6D TRACKING WITH COMPUTE UNIFIED DEVICE ARCHITECTURE (CUDA) TECHNOLOGY

S. Glukhov, E. Levichev, S. Nikitin, P. Piminov, D. Shatilov, S. Sinyatkin
BINP, Novosibirsk 630090, Russia

Abstract

A program code TrackKing for a 6D fully-coupled particle tracking in circular accelerators has been developed with the usage of parallel computations on Graphics Processing Units (GPU) with Compute Unified Device Architecture (CUDA). We can track several thousands of particles in parallel providing optical functions calculation, dynamic aperture and energy acceptance study, intrabeam scattering, radiation effects (including their realistic distribution along the lattice), etc. In the paper we discuss the code arrangement and compare its performance with PTC module of MAD-X.

INTRODUCTION

Various tools for beam tracking allow computing trajectories of beam particles in the given lattice. Coordinate transformation from initial to final coordinates is generally the same for each particle. Thus, simulation algorithm can be paralleled using SIMD principle (Single Instruction Multiple Data). This approach has been successfully implemented in GPU. There are several software-hardware complexes for GPGPU (General Purpose Computing on Graphics Processing Units), one of them is CUDA by NVIDIA company [1].

TrackKing is the program complex for particle tracking in circular accelerators, which has been developed with the usage of CUDA technology. Two main goals were set: on the one hand, tracking should be as fast as possible, on the other hand, the program code should be comprehensive and extendable. To achieve this, a clear division of labor between CPU (Central Processing Unit) and GPU has been implemented: all the preliminary work on lattice and results processing are performed on CPU, while GPU is assigned only to the tracking, because this part of the work can be most effectively paralleled. Also two-stage compilation is used: source code of CUDA kernel (central part of tracking code) is generated on CPU during runtime (first stage), then passed to nvcc compiler (developed by NVIDIA) and then the resulting code is executed on GPU (second stage).

Such an approach allows one to perform a task with lots of tracking involved (e.g. dynamic aperture study) on a single PC with CUDA enabled graphics card in a reasonable period of time. Usually such tasks are performed on CPU clusters at data processing centers, but the ability to perform calculations locally is very important when remote clusters are unavailable or overburdened.

PROGRAM LOGIC

Both CPU and GPU parts of TrackKing are implemented on C++ language, an extension to C++ provided by CUDA Toolkit [1] is also used in the GPU part. The logic of the program is shown in (Fig. 1), where parts of TrackKing code are denoted by gray rectangles.

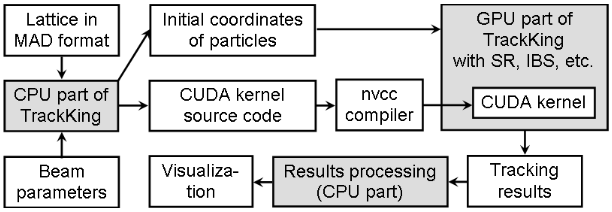

Figure 1: Scheme of TrackKing simulation program.

Reading Input Files

TrackKing accepts beamlines and sequences saved with SAVELINE or SAVE commands in MAD-8 [2] and MAD-X [3] formats. Non-saved beamlines and sequences described within a single file are also accepted. In this case all the mathematical expressions will be correctly parsed, but action commands (MATCH, CORRECT, SURVEY, etc.) will be ignored. After reading of the structure TrackKing saves parameters of beamline elements using its own class library.

Calculating Optical Functions and Structure Integrals

All necessary optical functions and structure integrals are calculated inside the CPU part of the program. TrackKing uses Twiss parameterization for non-coupled motion; for transversally coupled motion Teng — Edwards [4] or Lebedev — Bogasz [5] parameterizations can be used. TrackKing also has its own parameterization for fully coupled motion, which is a 6D extension of 4D Lebedev — Bogasz parameterization.

Generating Initial Coordinates of Particles and Source Code of CUDA Kernel

Usually the computational power is insufficient to track every particle in a beam, but one can track a sample (small ensemble of particles with the same distribution). The data obtained on the previous step can be used to inject matched beam (sample) with equilibrium emittance and energy spread, but also a beam with any other initial distribution can be injected.

Generated source code of CUDA kernel is highly optimized in view of the features of the lattice and the effects taken into account in the forthcoming tracking run. To improve tracking performance, separate functions are generated not only for each type of beamline elements (except DRIFT), but for each different element of the same type. For huge lattices with thousands of elements the CUDA kernel can

ISBN 978-3-95450-136-6

be splitted into several kernels to reduce compilation time and the amount of virtual memory used by nvcc compiler during the compilation.

Compiling and Performing Tracking

Now neither CPU nor GPU part of TrackKing has user interface, because the program is still under development. So, each tracking run means recompilation of the both parts with manual start of the compilers. But in the future we plan to convert the CPU part into an executable module with console interface, which will automatically start nvcc compiler after parsing the lattice and then launch the tracking on the GPU part. Other constituents of the GPU part, except the CUDA kernel (which depends on the lattice), will be precompiled to speed up the compilation process.

Visualization of the Results

6D phase space trajectories of the tracked particles are recorded into file at one or more observation points along the lattice. Observation points can be arranged automatically or manually, each of them has adjustable resolution (turns between two successive records) and selectivity (subset of particles, whose coordinates are recorded). This allows one to study beam distribution, its Poincaré sections, moments and related values (beam sizes, emittances, etc.) depending on time and coordinate along the lattice. Tools for Fourier analysis of the trajectories, dynamic aperture and energy acceptance study and analysis of particle losses are also implemented in TrackKing. In the case of insufficient disk space substantial part of data processing can be performed on-the-flight, when only distribution moments and other averaged values are recorded to disk instead of the full trajectories.

Results are exported as simple text files that can be processed with any data analyzing and plotting tool.

PHYSICAL EFFECTS TAKEN INTO ACCOUNT

Magnetic and Electric Fields of Beamline Elements

For each type of beamline elements TrackKing uses the same symplectic map as PTC module of MAD-X. Identity of the maps was carefully checked for various lattices containing all types of beamline elements possible in MAD [6]. The maps are obtained from expanded hamiltonian, all linear elements are thick, nonlinear elements can be splitted into any desired number of slices using drift-kick-drift scheme. Multipole fringe fields can be simulated in the hard-edge approximation [7].

Synchrotron Radiation

Two methods of SR simulation are implemented in TrackKing. On the one hand, radiation damping and quantum exitation due to SR can be simulated once per turn [8]. On the other hand, these effects can be taken into account in each beamline element, where beam particles follow curved trajectories. In high energy rings distributed radiation energy losses result in variation of equilibrium energy along the lattice (sawtooth effect) and closed orbit distortion. To preserve closed orbit magnet tapering is introduced (variation of magnetic field strength in beamline elements along the ring in proportion to the equilibrium energy). In TrackKing tapering can be introduced automatically. SR from quadrupoles and higher multipoles, as well as from dipoles, can affect beam dynamics. The techniques used in TrackKing for SR simulations allow taking these effects into account [9].

IBS and Touschek Effect

Simulations of IBS and Touschek effect in TrackKing are based on the theory described in [10]. Now these effects can be simulated once per turn as well as in each beamline element (previous version of algorithm was described in [11]). The new algorithm has shown the same results for various lattices and beam energies as the previous one.

BENCHMARKING

Tracking performance of TrackKing and PTC module of MAD-X were measured for various lattices and different numbers of tracked particles (N_p), results are summarized in Table 1 and Fig. 2. Initial conditions and coordinate transformations were exactly the same for each particle, all stochastic effects (SR, IBS) were switched off. Measurements were performed on a standard PC with the following characteristics:

- OS: Microsoft Windows XP SP3,
- CPU: Pentium Dual Core E5300 (2.60 GHz),
- GPU: NVIDIA GeForce GT 520,
- RAM: 3.25 GB.

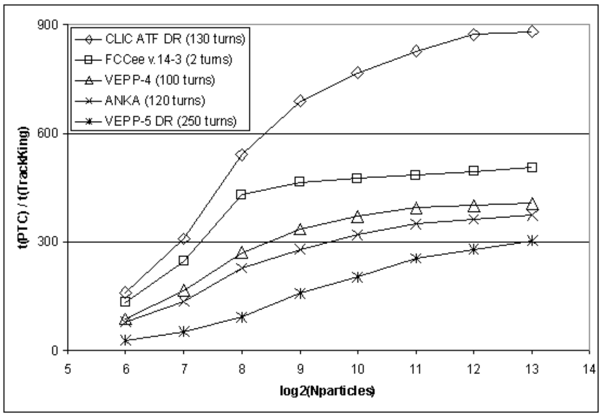

Figure 2: Tracking performance ratio of PTC and TrackKing for different lattices.

So, for sufficiently large number of tracked particles (>2000) TrackKing shows 200—900 times better performance than PTC, depending on lattice. This advantage is due to paralleling of the tracking process. But even if we assume that PTC tracking code can be fully paralleled on CPU (i.e. total PTC tracking time can be decreased N_p times), then for some lattices TrackKing still has an advantage for

Table 1: Time Elapsed for Tracking (in Seconds) with TrackKing (Upper) and PTC (Lower)

N_p	ATF	FCC	V4	ANKA	V5
64	0.25	0.16	0.22	0.22	0.24
	40	21	19	17	6.5
128	0.26	0.17	0.23	0.23	0.25
	80	42	38	31	13
256	0.3	0.2	0.28	0.27	0.28
	162	86	76	61	26
512	0.48	0.39	0.47	0.44	0.34
	331	181	157	123	53
1024	0.84	0.73	0.84	0.77	0.52
	644	347	311	246	105
2048	1.55	1.44	1.58	1.42	0.84
	1283	696	623	495	214
4096	2.98	2.83	3.05	2.73	1.53
	2605	1397	1220	992	427
8192	5.84	5.61	5.98	5.34	2.86
	5148	2829	2430	1989	868

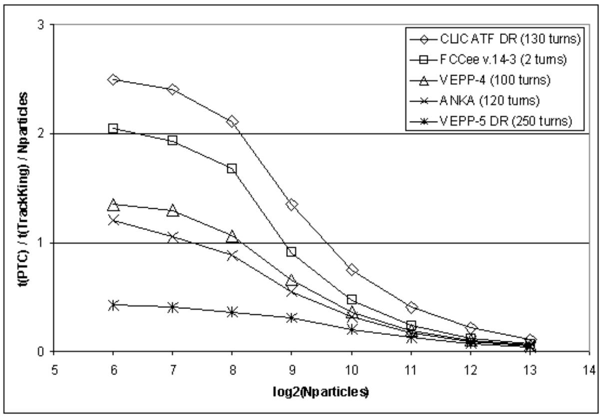

Figure 3: Tracking performance ratio of PTC and TrackKing per particle for different lattices.

$N_p < 10^9$ (Fig. 3). If a special GPGPU device (such as Tesla or Kepler graphics card by NVIDIA) is available for benchmarking, then the variety of lattices and range of N_p to see this advantage will increase substantially. This means that single particle tracking is on average faster in TrackKing than in PTC due to highly optimized CUDA kernel code.

FUTURE PLANS

Improvements to be done in the near future are the following: implementation of advanced tapering options, adding a module for dynamic aperture and energy acceptance optimization by sextupole adjustment, taking misalignments and field errors into account. Also the code for beam-beam and beamstrahlung simulations by D. Shatilov (BINP) is to be added to the program. The work on documentation and user interface part is being carried out in parallel.

CONCLUSION

New tracking tool based on CUDA technology has been developed, which is substantially faster than PTC code used in MAD-X. On the available hardware performance gain is up to 2.5 times per particle due to automatically generated highly optimized CUDA kernel code and up to 900 times totally due to paralleling. On the one hand, the new program named TrackKing has a big potential as an independent tracking tool. On the other hand, techniques used in TrackKing can be integrated into MAD-X to increase tracking performance considerably.

REFERENCES

[1] A. Boreskov, A. Kharlamov, Basics of working with CUDA technology. — DMK Press, 2010.

[2] H. Grote, F. C. Iselin, "The MAD program (methodical accelerator design) version 8.4: User's reference manual", CERN-SL-90-13-AP-REV.2, CERN-SL-90-13-AP (1991).

[3] H. Grote et al., "The MAD-X program (methodical accelerator design) version 5.02.06: User's reference manual", `http://madx.web.cern.ch/madx/madX/doc/latexuguide/madxuguide.pdf`

[4] D. A. Edwards, L. C. Teng, "Parametrization of linear coupled motion in periodic systems", IEEE Trans. Nucl. Sci. **20** (1973) 885.

[5] S. A. Bogacz, V. A. Lebedev, "Betatron motion with coupling of horizontal and vertical degrees of freedom", eConf C **010630** (2001) T511.

[6] R. De. Maria, M. Fjellstrom, "SixTrack physics manual (draft)", August 2015, `http://sixtrack.web.cern.ch/SixTrack/doc/physics_manual/sixphys.pdf`

[7] E. Forest, J. Milutinovic, "Leading Order Hard Edge Fringe Fields Effects Exact (1 + Delta) and Consistent With Maxwell's Equations for Rectilinear Magnets", Nucl. Instrum. Meth. A **269** (1988) 474.

[8] K. Ohmi, K. Hirata and K. Oide, "From the beam envelope matrix to synchrotron radiation integrals", Phys. Rev. E **49**, 751 (1994).

[9] S. Glukhov, E. Levichev, S. Nikitin, P. Piminov, D. Shatilov, S. Sinyatkin, "6D Tracking with Compute Unified Device Architecture (CUDA) Technology", these Proceedings.

[10] J. Le Duff, "Single And Multiple Touschek Effects", In *Berlin 1987, Proceedings, Advanced accelerator physics* 114-130 and Orsay Lin. Accel. Lab. - L.A.L.-RT-88-08 (88,rec.Sep.) 17 p.

[11] S. Glukhov, E. Levichev, S. Nikitin, P. Piminov, D. Shatilov, S. Sinyatkin, "IBS Simulations with Compute Unified Device Architecture (CUDA) Technology", Proceedings of IPAC'14.