KF Particle Finder

Yuri Fisyak[1], Ivan Kisel[2,3,4],
Iourii Vassiliev[4] , Maksym Zyzak[3,4]

1 – Brookhaven National Laboratory
2 – Goethe-Universität Frankfurt, Frankfurt am Main, Germany
3 – Frankfurt Institute for Advanced Studies, Frankfurt am Main, Germany
4 – GSI Helmholtzzentrum für Schwerionenforschung GmbH, Darmstadt, Germany

Fifth International Workshop for Future Challenges in
Tracking and Trigger Concepts
13.05.2014

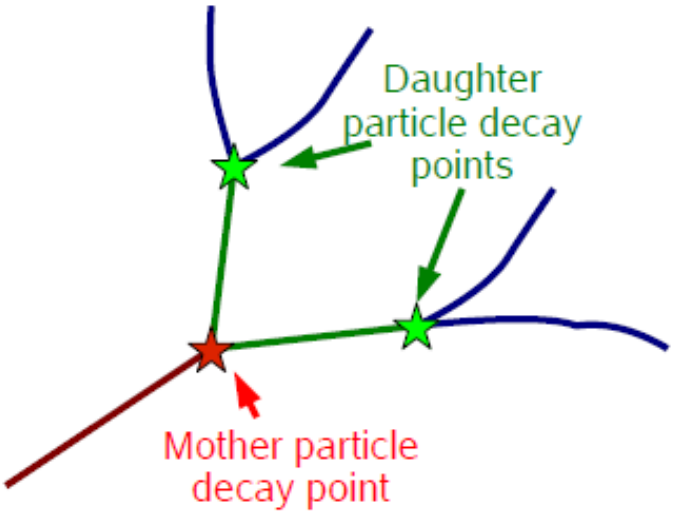
Content

- KF Particle: concept and functionality
- The KF Particle Finder package
- Short-lived particles reconstruction with KF Particle Finder:
  - pC collisions at SIS 100 energies
  - AuAu collisions at SIS 100 energies
  - AuAu collisions at SIS 300 energies
- Scalability of the package on many-core systems
- Primary Vertex finder
- Summary and Plans

Concept of KF Particle

Concept:

- Mother and daughter particles have the same state vector and are treated in the same way
- Geometry independent
- Kalman filter based



State vector — Position, momentum and energy

$$r = \{ x, y, z, p_x, p_y, p_z, E \}$$

Functionality of the package:

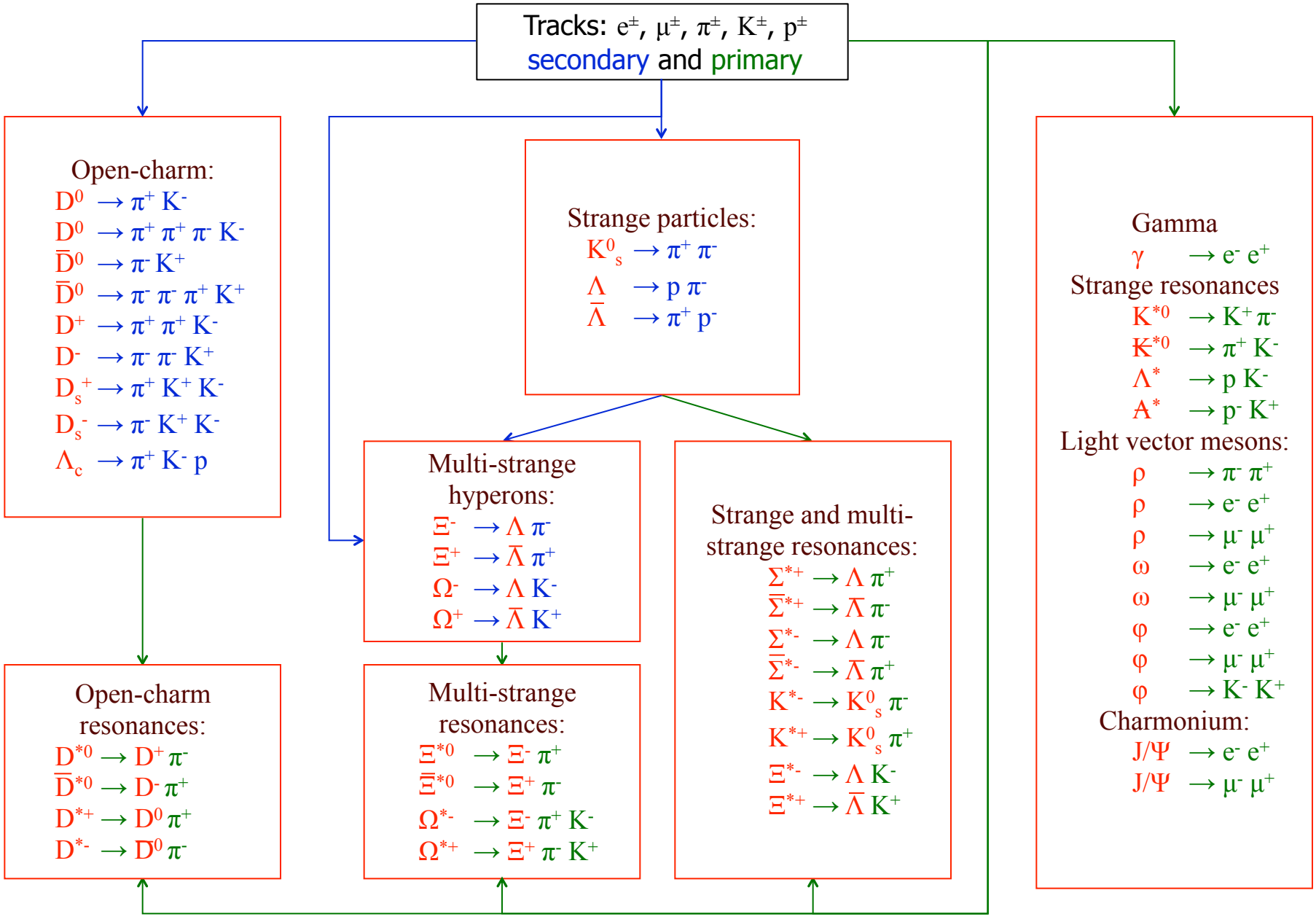
- Construction of the particles from tracks or another particles
- Decay chains reconstruction
- Transport of the particles
- Simple access to the particle parameters and their errors
- Calculation of the distance to point

Functionality of KF Particle

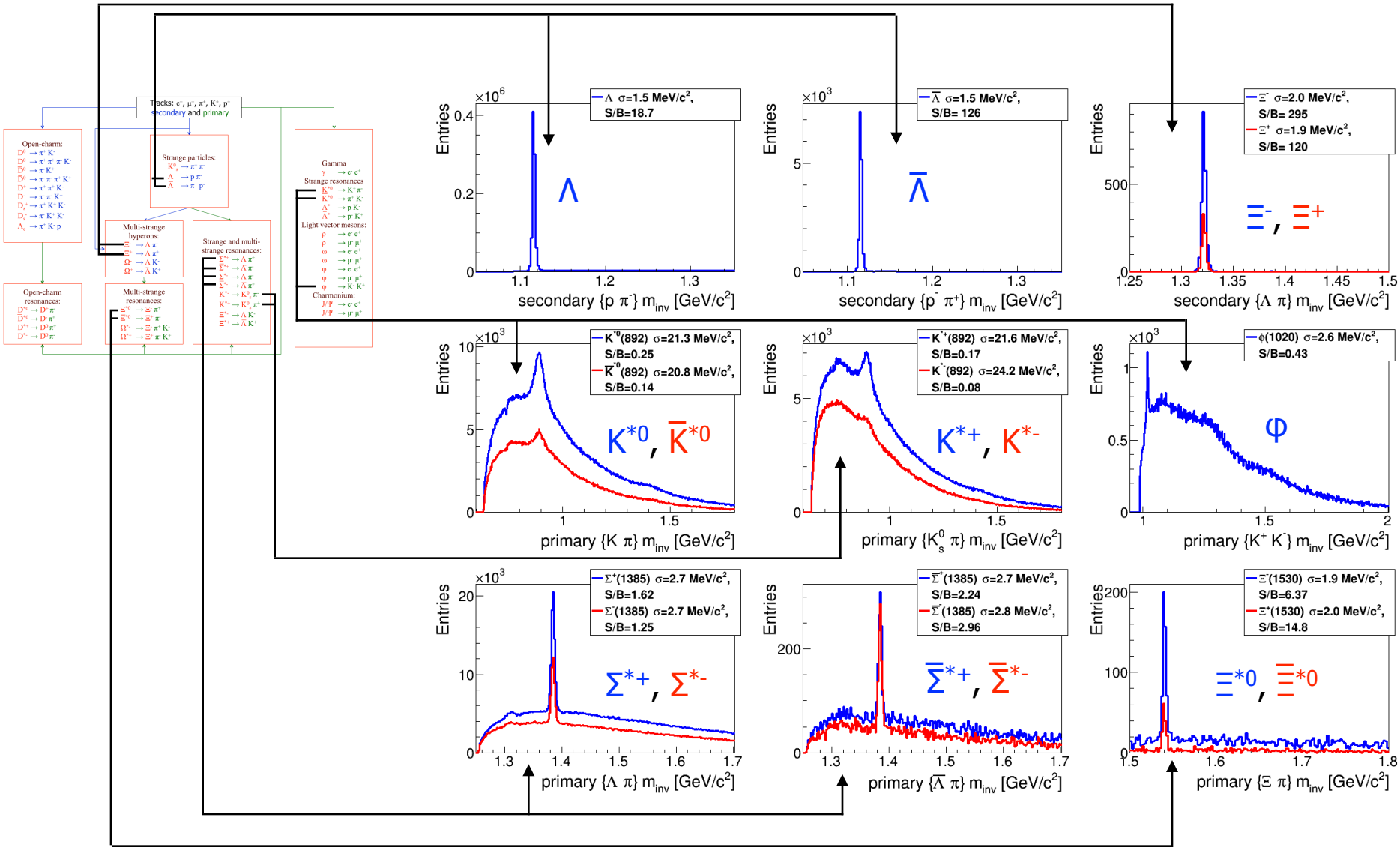
Functions	CBM	STAR	ALICE	PANDA
Construction of mother particles	+	+	+	+
Addition and subtraction of the daughter particle to (from) the mother particle	+	+	+	+
+= and -= operators	+	+	+	+
Accessors to the physical parameters (mass, momentum, decay length, lifetime, rapidity, etc)	+	+	+	+
Transport: to an arbitrary point, to the decay and production points, to another particle, to a vertex, on the certain distance	+	+	+	+
Calculation of a distance: to a point, to a particle, to a vertex	+	+	+	+
Calculation of a deviation: from a point, from a particle, from a vertex	+	+	+	+
Calculation of the angle between particles	+	+	+	+
Constraints: on mass, on a production point, on a decay length	+	+	+	+
KF Particle Finder	+	+	-	+

Functionality covers all current needs of CBM

KF Particle Finder

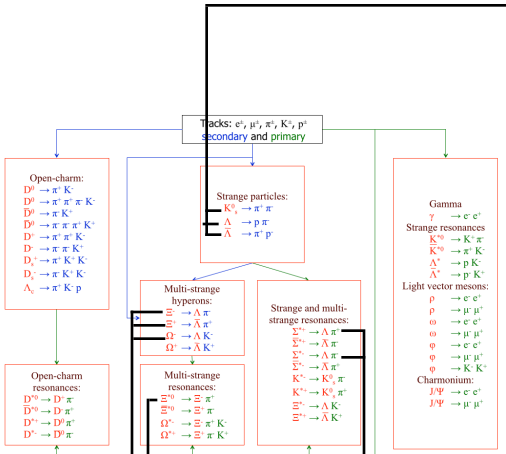


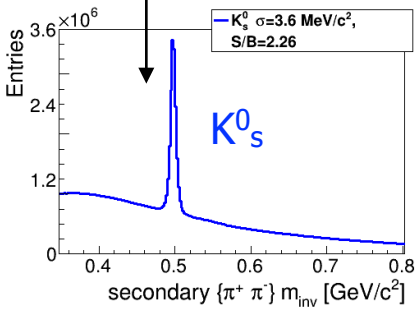
Particles Reconstruction, CBM, pC, SIS100

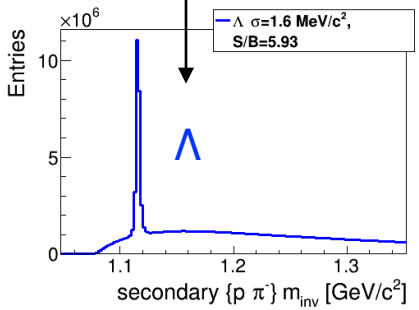


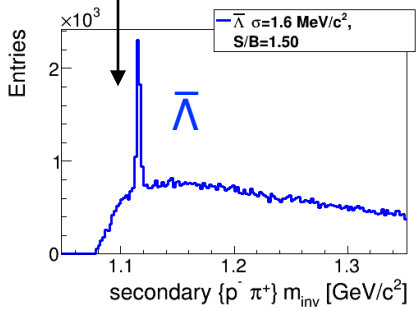
- 50 pC UrQMD MEvents at 25 AGeV with realistic ToF PID
- All particles - UrQMD output, signal was not embedded

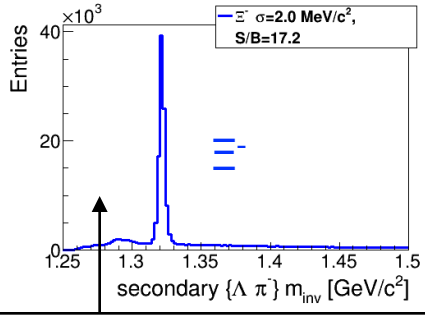
Particles Reconstruction, CBM, AuAu, SIS 100

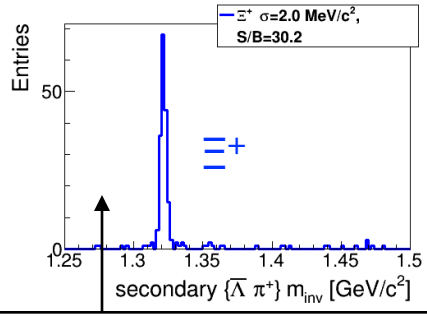


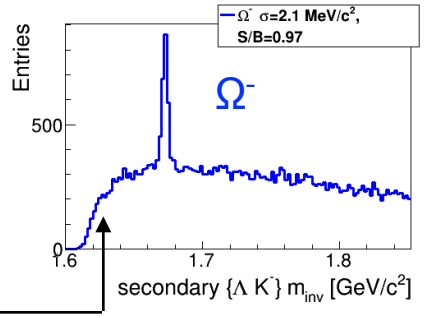


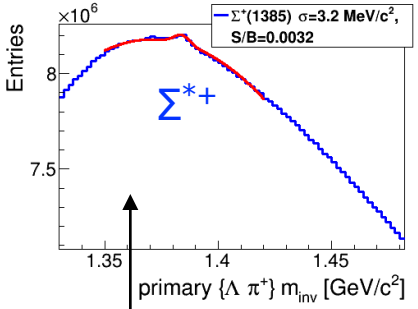


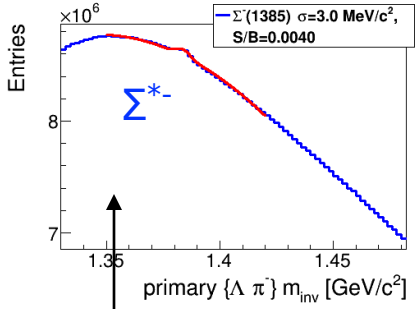


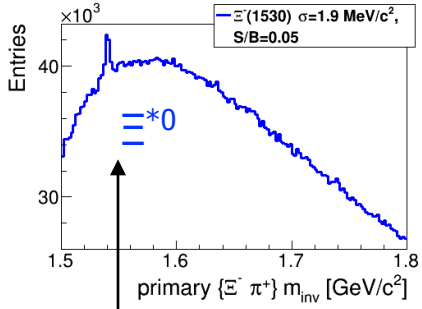






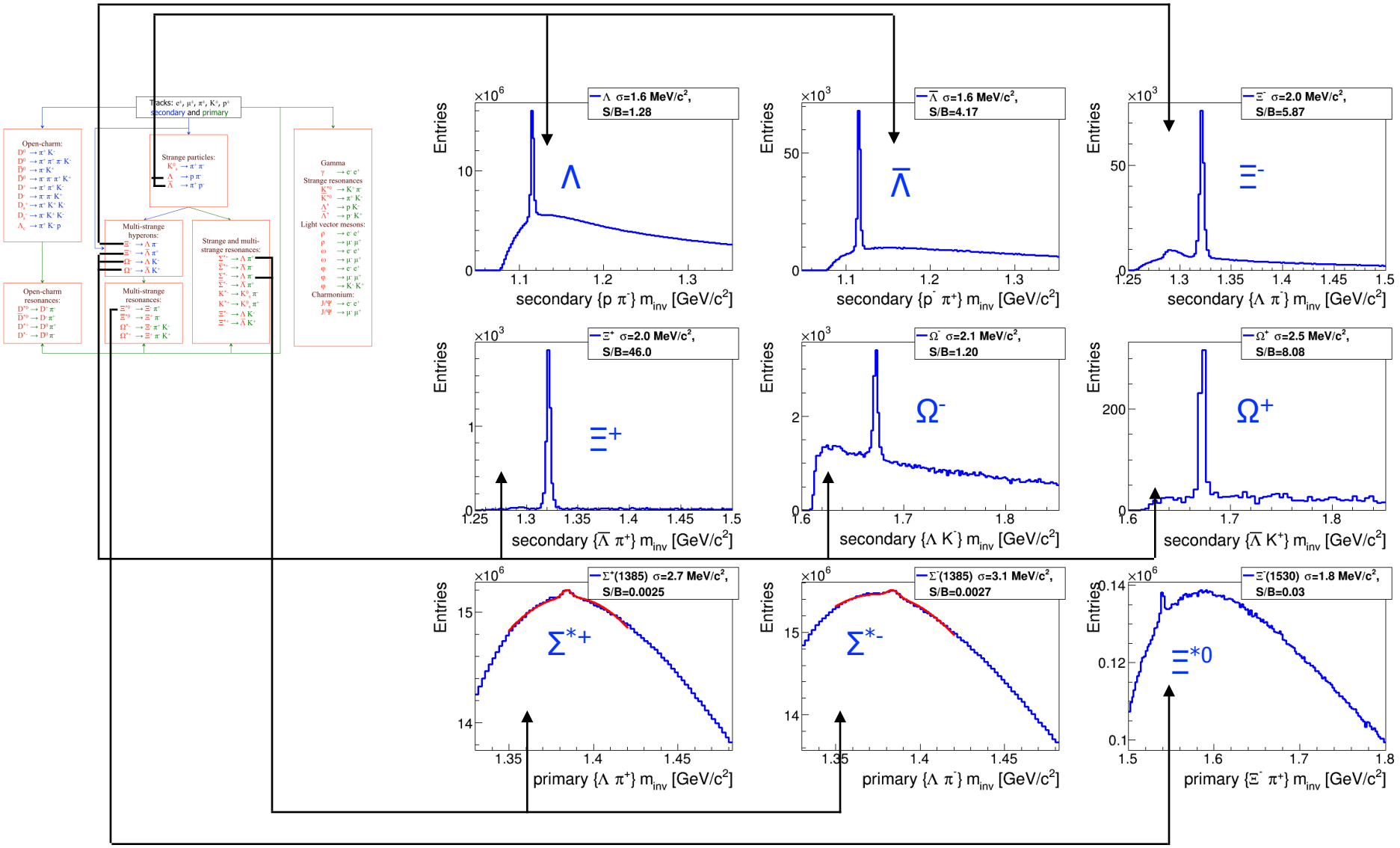






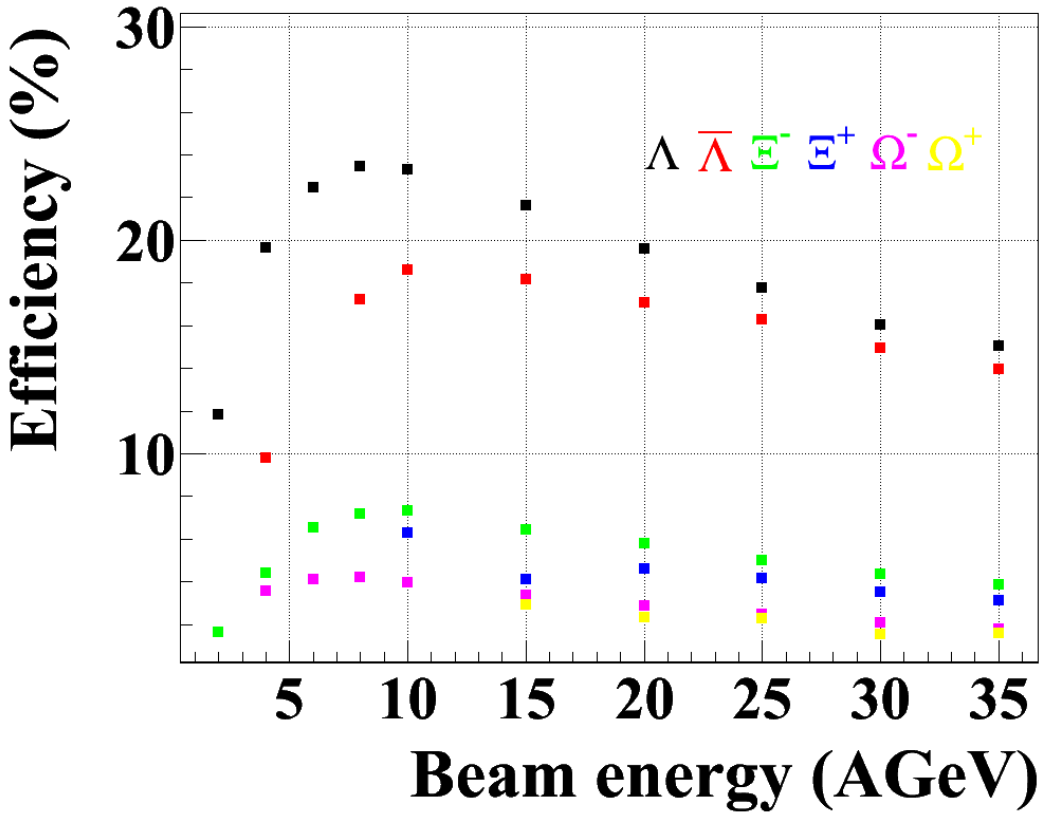
- 5 central AuAu UrQMD MEvents at 10 AGeV with realistic ToF PID
- All particles - UrQMD output, signal was not embedded

Particles Reconstruction, CBM, AuAu, SIS 300



- 5 central AuAu UrQMD MEvents at 35 AGeV with realistic ToF PID
- All particles - UrQMD output, signal was not embedded

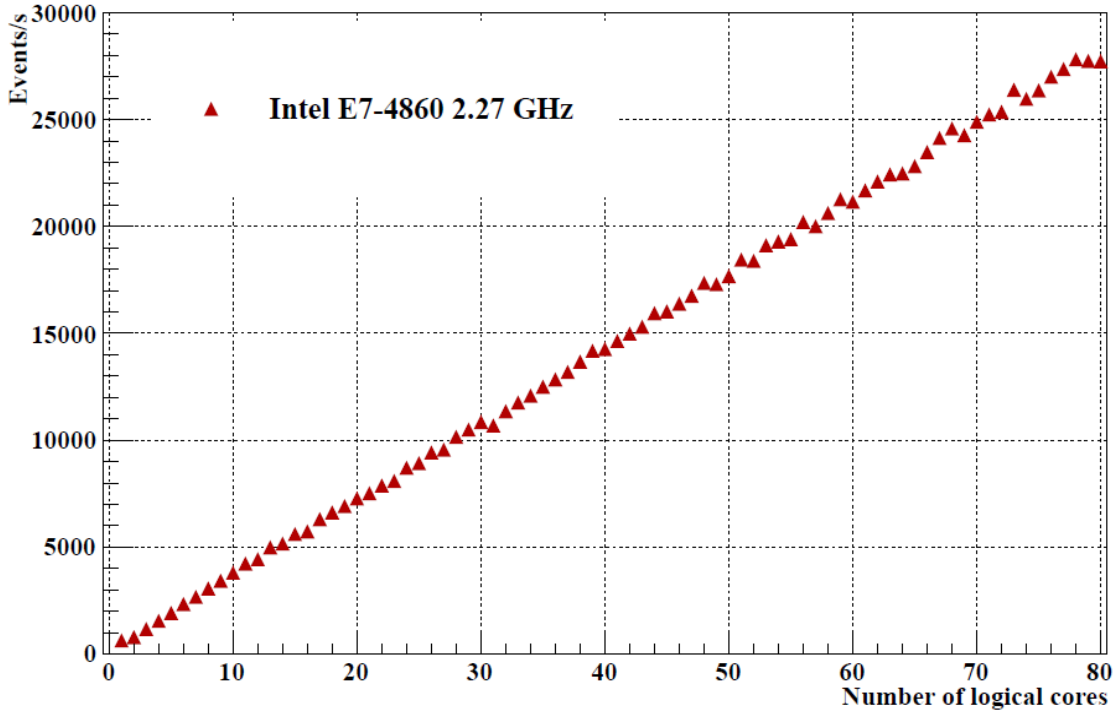
Reconstruction Efficiency of Hyperons in CBM



- The efficiency was normalised to all produced particles (in 4π).
- The package shows high efficiency.
- The efficiency for antiparticles is slightly lower due to the tighter criteria on antiproton selection.

Scalability on Many-core CPU System

- The KF Particle Finder has been parallelized using Intel TBB.
- The KF Particle Finder shows linear scalability on many-core machines (the scalability on a computer with 40 physical, 80 logical cores is shown).



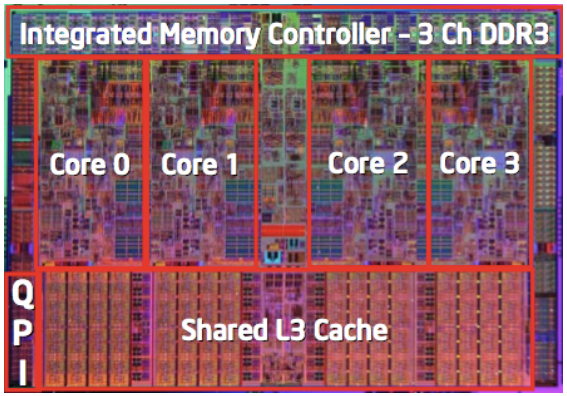
1 thread per 1 core, each filled with 1000 events,
AuAu mbias events at 25 AGeV

The speed of the package:

- mbias AuAu collisions at 25 AGeV – 1.5 ms/event
- central AuAu collisions at 25 AGeV – 10.5 ms/event

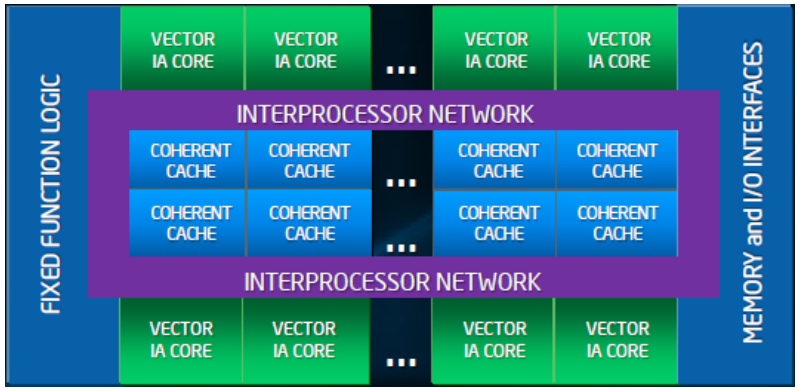
CPU vs Xeon Phi

Intel/AMD CPU



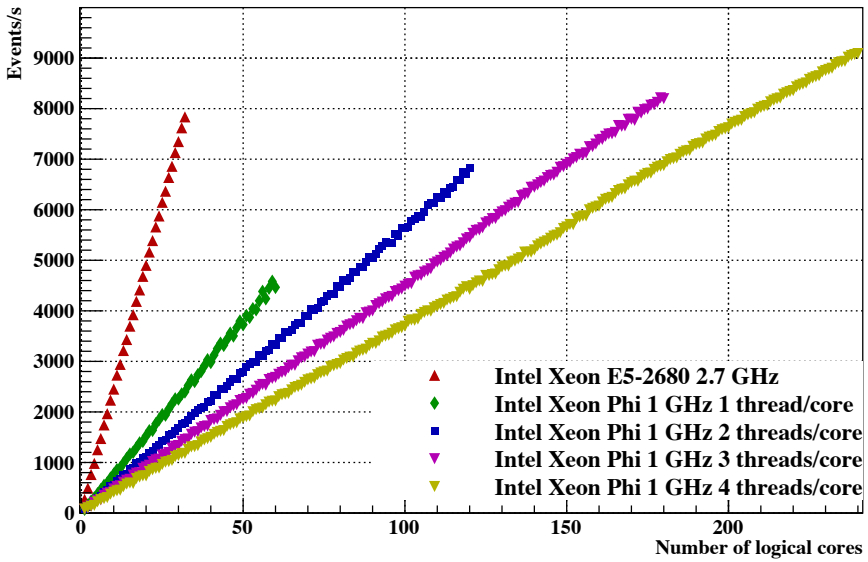
- Up to 12 cores per CPU, 4 CPUs per server
- SIMD vector of size 4-8
- Usually 1-2 threads per core
- Large caches (typically, 256 kB of L2, up to few MB of L3 cache per thread)
- High operation frequency (about 2-3 GHz)
- Large amount of RAM (several GB per thread)

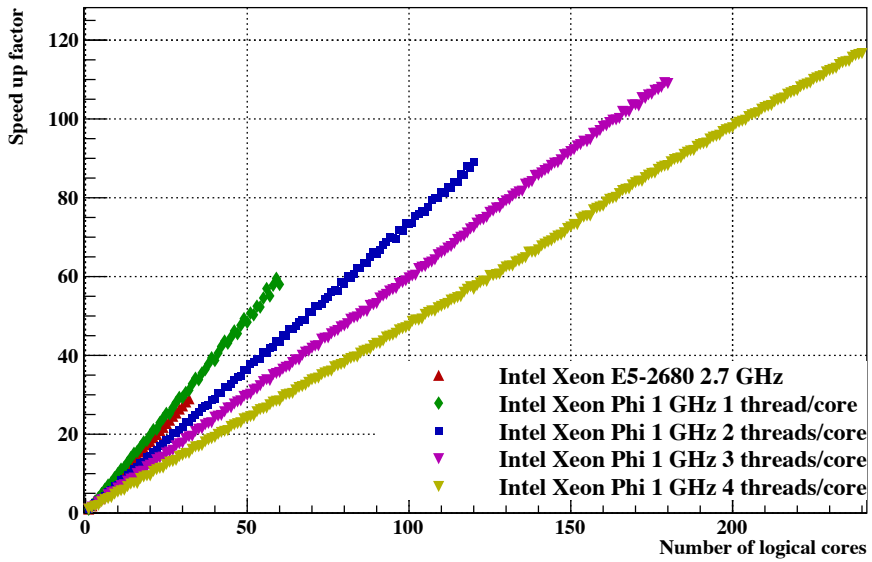
Intel Xeon Phi



- 60 or 61 cores per card
- SIMD vector of size 16
- 4 threads per core
- Small amount of cache (128 kB of L2, no L3 cahce)
- Small operation frequency of the cores (about 1 GHz)
- Only 8-16 GB of RAM for all 240 threads

KF Particle Finder on Xeon Phi

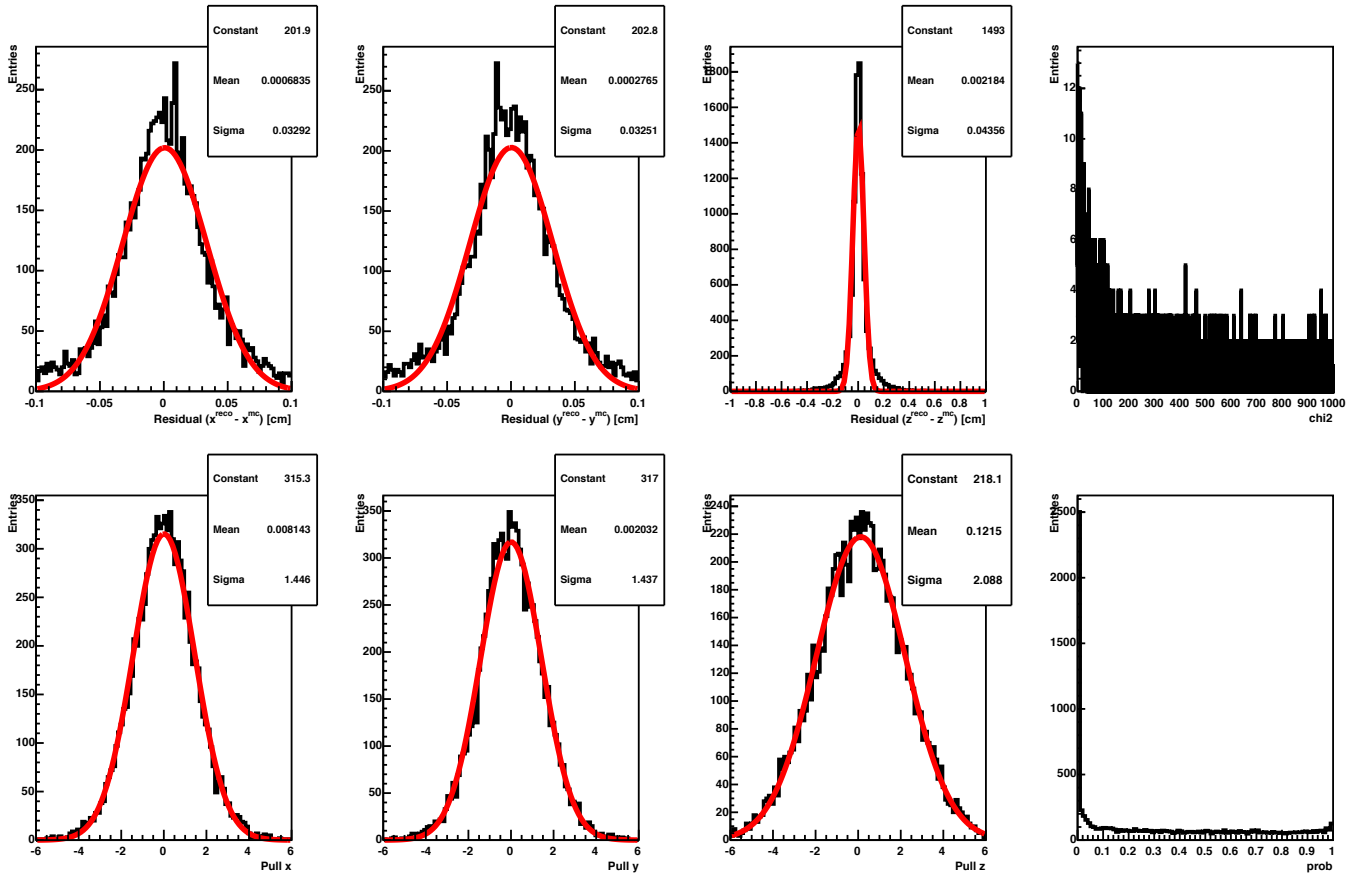




- Standalone KF Particle Finder for STAR is adapted for the Xeon Phi card. The same code is used for the CPU and the Xeon Phi.
- The program is tested on search of K_s^0, Λ, Ξ^- and Ω^- particles.
- The parallelism between cores is implemented on the event level. Tests with 100 U+U mbias events per thread were performed.
- The program scales up to 240 logical cores on the Xeon Phi.
- Time per one thread: CPU - 3.7 ms, Xeon Phi - 12.8 ms.
- SIMD Speedup: SSE (CPU, 4 elements) - 3.67, AVX (CPU, length 8) - 4.67, IMIC (Xeon Phi, length 16) - 8.43.

With the STAR HLT group

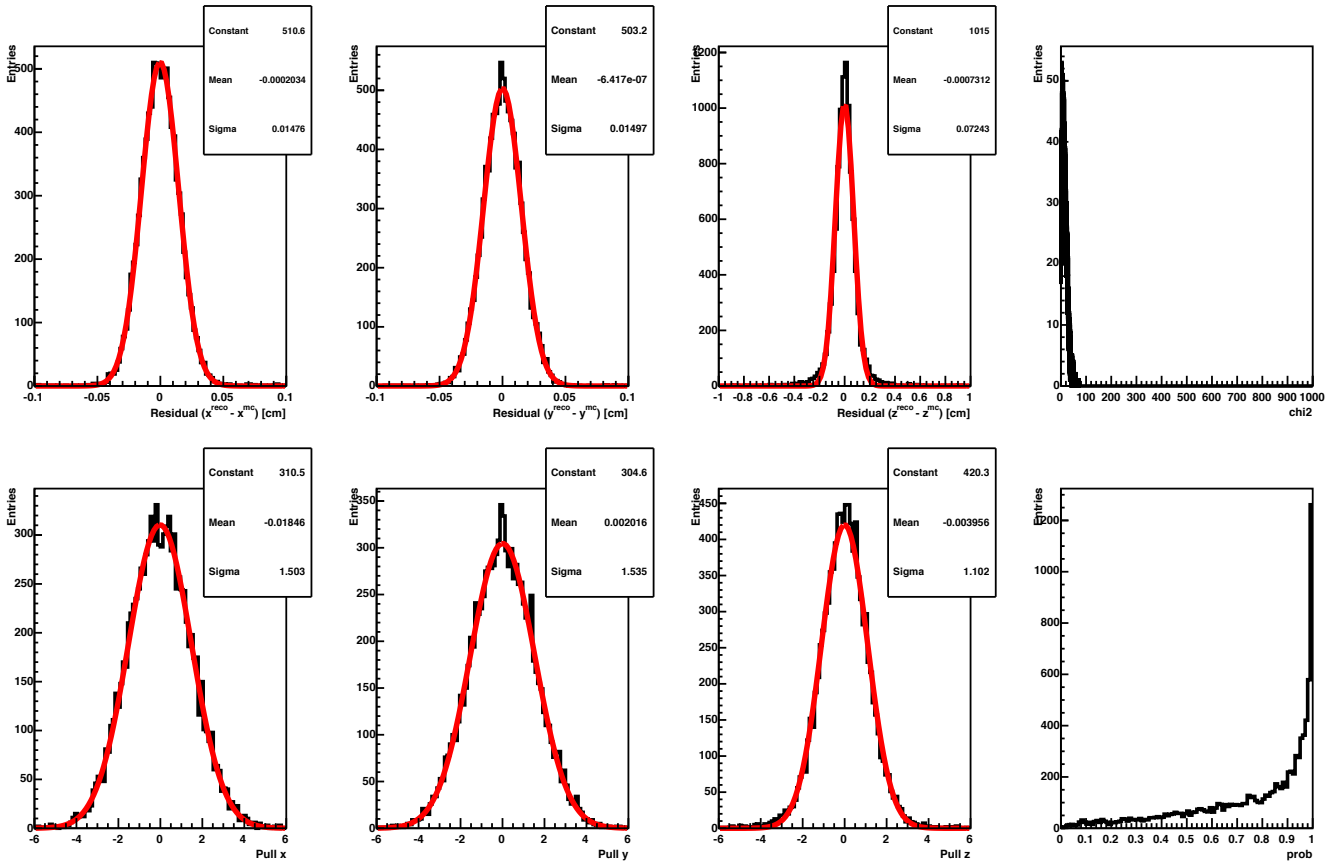
Primary Vertex Finder. AuAu Collisions



Efficiency	96.8 %
Clone level	0.0 %
Ghost level	2.9 %

STAR, AuAu mbias events, 200 GeV, only 1 PV is selected

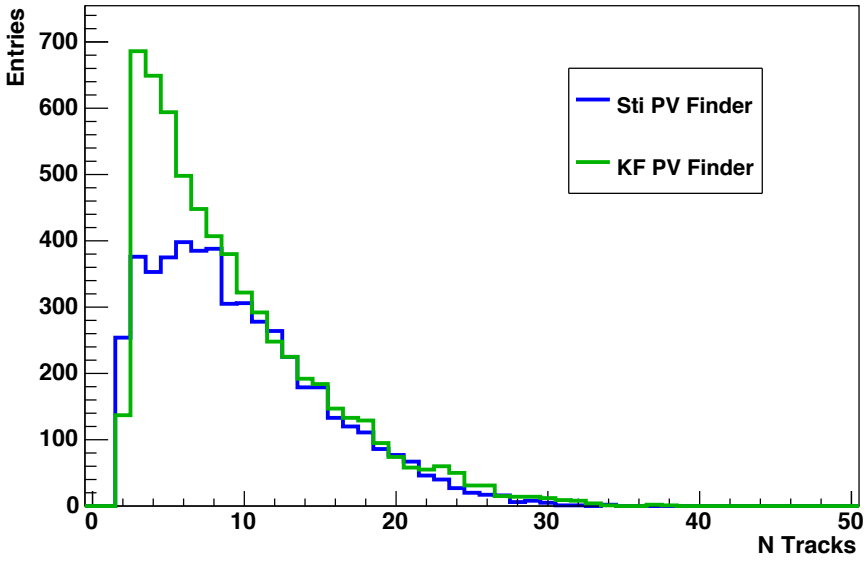
Primary Vertex Finder. pp Collisions

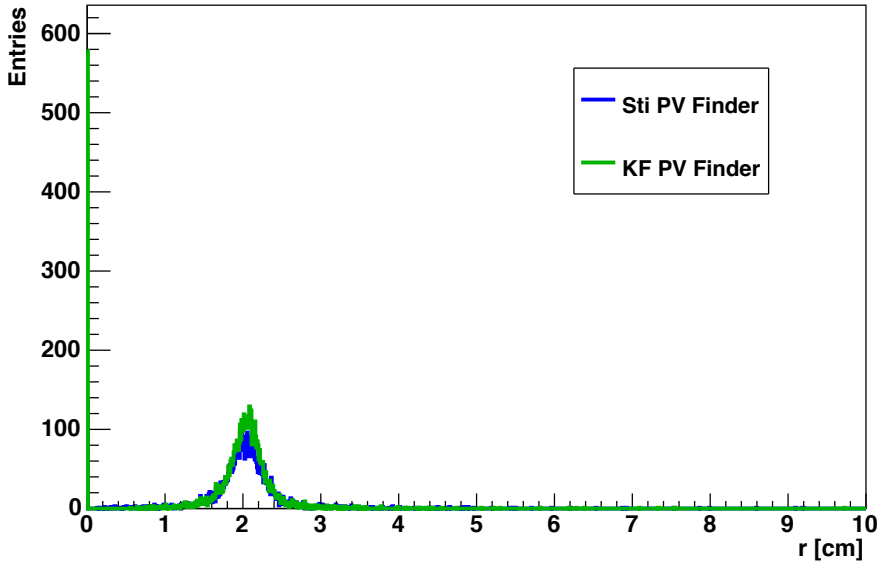


Efficiency	95.5 %
Clone level	3.2 %
Ghost level	6.9 %

STAR, pp events, 200 GeV

Reconstruction of Pipe Primary Vertex





- The PV Finder was tasted with simulated events with beam (Au, 200 AGeV) colliding with pipe.
- Reconstruction efficiency of pipe PVs is about 55%.
- About 90% of reconstructed vertices lie in the peak around the pipe position.
- The tests with real data are in progress.

Summary and Plans

Summary

- KF Particle Finder was tested with UrQMD events with a high statistics in wide energy ranges.
- The package shows reconstruction efficiencies of about 15% for Λ and 5% for Ξ with AuAu events at 35 AGeV together with high signal to background ratios (1.3 and 5.9 respectively).
- The algorithm shows high speed (1.5 ms per mbias AuAu event at 25 AGeV), shows linear scalability on many-core systems including Intel Xeon Phi coprocessor.
- The first version of Primary Vertex finder was added to the KF Particle Finder package. The PV finder shows high efficiency (more than 95%).

Future plans

- Unify all versions and put to the common repository.
- Use on the real data with STAR and ALICE experiments.