FAST PARALLEL TRACKING ALGORITHM FOR THE MUON DETECTOR OF THE CBM EXPERIMENT AT FAIR

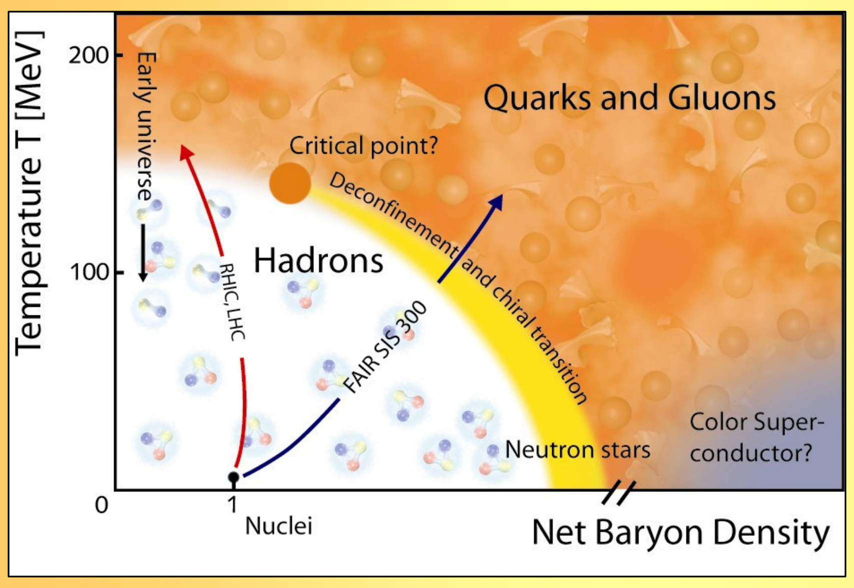
A.Lebedev[1,2], C.Höhne[1], I.Kisel[1] and G.Ososkov[2] for the CBM collaboration

[1]GSI Helmholtzzentrum für Schwerionenforschung GmbH, Darmstadt, Germany

[2] Laboratory of Information Technologies, Joint Institute for Nuclear Research, Dubna, Russia

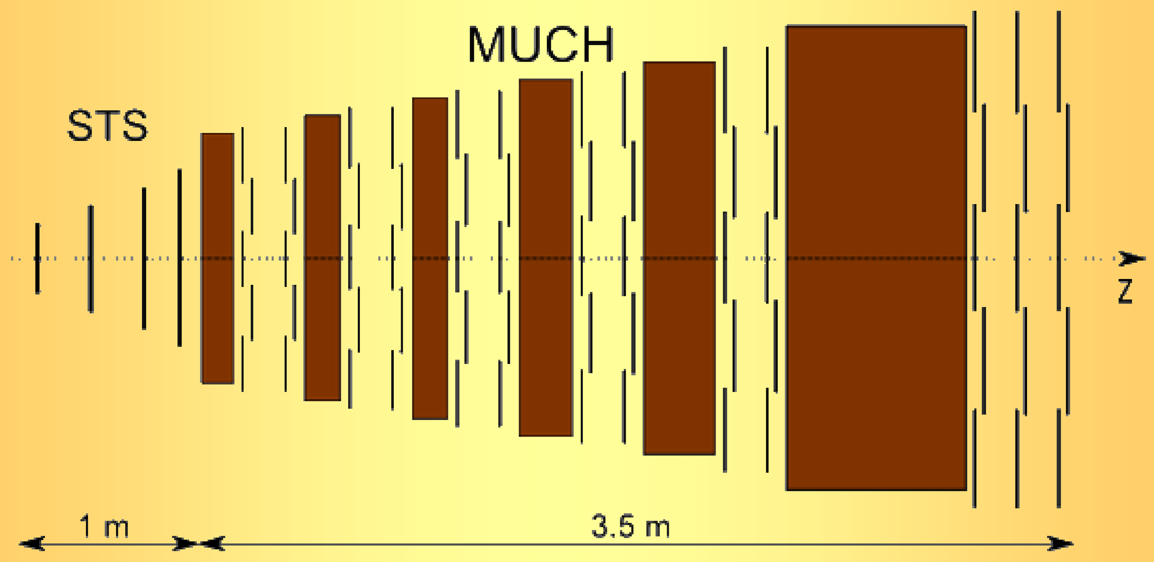
Particle trajectory recognition is an important and challenging task in the Compressed Baryonic Matter (CBM) experiment at the future FAIR accelerator at Darmstadt. The tracking algorithms have to process terabytes of input data produced in particle collisions. Therefore, the speed of the tracking software is extremely important for data analysis. In this contribution, a fast parallel track reconstruction algorithm which uses available features of modern processors is presented. These features comprize a SIMD instruction set and multithreading. The first allows one to pack several data items into one register and to operate on all of them in parallel thus achieving more operations per cycle. The second feature enables the routines to exploit all available CPU cores and hardware threads. This parallelized version of the tracking algorithm has been compared to the initial serial scalar version which uses a similar approach for tracking. A speed up factor of 140 was achieved (from 630 ms/event to 4.5 ms/event) for an Intel Core 2 Duo processor at 2.26 GHz.

Physics topics



Exploration of the QCD phase diagram in regions of high baryon density and moderate temperature

The CBM detector setup for muon measurement



- Fixed target experiment heavy-ion experiment.
- Muon identification by absorber technique:
  →absorber-detector sandwich for continues tracking.
- Up to 1000 charged particles per reaction.
- 10^7 event per second.

Fast tracking algorithms are essential!

Track reconstruction

- Initial seeds are tracks reconstructed in the STS with fast celular automaton algorithm
- Two main steps in track reconstruction:
  - Tracking finding:
    - Track following,
    - Kalman filter,
    - Validation gate,
    - Nearest neighbor hit-to-track assignment,
  - Global track selection:
    - Track quality criterion,
    - Check of the shared hits.

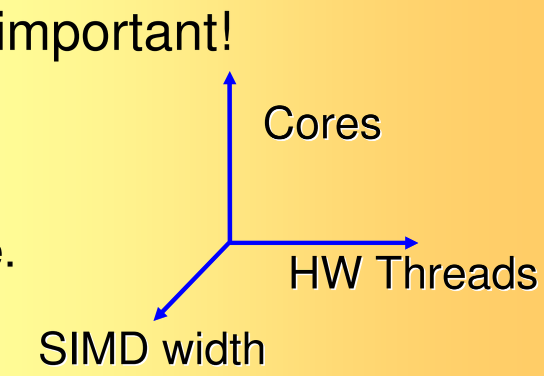
Track propagation

Important for muon detectors → large material budget

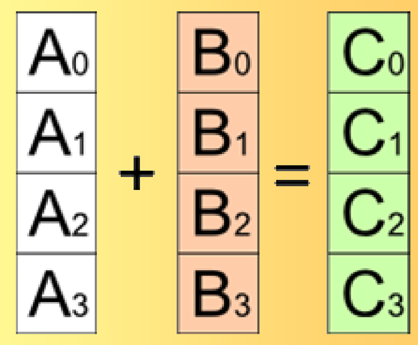
- Track extrapolation:
  - Straight line in case of absence of magnetic field.
  - Solution of the equation of motion in an inhomogeneous magnetic field with the 4th order Runge-Kutta method, with a parallel integration of the derivatives.
- Material effects:
  - **Energy loss** (ionization: Bethe-Bloch, bremsstrahlung: Bethe-Heitler, pair production),
  - **Multiple scattering** (Gaussian approximation).
- Simplified geometry and navigation.

Parallelism: SIMD and multithreading

- We are already at the age of parallel computing!
- So parallel programming is very important!
  - New way of thinking.
  - Identification of parallelism.
  - Design of parallel algorithm.
  - Implementation can be a challenge.



- SIMD – Single Instruction Multiple Data
  - CPU's have it!
  - **Today**: SSE - 128 bit registers
    - 4 x 32-bit floats
  - **Tomorrow**: AVX, LRB
    - AVX: 8 x 32-bit floats
    - LRB: 16 x 32-bit floats
  - Benefits:
    - X time more operations per cycle,
    - X time more memory throughoutput.



4 concurrent add operations

- Multithreading:
  - Many core era coming soon…
  - Tool for CPUs
    - Threading Building Blocks

Speed up of the algorithm

- Minimize access to global memory, use cache:
  - Approximation of the magnetic field with polynoms.
- Simplification of the detector geometry:
  - Optimized geometry navigation.
- Computational optimization of the Kalman Filter

SIMDization of the Kalman filter track fit procedure

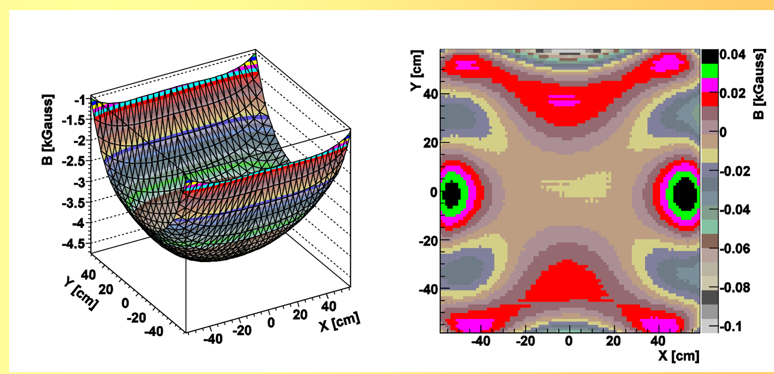
↓

Implement SIMDized KF into the track finder

↓

Implement multithreading into the track finder

- Optimization of the combinatorics:
  - Fast search of hits.



Magnetic field approximation (left) and difference between the approximation and field map (right)

Results

	Initial version	Parallel version
Track reconstruction efficiency [%]	93.7	92.5
Time [ms/event]	630	4.5
Speedup factor	140	

*Tested on Intel Core 2 Duo processor at 2.26 GHz

Plans

- Continue investigation of the SIMD and multithreading for CPUs.
- Investigate new languages (OpenCl, CUDA) for the performance critical cores of the algorithms.
- Exploit dedicated purpose processors like GPUs.
- Be ready for the emerging heterogeneous many-core systems.

Conclusion

Next generation experiments as the CBM detector planned at FAIR aim at data recording rates not experienced so far. In order to perform an efficient analysis, fast tracking algorithms are essential. For CBM a parallel tracking algorithm for the muon detector was developed based on track seeds from the track reconstruction in the silicon tracking system located inside a dipol magnet. The algorithm uses two features of modern CPUs: a SIMD instruction set and multithreading. The comparison between the parallel tracking algorithm and the initial serial scalar one shows that on account of a minor loss in track reconstruction efficiency (from 93.7 % to 92.5 %) the parallel tracking algorithm is 140 times faster than the initial one (from 630 ms/event to 4.5 ms/event) for an Intel Core 2 Duo processor at 2.26 GHz. The use of multithreading and SIMD will be much more obvious in the future with the next generation of many core processors.

References

Lebedev A. et al. Track Reconstruction and Muon Identification in the Muon Detector of the CBM Experiment at FAIR // PoS. V. ACAT08, No 068. 2008.

Lebedev A. et al. Fast parallel tracking algorithm for the muon detector of the CBM experiment at FAIR // To be published in Letters to PEPAN