Journal of Physics: Conference Series

PAPER • OPEN ACCESS

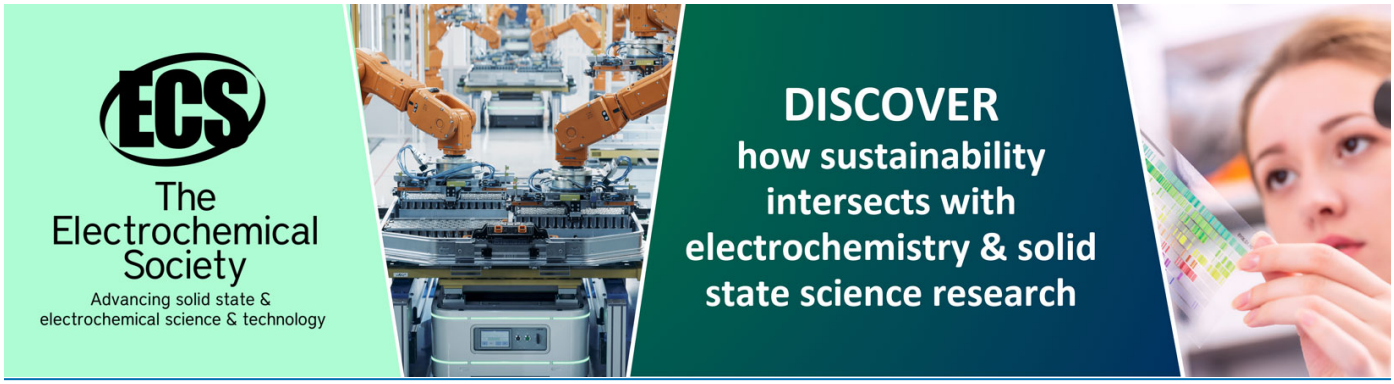
Time-based Cellular Automaton track finder for the CBM experiment

To cite this article: Valentina Akishina and Ivan Kisel 2015 *J. Phys.: Conf. Ser.* **599** 012024

View the article online for updates and enhancements.

You may also like

- CBM First-level Event Selector Input Interface Demonstrator
Dirk Hutter, Jan de Cuveland and Volker Lindenstruth
- Review on Higgs hidden-dark sector physics
Theodota Lagouri
- Explore the lifetime frontier with MATHUSLA
C. Alpigiani



This content was downloaded from IP address 3.129.19.251 on 04/05/2024 at 12:41

Journal of Physics: Conference Series **599** (2015) 012024 doi:10.1088/1742-6596/599/1/012024

Time-based Cellular Automaton track finder for the CBM experiment

Valentina Akishina[1,2,3,4] **and Ivan Kisel**[1,2,3]

[1]Goethe University, Frankfurt am Main 60325, Germany
[2]Frankfurt Institute for Advanced Studies, Frankfurt am Main, 60438, Germany
[3]GSI Helmholtzzentrum für Schwerionenforschung, Planckstraße 1, 64291 Darmstadt, Germany
[4]Joint Institute for Nuclear Research, Joliot-Curie 6, 141980 Dubna, Russia

E-mail: v.akishina@gsi.de

Abstract. The future heavy-ion experiment CBM (FAIR/GSI, Darmstadt, Germany) will focus on the measurement of rare probes at interaction rates up to 10 MHz with data flow of up to 1 TB/s. The beam will provide free stream of particles without bunch structure. That requires full online event reconstruction and selection not only in space, but also in time, so-called 4D event building and selection. This is a task of the First-Level Event Selection (FLES) package. The FLES reconstruction and selection package consists of several modules: track finding, track fitting, short-lived particles finding, event building and event selection. The input data are distributed within the FLES farm in a form of so-called time-slices, in which time length is proportional to a compute power of a processing node. A time-slice is reconstructed in parallel between cores within a CPU, thus minimising communication between CPUs. After all tracks of the whole time-slice are found and fitted, they are collected into clusters of tracks originated from common primary vertices. After that short-lived particles are found and the full event building process is finished.

1. Introduction

The CBM experiment [1] will address fundamental questions of strong interaction physics. The experiment will investigate the properties of highly compressed baryonic matter as it is produced in relativistic nucleus-nucleus collisions. The CBM detector is designed to measure both bulk observables with large acceptance and rare diagnostic probes such as charmed particles and vector mesons decaying into lepton pairs.

Since the CBM physics program includes measurement of rare observables, the experiment is required to cope with extremely high interaction rates in order to obtain sufficient statistics. Namely, CBM is being designed to operate with collision rates up to 10 MHz, which was never a case in heavy-ion experiments before. Taking into account that rare probes are to be measured in a heavy-ion collision environment (up to 1000 charged particles per collision) one should expect a data flow rate of 1 TB/s. Such a huge data rate makes it mandatory to select interesting events online with a reduction factor of three orders of magnitude in order to meet a recordable data rate of 1 GB/s. In addition to such high input collision rate and complicated event topology, the full event reconstruction for the event selection is to be done online at the First Level Event Selection (FLES) stage, due to absence of simple hardware triggers. This poses challenges for the CBM computing and requires the algorithms to be both efficient and fast.


Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

On top of that the FAIR accelerator beam will be a continuous stream of particles. As a result data from different collisions may overlap in time, making traditional event-by-event analysis not applicable in CBM. Instead of the event-based approach, a so called time-slice-based reconstruction will be implemented. Such a time-slice, containing not only space coordinates, but also time information from a number of possibly overlapping in time events, will be provided as an input to the reconstruction algorithm to perform a so-called four-dimensional (4D) event building and selection.

This problem is to be solved on a dedicated many-core CPU/GPU computer farm by the FLES package [2]. The package is being developed as a platform and operating system independent package, which includes several modules of the selection chain: track finding, track fitting, short-lived particles finding, event building and event selection. It has to be fast, precise and able to fully utilise potential of modern many-core architectures in order to be suitable for online data processing. The Cellular Automaton (CA) track finder [3] is used to reconstruct tracks of charged particles within a time-slice.

2. Event-based Cellular Automaton track finder

Finding trajectories of charged particles in a typical dense environment of heavy ion collisions is often considered as the most challenging and time-consuming stage of the reconstruction procedure. The reason for that usually is a very specific problem of dealing with combinatorial enumeration, which grows fast with track multiplicity, while grouping together detector measurements into tracks in the presence of noise. The CA track finder proposes a solid solution for the combinatorial search optimisation. Namely, this method benefits from drastic suppression of combinatorial enumeration by introducing a phase of building up short track segments at an early stage before going into the main search. In addition to that, the method is intrinsically local with respect to data processing and, thus, can be run in parallel on modern many-core CPU/GPU computer architectures. The CA method features made the algorithm an appropriate solution for the track reconstruction in the main tracking detector Silicon Tracking System (STS).

In the case of the CBM experiment short track segments at the first stage are triplets: all possible three measurement combinations on a neighboring STS stations are checked to be physical taking into account the multiple scattering. The hit measurement information is locally processed and stored in a new consolidation extent format, namely triplets, with no need to consider individual hits again later in the algorithm. The optimization is also achieved due to saving computational and memory access efforts by combining tasks. For example, at the stage of triplets creation the track finder also checks on a fly the neighboring relation between the triplets in order to remember potential neighbors and estimate the position of the triplet in the track. These links are stored together with triplets and used later at the next stage of connecting triplets in order to build track candidates out of them. The neighboring information allows to get a tree of candidates just by following the neighboring relation. Moving from the first triplet to the next neighboring triplet, the track finder obtains the full tree of possible tracks. In the last stage a competition between the track candidates takes place: only the longest tracks with best χ^2-value sharing no hits in common are to survive.

The task of finding tracks in CBM is split in several iterations to make the reconstruction fast and reliable in the case of a high track density: at the first iteration track finder searches for high-momentum primary tracks only, at the second one — for low-momentum primary tracks, and at the last one — for secondary tracks as well. After each iteration all hits from reconstructed tracks are marked as used and removed from further consideration, thus significantly reducing the combinatorial enumeration.

For the performance purposes we define the tracking efficiency as a ratio of reconstructed and reconstructable tracks. A track is considered as reconstructable if it has at least 4 consecutive

Monte-Carlo points. By definition, a reconstructed track is assigned to a generated particle, if at least 70% of its hits have been produced by the particle. A generated particle is regarded as found, if it has been assigned to at least one reconstructed track. If a particle is found more than once, all additionally reconstructed tracks are regarded as clones. A track is called ghost, if it is not assigned to any generated particle according to the 70% criterion.

The track reconstruction efficiency for different sets of tracks and the level of clones and ghost tracks for the case of event-based analysis are shown in the first (3D) column of Table 1. The majority of tracks of particular physics interest are particles with momentum higher than 1 GeV/c originating from the region very close to the collision point (e.g. decay products of D-mesons, charmonium, light vector mesons). The efficiency for this group is 96.1%. Since the high-momentum secondary particles (e.g. from decays of K^0_s and Λ and cascade decays of Ξ and Ω) are created far from the primary vertex, their reconstruction is more complex and the efficiency is 76.6%. A significant multiple scattering influence on low-momentum tracks in the material of the detector system and larger curvature in the magnetic field make it a more complicated task to reconstruct them. Thus, their reconstruction efficiency is 79.8%. The levels of clones and ghost tracks are 0.4% and 0.2% respectively.

3. Track finding in the case of extreme track multiplicities

As a special study of the CA track finder stability the algorithm behavior was investigated with respect to the track multiplicity. For the study a super-event, which includes a number of minimum bias events, was reconstructed with no time information taken into account. To create such a super-event we combine space coordinates of hits from a number Au+Au minimum bias events at 25AGeV ignoring such information as event number or time. The super-event is given to the CA track finder as an input and reconstructed as a regular event with no changes in the reconstruction procedure. It is important to mention that such an approach does not precisely correspond to a pile-up simulation, since in this situation fake hits in STS are created from strips on the event level, not within the whole super-event.

Varying the number of minimum bias events in a super-event we have studied the track reconstruction efficiency with respect to the track multiplicity. The efficiency dependence is stable (see Figure 1). In particular, the efficiency of the algorithm decreases by 4% only for the extreme case of 100 minimum bias events in the super-event (see (3+1)D column of Table 1), comparing to the case of single minimum bias events (see 3D column of Table 1). The efficiencies for the reference tracks ($p > 1$ GeV/c) remains high for all track multiplicities range. The efficiencies for extra (100 MeV/c $< p <$ 1 GeV/c) and secondary tracks are also stable. The level of ghost tracks is always beyond 10% and changes slowly.

Summarising, the CA track finder reconstruction algorithm shows high stability with respect to the track multiplicities up to the extreme case of more than 10 000 reconstructed tracks.

4. In-event level parallelism of CA track finder

The newest high-performance architectures tend to have more CPUs, number of physical and logical cores per processor and longer vector registers. This tendency requires reconstruction algorithms to scale with number of cores and vector width. A vectorised sequential version of the CA track finder was taken as a starting point for developing a parallel one. Each step of the algorithm was parallelilized inside a super-event with OpenMP [4] and Pthreads [5] interfaces.

The sequential implementation of algorithms needed to undergo certain changes in order to be run in parallel. Some parts of the code, being essentially sequential, had to be significantly changed. Due to the fact that on modern architectures calculating time is negligible in comparison with data access time, memory optimisation and data structures scheme are essential for parallel programming. One should keep frequently used data in the fast cache memory and thus reduce the time for memory access, since the main memory is slower by about two orders

Efficiency, %	3D	(3+1)D	4D
All tracks	83.8	80.4	83
Primary high-p	96.1	94.3	92.8
Primary low-p	79.8	76.2	83.1
Secondary high-p	76.6	65.1	73.2
Secondary low-p	40.9	34.9	36.8
Clone level	0.4	2.5	1.7
Ghost level	0.1	8.2	0.3
Time/event/core	8.2 ms	31.5 ms	8.5 ms

Table 1: Track reconstruction performance for 3D event-by-event analysis, super-event (3+1)D and time-based 4D reconstruction for 100 minimum bias Au+Au collisions at 25 AGeV.

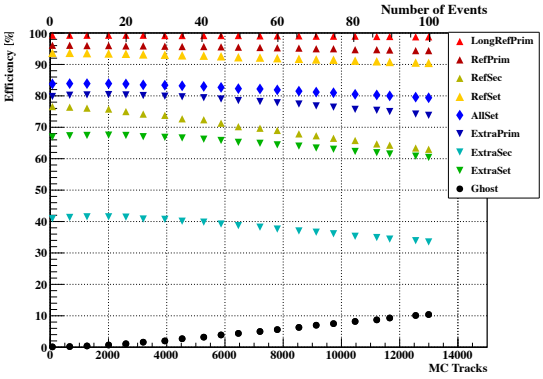


Figure 1: Track reconstruction efficiencies and ghost rate for different sets of tracks versus track multiplicity.

of magnitude than the low-level cache. A lot of optimisation efforts are required for a certain computing architecture in order to benefit from the layered structure of the CPUs caches.

The optimisation and testing of the parallel CA track finder were performed on a server with 4 Intel Xeon E7-4860 processors with 10 physical cores each. The processor supports the Intel Hyper-Threading technology: architecturally, every core consists of two logical ones. Unlike the multiple processor systems with independent processor units, in case of the Hyper-Threading the logical cores share some resources, like caches and the execution engine. As a result a logical core doesn't give 100% of a performance gain, but an estimated gain of 30% instead.

As it was mentioned above, the algorithm consists of several parts. First, a short (2% of the execution time) initialisation takes place, when we prepare the hit information for tracking. The main and the most time consuming part of triplet construction takes about 90% of the sequential execution time. Out of triplets we construct tracks, that takes about 4%, and when we prepare information for the next iteration (3.4%). All steps of the algorithm were parallelised inside a super-event using different sources of parallelism in each step: hits in the initialisation and the final stages, triplets for the major part, track candidates for the track construction step. In order to have enough sources of parallelism to fill a whole CPU, a super-event of 100 minimum bias events was processed. The resulting speed-up factors for different steps as well as for the full algorithm within one CPU (20 hyper-threaded logical cores) are presented in Figure 2.

The algorithm shows a linear scalability. Due to hyper-threading theoretically one can expect the maximum speed-up factor of about 13 on such a CPU. The achieved speed-up factor is 10.6 for the CA track finder reconstruction algorithm on a 10 hyper-threaded physical cores CPU [6].

5. Reconstruction of time-slices

Since resolving different events is a non-trivial task in CBM, an efficient time-based track finder is essential in order to define exact borders of events within a time-slice and grouping tracks into event-corresponding clusters. In order to include time measurement into the algorithm an event start time was assigned to each minimum bias event in a 100 events group during the simulation phase. The start time was simulated with the Poisson distribution, assuming the interaction rate of 10^7 Hz. A time stamp, which we assign to a certain hit, was simulated by this event start time plus a time shift due to the time of flight from the collision point to a detector plane. In order to obtain time measurement we smear a hit time stamp according to the Gaussian distribution with the σ value of the detector resolution 5 ns.

The time measurement information is to be used in the tracking. At the triplet building

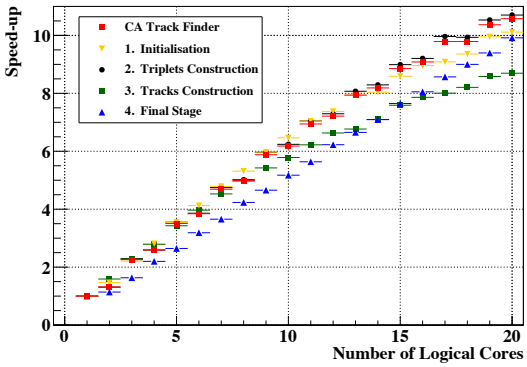


Figure 2: Speed-up factor due to parallelisation for different steps and the full algorithm on Intel Xeon E7-4860 CPU with 10 physical cores and hyper-threading.

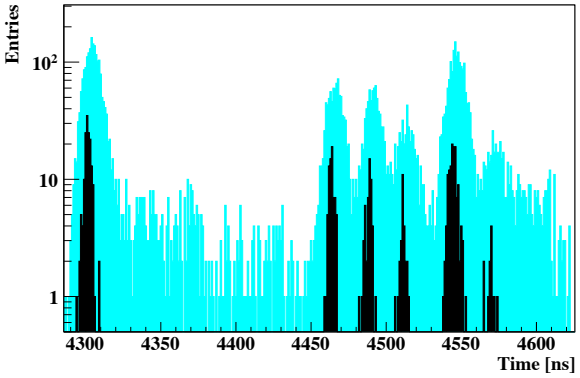


Figure 3: Distribution of time measurement in a part of a time-slice: hit time measurement (light blue), track time (black).

stage triplets are only build out of hits with the same time measurement within 3σ of the detector resolution. It is a justified assumption, since the time of flight between detector planes is negligible in comparison to the detector precision. The same rule is valid in the search for neighboring triplets. Apart from that, the reconstruction is performed in a regular way.

Including the time information in the tracking has resulted in a higher efficiency of the CA track finder (see 4D column in Table 1). In particular the time information drastically decreased ghost and made the reconstruction 3.7 times faster than without the time information ((3+1)D column of Table 1). The speed now is 8.5 ms and comparable with the event-based analysis.

The initial distribution of hits measurements representing the complexity of defining event borders in an overlapping region of a time-slice at the interaction rate of 10^7 Hz is shown Figure 3 with blue color. The resulting distribution of reconstructed track measurements is shown in black. Reconstructed tracks clearly represent event-corresponding groups.

6. Conclusions

The standalone FLES package for the CBM experiment contains all reconstruction stages: track finding, track fitting, short-lived particles finding, event building and event selection. The Cellular Automaton track finder is used to reconstruct tracks of charged particles in a time-slice. The algorithm is parallelized between cores with a speed-up factor of 10.6 on a CPU with 10 physical cores with hyper-threading. Since resolving different events is a non-trivial task in the CBM experiment, the FLES package includes an event building, the process of defining exact borders of events within a time-slice. Thus, the CA track finder is suitable for the task of event building for the CBM experiment.

References

[1] B. Friman, C. Hohne, J. Knoll, S. Leupold, J. Randrup, R. Rapp and P. Senger, Lect. Notes Phys. **814**, 1 (2011)
[2] I. Kisel, I. Kulakov and M. Zyzak, Standalone First Level Event Selection package for the CBM experiment, IEEE Trans. Nucl. Sci. (2013), vol. 60, no. 5, 3703-3708
[3] I. Kisel, "Event reconstruction in the CBM experiment," *Nucl. Instr. and Meth.*, vol. A566, pp. 85–88, 2006
[4] OpenMP API specification for parallel programming, `http://openmp.org/wp`
[5] Pthreads, IEEE. Information Technology-Portable Operating System Interface (POSIX)-Part 1: System Application: Program Interface (API) [C Language]. IEEE/ANSI Std 1003.1, 1996 Edition
[6] I. Kisel, V. Akishina *Online 4-Dimensional Event Building in the CBM Experiment*, RT-2014 - 19th Real-Time Conference (Nara, Japan, May 2014)