momentum secondary particles, e.g. in decays of K_s^0 and Λ particles and cascade decays of Ξ and Ω, are created far from the primary vertex, therefore their reconstruction efficiency is lower – 81.2%. Significant multiple scattering of low-momentum tracks in the material of the detector system and large curvature of their trajectories lead to lower reconstruction efficiencies of 90.4% for primary tracks and of 51.1% for secondary low-momentum tracks. The total efficiency for all tracks is

88.5% with a large fraction of low-momentum secondary tracks. The levels of clones (double found tracks) and of ghost (wrong) tracks are 0.2% and 0.7% respectively. The reconstruction efficiency of the CA track finder is stable with respect to the track multiplicity.

The high track finding efficiency and the track fit quality are crucial, especially for reconstruction of the short-lived particles, which are of the particular interest for the CBM experiment. The reconstruction efficiency of short-lived particles depends quadratically on the daughter track reconstruction efficiency in case of two-particle decays. The situation becomes more sensitive for decays with three daughters and for decay chains. The level of a combinatorial background for short-lived particles depends strongly on the track fit quality. The correct estimation of the errors on the track parameters improves distinguishing between the signal and the background particle candidates, and thus to suppress the background. The ghost (wrong) tracks usually have large errors on the track parameters and therefore are easily combined with other tracks into short-lived particle candidates, thus a low level of ghost tracks is also important to keep the combinatorial background low. As a result, the high track reconstruction efficiency and the low level of the combinatorial background improve significantly the event reconstruction and selection by the FLES package.

4 4-Dimensional time-based event building

Since resolving different events is a non-trivial task in the CBM experiment, the standard reconstruction routine will include an event building, the process of defining exact borders of events within a time-slice and grouping tracks into even-corresponding clusters, which they originate from. For this task an efficient time-based tracking is essential. Since the CA track finder is proved to be fast and stable with respect to the track multiplicity, the next step towards the time-slice based reconstruction would be the implementation of time measurements.

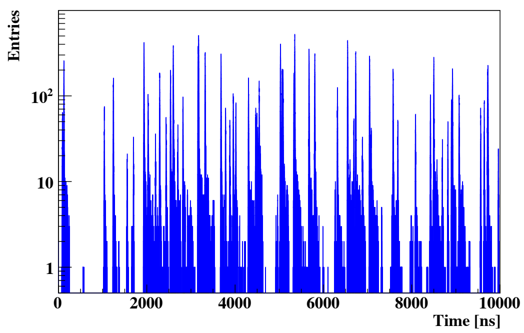

Figure 2: Part of a time-slice with 100 minimum bias events. With blue color the distribution of hit time measurements in a time-slice is shown.

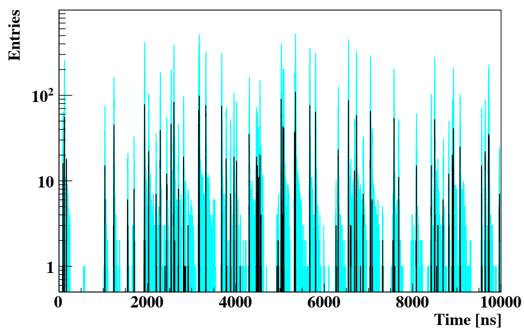

Figure 3: Part of a time-slice with 100 minimum bias events. With light blue color the initial distribution of hit measurements is reproduced, black color shows time measurements of reconstructed tracks.

In order to introduce a time measurement into the reconstruction procedure to each minimum bias event in a 100 events group an event start time was assigned during simulation phase. The start time was obtained with the Poisson distribution, assuming the interaction rate of 10^7 Hz. A time stamp we assign to a certain hit consists of this event start time plus the time shift due to the time of flight, which is different for all hits. In order to obtain a time measurement for a hit we then smear a time stamp according to a Gaussian distribution with a sigma value of the detector resolution of 5 ns.

After introducing the time measurement we can use the time information in the CA track finder. We do not allow to build triplets out of hits, which time difference is greater than 3σ of the detector time resolution. It is a very good approximation, since the time of flight between the detector planes is negligible in comparison to the detection precision. Apart from that, we perform the reconstruction procedure in a regular way. After the reconstruction we assign to each track a time measurement, which is calculated as an average of its hits measurements.

The initial distribution of hits measurements representing the complexity of defining event borders in a time-slice at interaction rate of 10^7 Hz is shown in Fig. 2 with blue color. The resulting distribution of reconstructed track measurements (black color), as well as the distribution of initial hit measurements (light blue color), one can see in Fig. 3. The reconstructed tracks clearly represent groups, which correspond to events, which they originate from.

5 KF Particle Finder – a package for reconstruction of short-lived particles

Today the most interesting physics is hidden in the properties of short-lived particles, which are not registered, but can be reconstructed only from their decay products. A fast and efficient KF Particle Finder package, based on the Kalman filter (hence KF) method, for reconstruction and selection of short-lived particles is developed to solve this task. A search of more than 50 decay channels has been currently implemented.

In the package all registered particle trajectories are divided into groups of secondary and primary tracks for further processing. Primary tracks are those, which are produced directly in the collision point. Tracks from decays of resonances (strange, multi-strange and charmed resonances, light vector mesons, charmonium) are also considered as primaries, since they are produced directly at the point of the primary collision. Secondary tracks are produced by the short-lived particles, which decay not in the point of the primary collision and can be clearly separated. These particles include strange particles (K_s^0 and Λ), multi-strange hyperons (Ξ and Ω) and charmed particles (D_0, $D^{\pm}$, $D_s^{\pm}$ and Λ_c). The package estimates the particle parameters, such as decay point, momentum, energy, mass, decay length and lifetime, together with their errors. The package has a rich functionality, including particle transport, calculation of a distance to a point or another particle, calculation of a deviation from a point or another particle, constraints on mass, decay length and production point. All particles produced in the collision are reconstructed at once, that makes the algorithm local with respect to the data and therefore extremely fast.

KF Particle Finder shows a high efficiency of particle reconstruction. For example, for the CBM experiment efficiencies of about 15% for Λ and 5% for Ξ^- with AuAu collisions at 35 AGeV are achieved together with high signal-to-background ratios (1.3 and 5.9 respectively).

6 A standalone FLES package for the CBM experiment

The First Level Event Selection (FLES) package of the CBM experiment is intended to reconstruct on-line the full event topology including tracks of charged particles and short-lived particles. The FLES package consists of several modules: CA track finder, KF track fitter, KF Particle Finder and physics selection. In addition, a quality check module is implemented, that allows to monitor and control the reconstruction process at all stages. The FLES package is platform and operating system independent.

The FLES package is portable to different many-core CPU architectures. The package is vectorized using SIMD instructions and parallelized between CPU cores. All algorithms are optimized with respect to the memory usage and the speed.

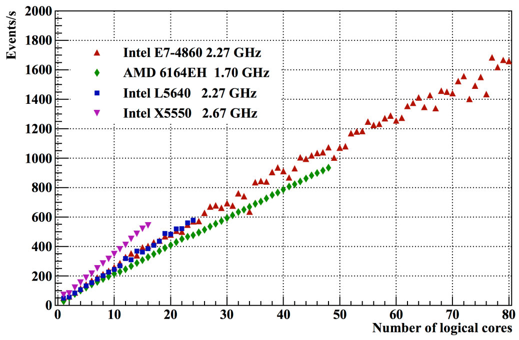

Figure 4: Scalability of the FLES package on many-core servers with 16, 24, 48 and 80 logical cores.

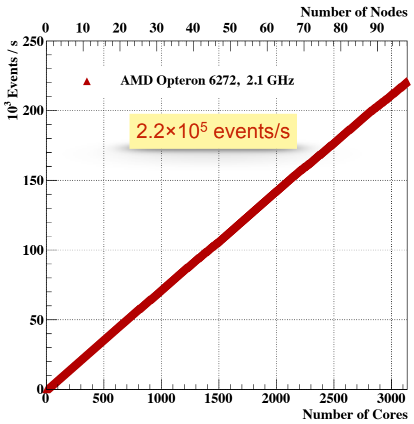

Figure 5: Scalability of the FLES package on 3 200 cores of the FAIR-Russia HPC cluster (ITEP, Moscow).

Four servers with Intel Xeon X5550, L5640 and E7-4860 processors and with AMD 6164EH processor have been used for the scalability tests. The AMD server has 4 processors with 12 physical cores each, in total 48 cores. All Intel processors have the hyper-threading technology, therefore each physical core has two logical cores. The Intel X5550 and L5640 servers with two CPUs each have in total 16 and 24 logical cores respectively. The most powerful Intel server has 4 processors E7-4860 with 10 physical cores each, that gives 80 logical cores in total.

The FLES package has been parallelized with ITBB (Intel Threading Building Blocks) implementing the event-level parallelism by executing one thread per one logical core. Reconstruction of 1000 minimum bias Au-Au UrQMD (Ultrarelativistic Quantum Molecular Dynamics model) events at 25 AGeV has been processed per each thread. In order to minimize the effect of the operating system each thread is fixed to a certain core using the pthread functionality provided by the C++ standard library. Fig. 4 shows a strong scalability for all many-core systems achieving the reconstruction speed of 1700 events per second on the 80-cores server.

The FLES package in the CBM experiment will be performed for the on-line selection and the off-line analysis on a dedicated many-core CPU/GPU farm. The farm is currently estimated to have a compute power equivalent to 60 000 modern CPU cores. Fig. 5 shows the scalability of the FLES package on a many-core computer farm with 3 200 cores of the FAIR-Russia HPC cluster (ITEP, Moscow).

7 Summary

The standalone FLES package has been developed for the CBM experiment. It contains track finding, track fitting, short-lived particles finding and physics selection. The Cellular Automaton and the Kalman filter algorithms are used for finding and fitting tracks, that allows to achieve a high track reconstruction efficiency. The event-based CA track finder was adapted for the time-based reconstruction, which is a requirement in the CBM experiment for the event building. The 4D CA track finder allows to resolve hits from different events overlapping in time into event-corresponding clusters of tracks. Reconstruction of about 50 decay channels of short-lived particles is currently implemented in the KF Particle Finder. The package shows a high reconstruction efficiency with an optimal signal to background ratio.

The FLES package is portable to different many-core CPU architectures. The package is vectorized and parallelized. All algorithms are optimized with respect to the memory usage and the processing speed. The FLES package shows a strong scalability on the many-core CPU systems and the processing and selection speed of 1700 events per second on a server with 80 Intel cores. On a computer cluster with 3 200 AMD cores it processes up to $2.2 \cdot 10^5$ events per second.

References

[1] The CBM Collaboration, *Compressed Baryonic Matter Experiment*, Tech. Stat. Rep., GSI, Darmstadt, 2005; 2006 update

[2] I. Kisel, I. Kulakov and M. Zyzak, IEEE Trans. Nucl. Sci. **60**, 3703–3708 (2013)

[3] I. Kisel, Nucl. Instr. and Meth. **A566** 85–88 (2006)

[4] S. Gorbunov, U. Kebschull, I. Kisel, V. Lindenstruth, and W.F.J. Müller, Comp. Phys. Comm., **178**, 374–383 (2008)