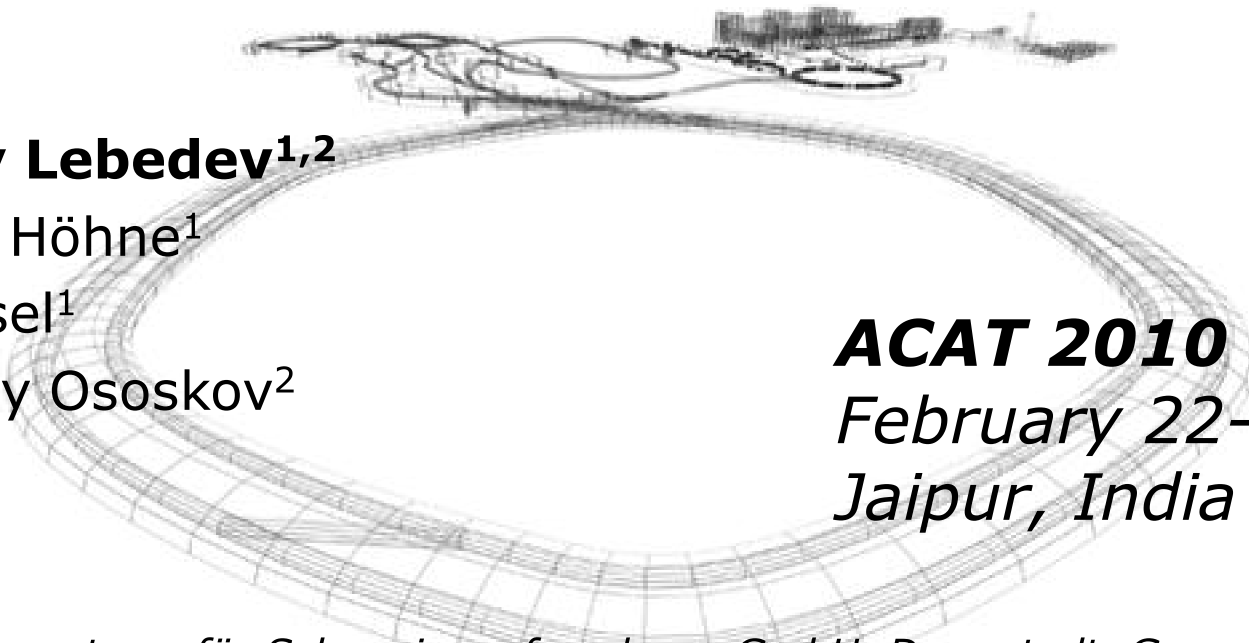
Fast parallel tracking algorithm for the muon detector of the CBM experiment at FAIR



Andrey Lebedev[1,2]

Claudia Höhne[1]

Ivan Kisel[1]

Gennady Ososkov[2]

ACAT 2010

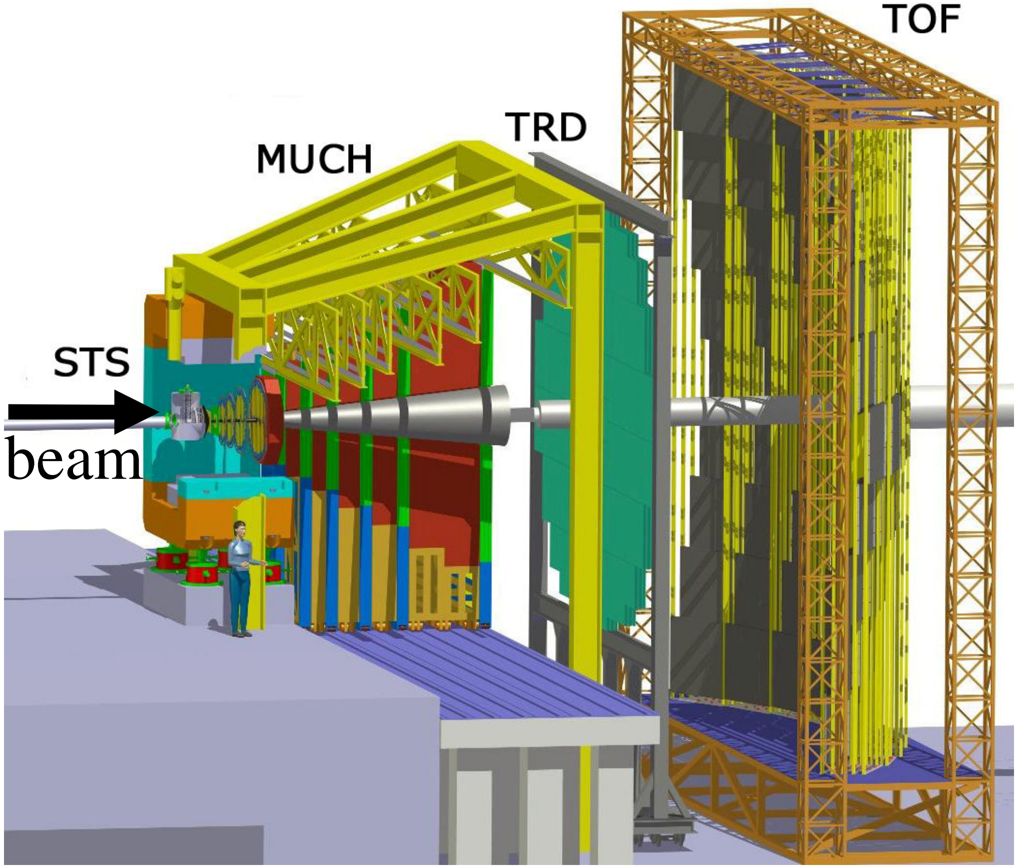
February 22-27, 2009

Jaipur, India

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

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

The CBM experiment at FAIR



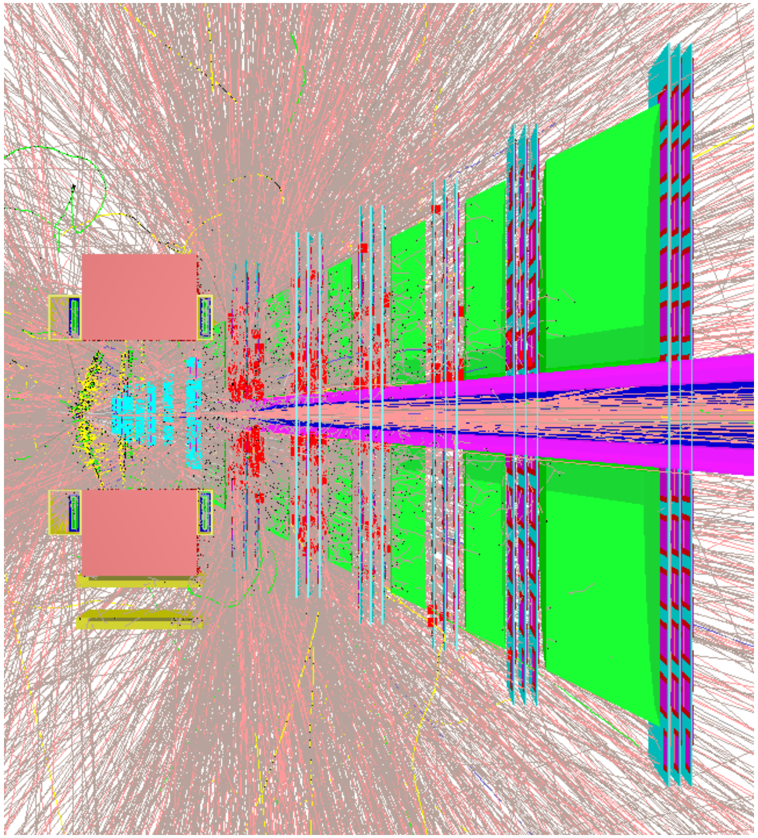
STS	track, vertex and momentum reconstruction
MUCH	muon identification
TRD	in case of muons only tracking
TOF	time of flight measurement

- exploration of the QCD phase diagram in regions of high baryon densities and moderate temperatures.
- comprehensive measurement of hadron and lepton production in pp, pA and AA collisions for 8-45 AGeV beam energy
- fixed target experiment

Challenges for tracking

Peculiarities for CBM:

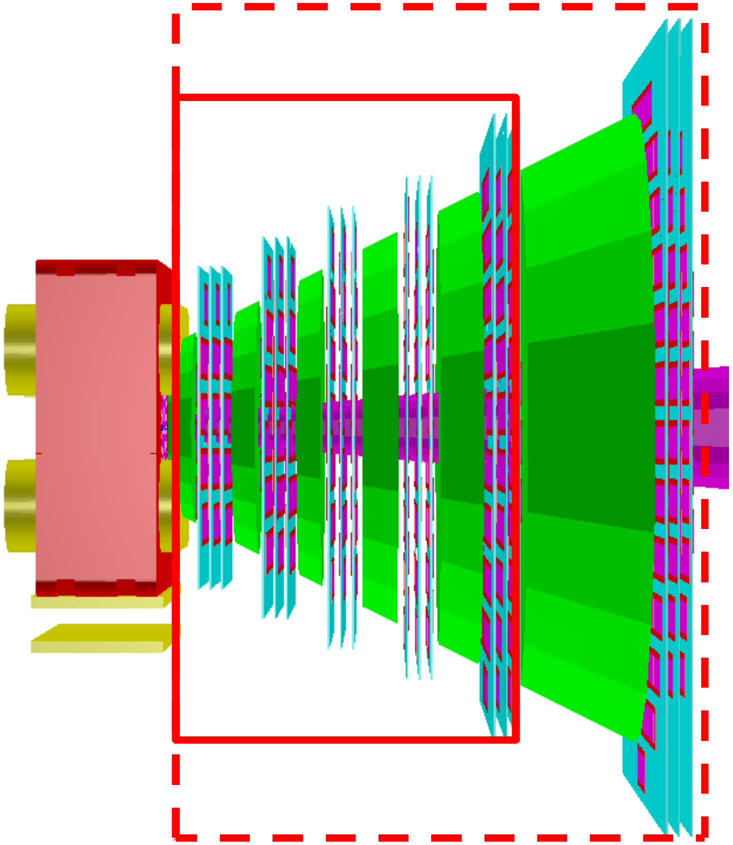
- large track multiplicities
  - up to **800** charged particles per reaction in +/- 25°
- high reaction rate
  - up to **10 MHz**
- measurement of rare probes → need for fast and efficient trigger algorithms
- high hit density
- large material budget – especially in the muon detector
- complex detector structure, overlapping sensors, dead zones
- enormous amount of data: foreseen are currently 1Gb/s archiving rate (600Tb/week)



Central Au+Au collision at 25 *A*GeV (UrQMD + GEANT3)

→ **fast tracking algorithms are essential**

The Muon detector (MUCH)



Choose alternating detector-absorber layout for continuos tracking of the muons through the absorber

Measurements of:

Low mass vector mesons
5 Fe absorbers (**125 cm**)
7.5 λ_I, p> 1.5 GeV/c

Charmonium
6 Fe absorbers (**225 cm**)
13.5 λ_I, p> 2.8 GeV/c

3 detector stations between the absorbers

Detector challenges:

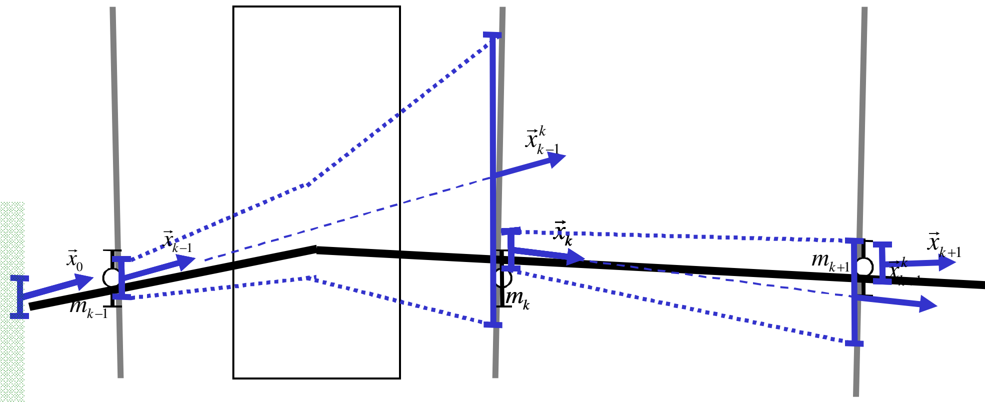
- High hit density (up to 1 hit per cm^2 per event)
- High event rates (10^7 events/s)
- Position resolution < 300 μm

→ use pad readout (e.g. GEMs), minimum pad size $2.8x2.8\ mm^2$.
→ for the last stations **straw tubes** are under investigation

Track reconstruction algorithm

Two main steps:

- **Tracking**
- **Global track selection**



Tracking is based on

- Track following
- Initial seeds are tracks reconstructed in the STS (fast Cellular Automaton (CA), I.Kisel)
- Kalman Filter
- Validation gate
- Hit-to-track association techniques
  - **nearest neighbor:** attaches the closest hit from the validation gate

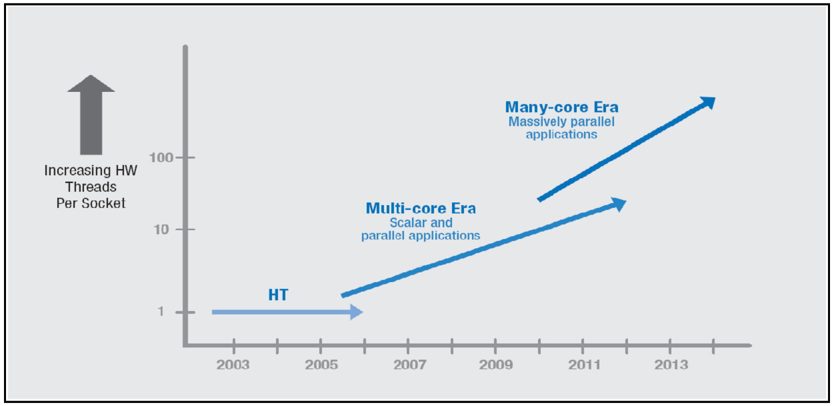
Global track selection

- aim: remove clone and ghost tracks
- tracks are sorted by their quality, obtained by chi-square and track length
- Check for shared hits

Track propagation

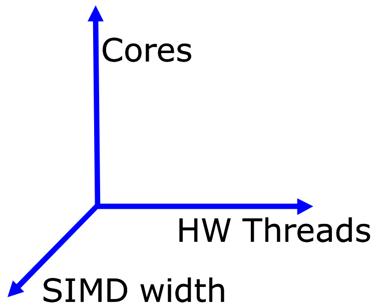
- *Inhomogeneous magnetic field:*
  - solution of the equation of motion with the 4th order Runge-Kutta method
- *Large material budget:*
  - Energy loss (ionization: Bethe-Bloch, bremsstrahlung: Bethe-Heitler, pair production)
  - Multiple scattering (Gaussian approximation)

Parallelism

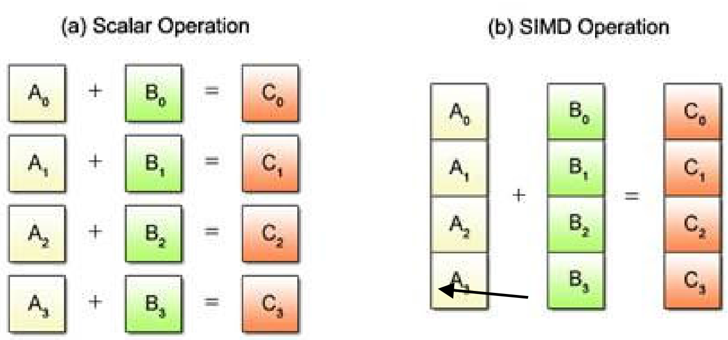


S. Borkar et al. (Intel), "Platform 2015: Intel Platform Evolution for the Next Decade", 2005.

- We´re already at the age of parallel computing
- Parallel computing relies on parallel hardware
- Parallel computing needs parallel software
- So parallel programming is very important
  - new way of thinking
  - identification of parallelism
  - design of parallel algorithm
  - implementation can be a challenge



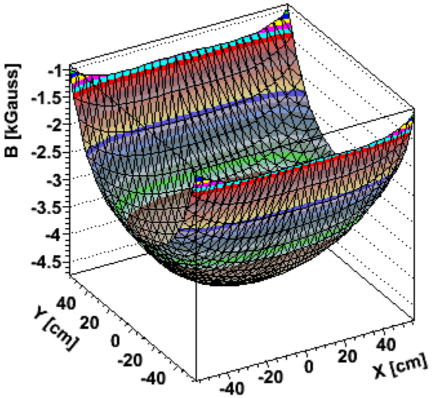
- SIMD – Single Instruction Multiple Data
  - CPU’s have it!
  - **Today**: SSE - 128 bit registers
    - 4 x float
  - **Future**: AVX, LRB
    - AVX: 8 x float
    - LRB: 16 x float
  - Benefits:
    - X time more operations per cycle
    - X time more memory throughoutput
- Multithreading
  - Many core era coming soon…
  - Tool for CPU: Threading Building Blocks

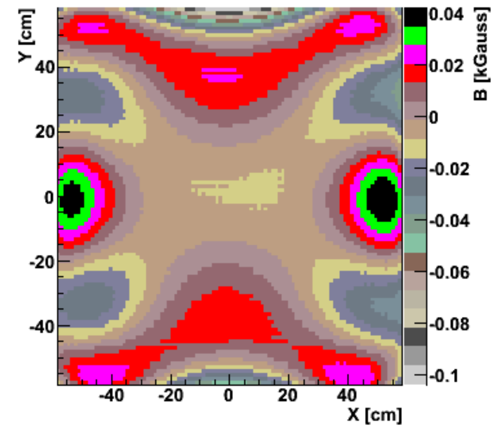


4 concurrent add operations

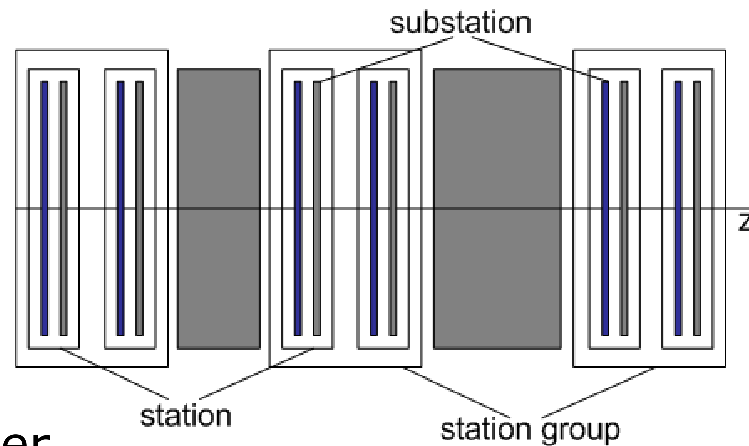
Optimization of the algorithm

- Minimize access to global memory
  - Approximation of the 70 MB large magnetic field map
    - 5 degree polynomial in the detector planes
    - parabola between the stations





- Simplification of the detector geometry
  - Problem
    - Monte-Carlo geometry consists of 800000 nodes
    - Geometry navigation based on ROOT TGeo
  - Solution
    - Create simplified geometry by converting Monte-Carlo geometry
    - Implement fast geometry navigation for the simplified geometry



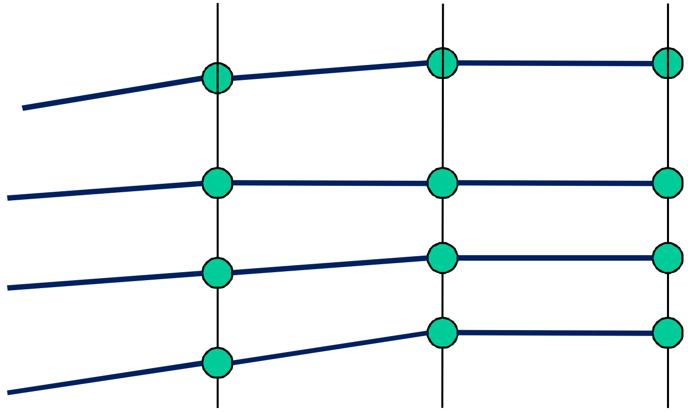
- Computational optimization of the Kalman Filter
  - From **double** to **float**
  - Implicit calculation on non-trivial matrix elements
  - Loop unrolling
  - Branches (`if then else` ..) have been eliminated

All these steps are necessary to implement SIMD tracking

SIMDization of the track fitter

- Vectorization of the tracking data
  - X_{vector} contains (X_0, X_1, X_2, X_3)
- Vectorization of the track fitter algorithm
  - SSE instructions were overwritten in a header file
- All tracks are independent and fitted by the same algorithm: 4 tracks packed into one vector and fitted in parallel

Finally SIMD version of the track fitter can fit 4 tracks at a time!



Serial tracking

Initial serial scalar version

```
for (nofTrackSeeds)
    Follow each track through the detector
    Pick up nearest hits
end for (nofTrackSeeds)
```

Can not be directly used in SIMD approach, so it has to be significantly restructured!

Parallel tracking

```
for (station groups)
    parallel_for (nofTracks/4)
        PackTrackParameters
        SIMDPropagateThroughAbsorber
    end parallel_for (nofTracks/4)
    for (stations)
        parallel_for (nofTracks/4)
            PackTrackParameters
            SIMDPropagateToStation
            AddHitToTrack
        end parallel_for (nofTracks/4)
    end for (stations)
end for (station groups)
```

Parallel loop (parallel_for over nofTracks/4, inside the station-groups loop)

Parallel loop (parallel_for over nofTracks/4, inside the stations loop)

Performance of the track fit

Track fit quality

Residuals					Pulls				
X [cm]	Y [cm]	Tx $*10^{-3}$	Ty $*10^{-3}$	q/p $*10^{-3}$ [GeV^{-1}]	X	Y	Tx	Ty	q/p
0.38	**0.39**	**9.1**	**8.7**	**3.4**	**1.02**	**0.99**	**1.08**	**1.08**	**0.92**

Speedup of the track fitter

	Time [μs/track]	Speedup
Initial	**1200**	**-**
Optimization	**13**	**92**
SIMDization	**4.4**	**3**
Multithreading	**0.5**	**8.8**
Final	**0.5**	**2400**

Throughput: $2*10^6$ tracks/s

Computer with 2xCPUs Intel Core i7 (8 cores in total) at 2.67 GHz

Performance of the parallel tracking

Simulation:

• 1000 UrQMD events at 25 AGeV Au-Au collisions + 5 μ+ and 5 μ- embedded in each event

	Initial version	Parallel version
Efficiency [%]	**94.7**	**94.0**

Speedup of the track finder

	Time [ms/event]	Speedup
Initial	**730**	**-**
Optimization	**7.2**	**101**
SIMDization	**4.8**	**1.5**
Multithreading	**1.5**	**3.3**
Final	**1.5**	**487**

Computer with 2xCPUs Intel Core i7 (8 cores in total) at 2.67 GHz

Summary

- fast reconstruction algorithms are essential!
- succesful demonstration that tracking in the muon detector is feasible in a high track density environment:
  - 95% tracking efficiency for muons passing the absorber
  - low rate of ghosts, clones and track mismatches
- Considerable optimization and parallelization of the algorithms allows to achieve a speed up factor of 2400 for the track fit and a factor 487 for the track finder
- All algorithms were implemented and tested in the CBMROOT framework and are intensivly used by CBM physicists

→ use established tracking routines for layout optimization

→ Investigate new parallel languages (CUDA, OpenCL)