Parallel Implementation of the KFParticle Vertexing Package for the CBM and ALICE Experiments

Ivan Kisel[1,2,3], Igor Kulakov [1,4], Maksym Zyzak[1,4]

1 – Goethe-Universität Frankfurt, Frankfurt am Main, Germany
2 – Frankfurt Institute for Advanced Studies, Frankfurt am Main, Germany
3 – GSI Helmholtzzentrum für Schwerionenforschung GmbH, Darmstadt, Germany
4 – Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Computing in High Energy and Nuclear Physics
24 May 2012
New York

HGS-HIRe for FAIR
Helmholtz Graduate School for Hadron and Ion Research

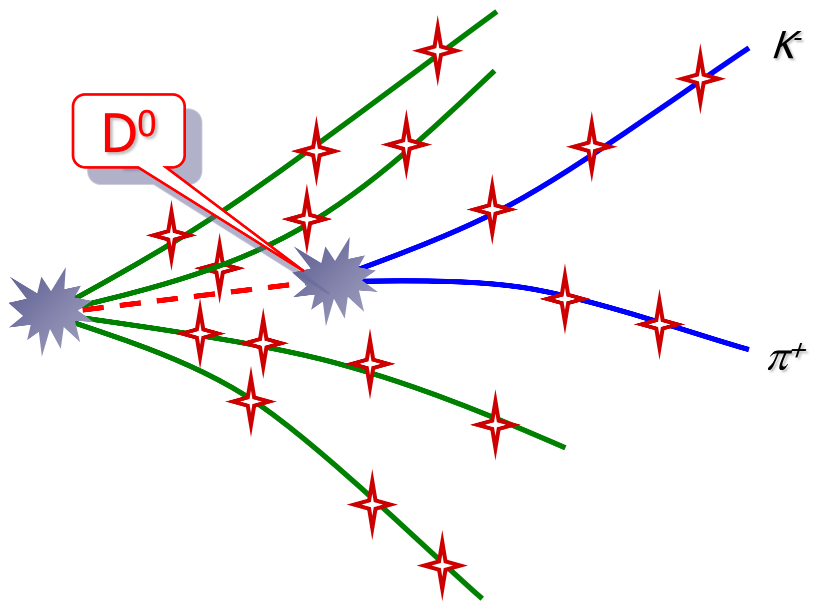
HIC for FAIR
Helmholtz International Center

HESSEN

Content

- Concept of the KFParticle package
- The block-diagram of KFParticle algorithm
- Functionality of the KFParticle package
- SIMDized KFParticle
- Particles finding with the SIMDized KFParticle package

Reconstruction of Vertices and Decayed Particles



State vector

Position, direction and momentum

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

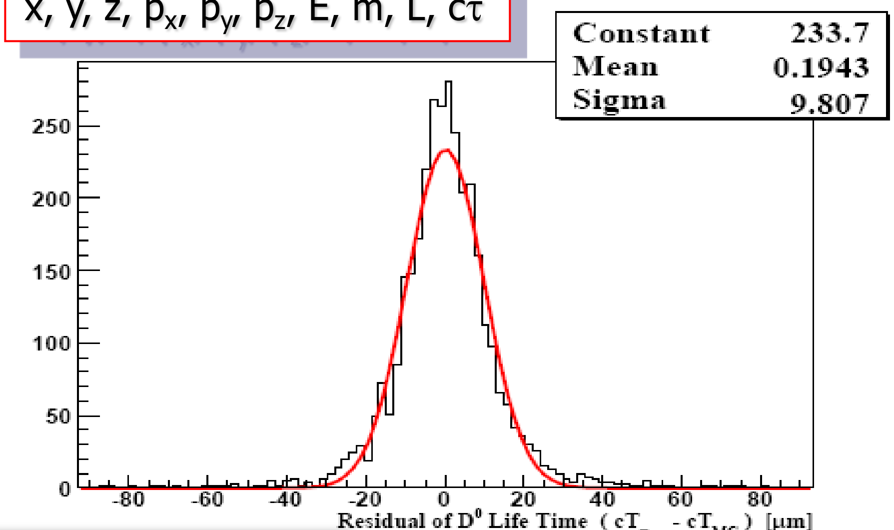
- Mother and daughter particles have the same state vector and are treated in the same way
- Geometry independent
- Reconstruction of decay chains
- Kalman filter (KF) based

```
AliKFVertex PrimVtx( ESDPrimVtx ); // Set primary vertex
// Set daughters
AliKFParticle K( ESDp1, -321 ), pi( ESDp2, 211 );

AliKFParticle  D0( K, pi ); // Construct mother
PrimVtx += D0;                // Improve the primary vertex.

D0.SetProductionVertex( PrimVtx ); // m3 is fully fitted
K.SetProductionVertex( D0 );       // K is fully fitted
pi.SetProductionVertex( D0 );       // pi is fully fitted
```

$x, y, z, p_x, p_y, p_z, E, m, L, c\tau$

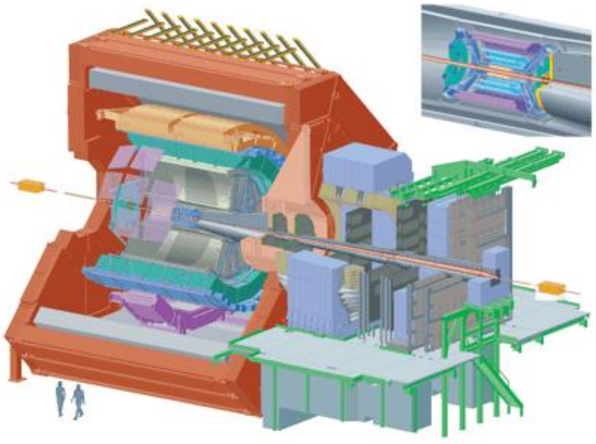


KFParticle: powerful tool for physics analysis

Experiments

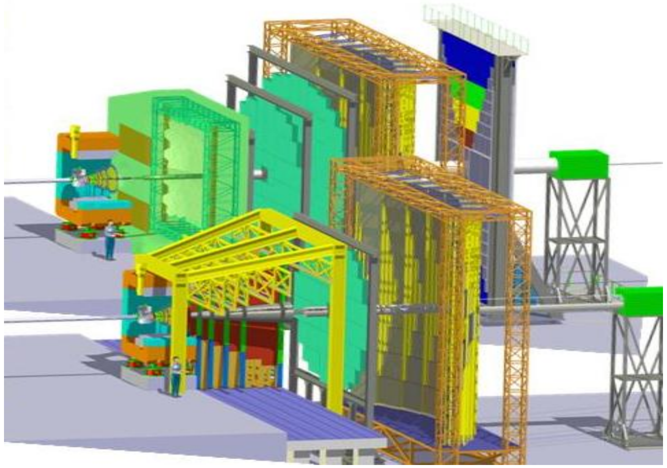
KFParticle is developed based on the ALICE and CBM experiments.

ALICE (CERN, Switzerland) – a collider experiment



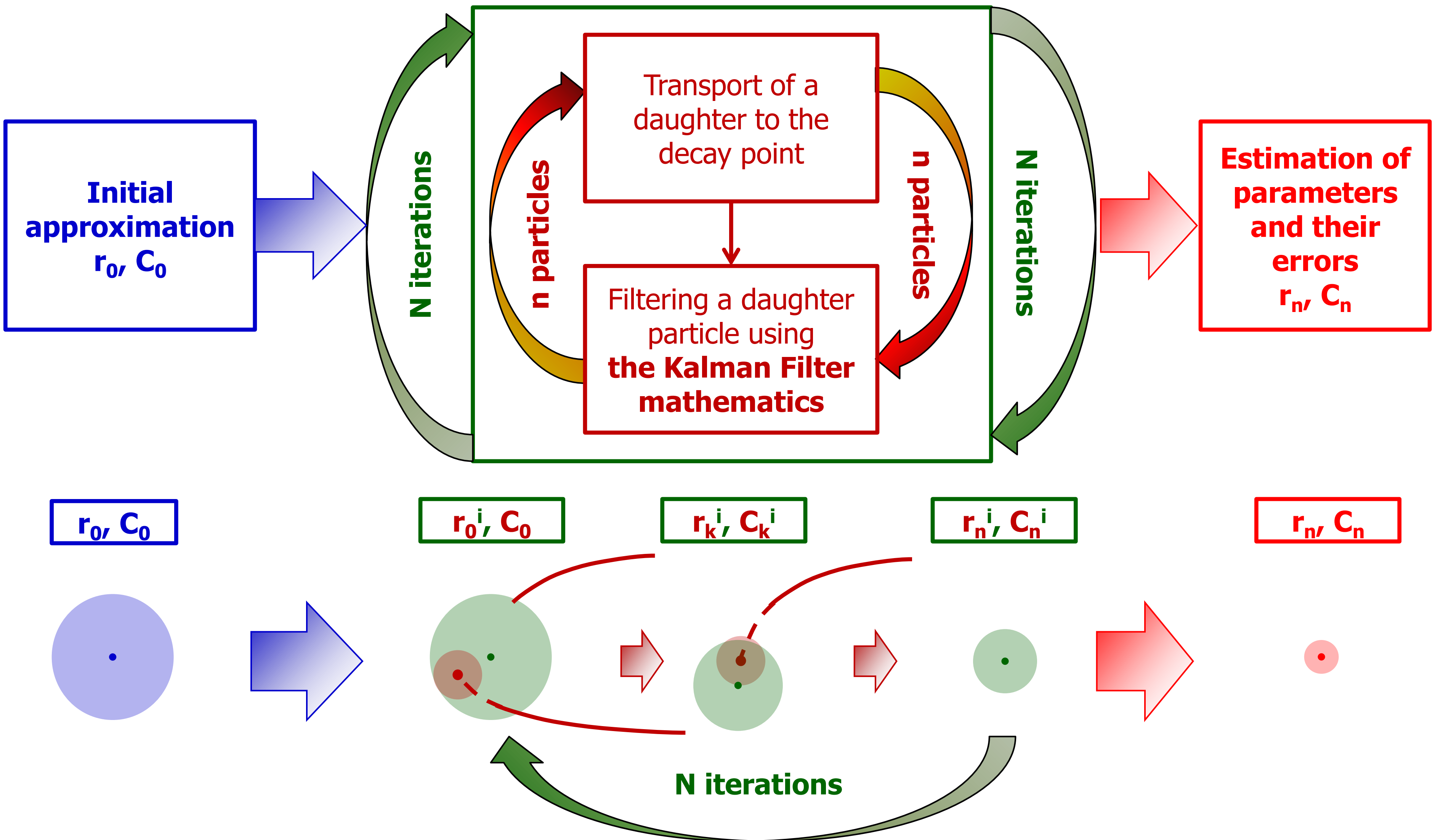
- Few 1000s charged particles/collision
- High statistic is collected – a speed of short-lived particles reconstruction is important for the physics analysis

CBM (FAIR, Germany) – a fixed-target experiment



- Up to 1000 charged particles/collision
- Non-homogeneous magnetic field
- 10^7 AuAu collisions/sec
- Reconstruction of the full event topology is required in the first level trigger
- The speed and efficiency of the reconstruction is crucial

KFParticle Algorithm



The mathematics is described in:
S. Gorbunov and I. Kisel, Reconstruction of decayed particles based on the Kalman filter. CBM-SOFT-note-2007-003, 7 May 2007

Structure of KFParticle

Low level (for developers, basic functionality)

- Transport functions
- Calculation of distances and deviations between particles, a particle and a point
- KF mathematics
- Constraints

Intermediate level (for advanced users)

- Feasibility studies
- Reconstruction of particles
- Reconstruction of decay chains

High level or KFParticle-Light (for users and triggering)

- Reconstruction of standard decays (K^0_s, hyperons, D^0-mesons,…)
- Reconstruction of event topology
- On-line selection of events

Functionality in ALICE and CBM

Functions	ALICE	CBM
Construct, SetMassConstraint, SetProductionVertex, SetVtxGuess	+	+
GetMass, GetMomentum, GetDecayLength, GetLifeTime	+	+
GetDecayLengthXY, GetPhi, GetR	+	
Extrapolate, TransportToProductionVertex(), TransportToDecayVertex()	+	+
TransportToPoint, TransportToVertex, TransportToParticle, TransportToDS,	+	
GetDStoPoint	+	+
GetDStoParticle, GetDStoParticleXY, GetDistanceFromVertex, GetDistanceFromVertexXY, GetDistanceFromParticle, GetDistanceFromParticleXY, GetDeviationFromVertex, GetDeviationFromVertexXY, GetDeviationFromParticle, GetDeviationFromParticleXY	+	
GetAngle, GetAngleXY, GetAngleRZ	+	
SubtractFromVertex, ConstructGamma	+	
SetNoDecayLength, +=, -=	+	
Particles finder		+

Functionality becomes more and more advanced

SIMDization of KFParticle

- KFParticle has been SIMDized
- The reconstruction quality is the same for the scalar version and the SIMD version:
  - Λ reconstruction in CBM

	Resolution				Pull			
	M, MeV/c^2	X, cm	Y, cm	Z, cm	M	X	Y	Z
Scalar	1.2	0.011	0.015	0.18	1.54	1.50	1.42	1.63
SIMD	1.2	0.013	0.015	0.18	1.54	1.51	1.50	1.69

  - D^0 reconstruction in ALICE (using MC data)

	Resolution				Pull			
	M, MeV/c^2	X, cm	Y, cm	Z, cm	M	X	Y	Z
Scalar	18.4	0.012	0.011	0.016	1.16	1.15	1.12	1.12
SIMD	18.5	0.012	0.012	0.016	1.19	1.16	1.15	1.11

- Speedup factor of 5 for CBM and 3 for ALICE has been achieved

Examples

Scalar version

```
AliKFParticle P1, P2;

P1 = AliKFParticle( *pTrack, PDG );


AliKFParticle V0( P1,  P2);
```

```
Double_t length, sigmaLength;
V0.GetDecayLength( length, sigmaLength ) ;
Double_t mass, sigmaMass;
V0.GetMass(mass, sigmaMass ) ;
```

```
TH1F *MassDistribution;
…
MassDistribution->Fill(mass[i]);
```

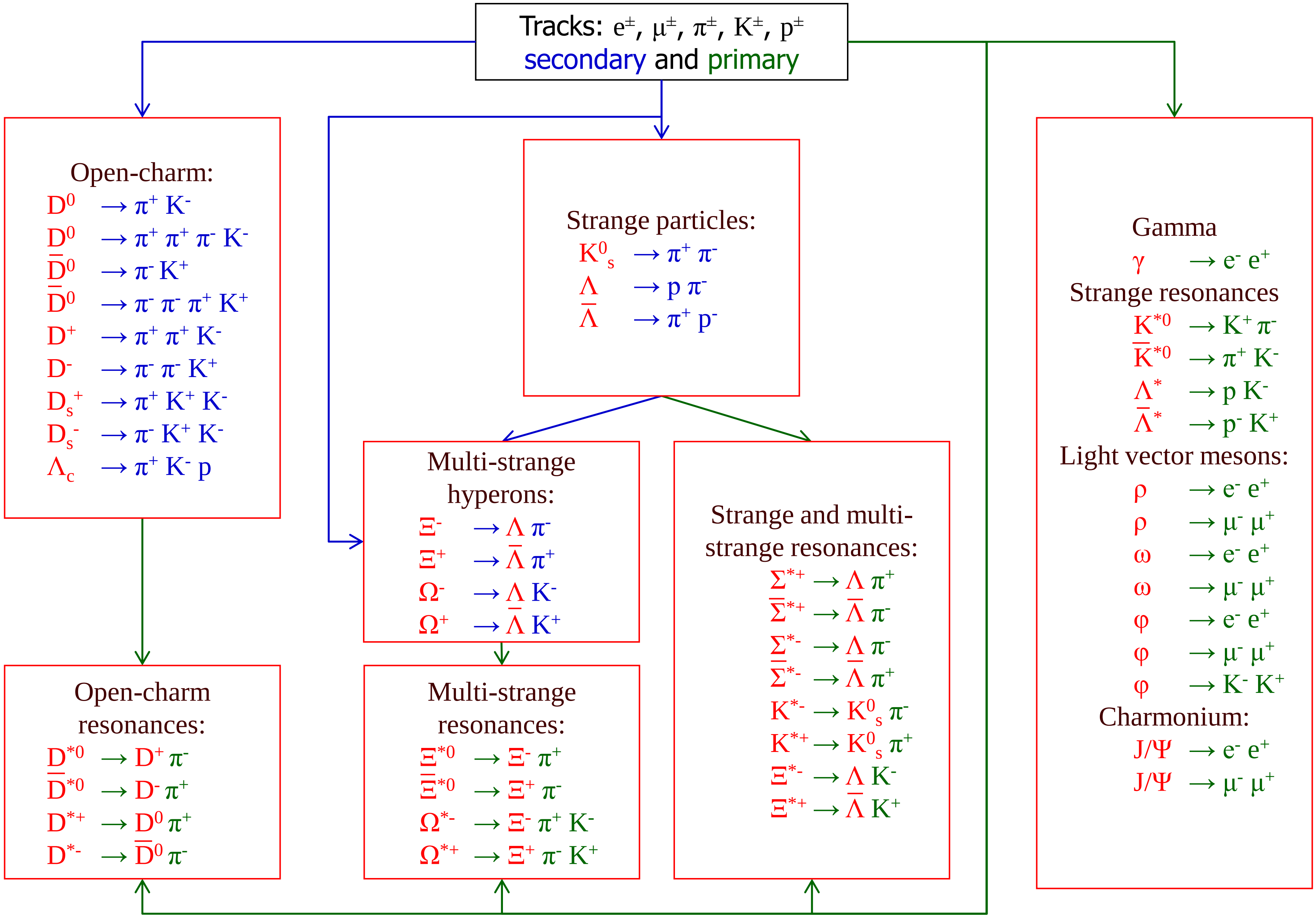
SIMD version

```
AliKFParticle P1[fvecLen], P2[fvecLen];
for(int i=0; i<fvecLen; i++)
   P1[i] = AliKFParticle( *pTrack, PDG );
AliKFParticleSIMD PartPos( P1, PDG );
AliKFParticleSIMD PartNeg( P2, PDG2 );
AliKFParticleSIMD V0( PartPos, PartNeg );
```

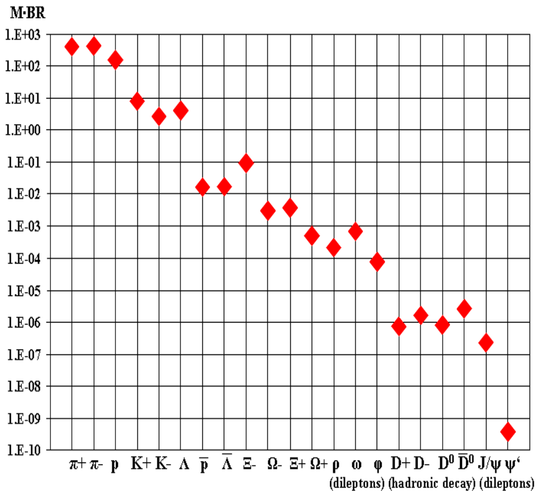
```
fvec length, sigmaLength;
V0.GetDecayLength( length, sigmaLength ) ;
fvec mass, sigmaMass;
V0.GetMass( mass, sigmaMass );
```

```
TH1F *MassDistribution;
…
for(int i=0;i<fvecLen; i++)
   MassDistribution->Fill( mass[i] );
```

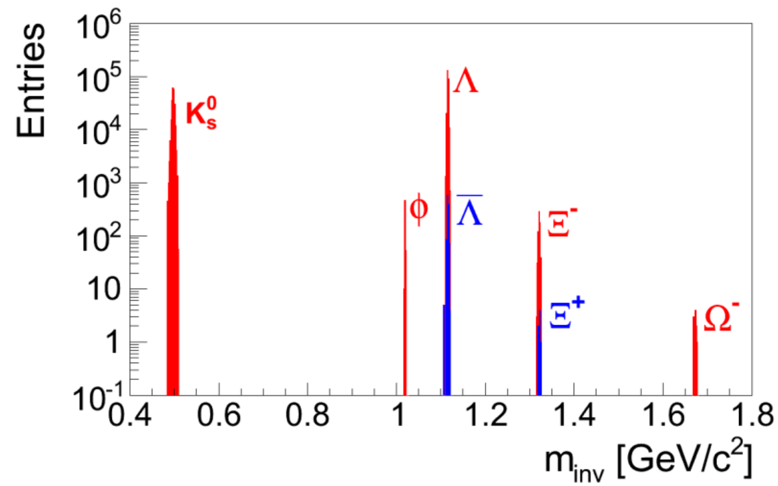
KF Particle Finder for the CBM Experiment



Multiplicities times branching ratio, heavy ion collisions, statistical model



Signal of the found particles



central AuAu collisions at 25 AGeV

The speed of the package:

- central AuAu collisions at 25 AGeV – 11.7 ms/event
- minbias AuAu collisions at 25 AGeV – 1.5 ms/event

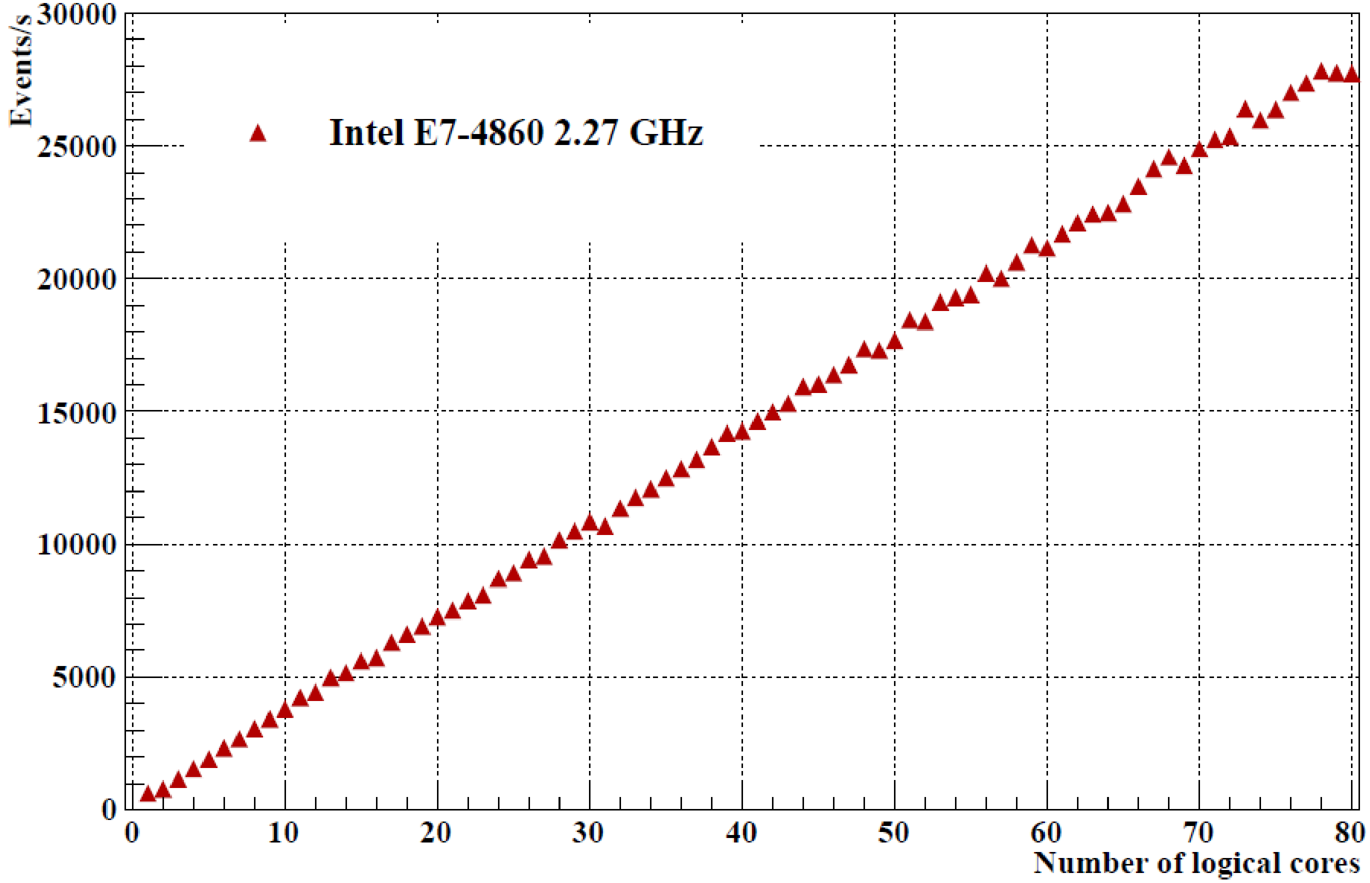
Efficiencies of the KF Particle Finder (job summary)

Particle	Eff	Ghost	BackGr	N Ghost	N BackGr	N Reco	N Clone	N MC
Kshort	0.249	0.972	0.015	18155772	284536	242696	254	972992
Lambda	0.201	0.972	0.014	18155772	257777	269527	181	1341971
Lambda b	0.213	0.972	0.028	18155772	526299	1187	0	5568
Xi-	0.023	0.969	0.001	22934	25	708	0	30198
Xi+	0.026	1.000	0.000	21842	1	9	0	348
Omega-	0.020	0.955	0.044	8869	411	10	0	506
Omega+	0.000	0.999	0.001	9391	6	0	0	11

UrQMD events, central AuAu collisions at 25 AGeV, 80 kEvents, w/o PID

Scalability on Many-core System

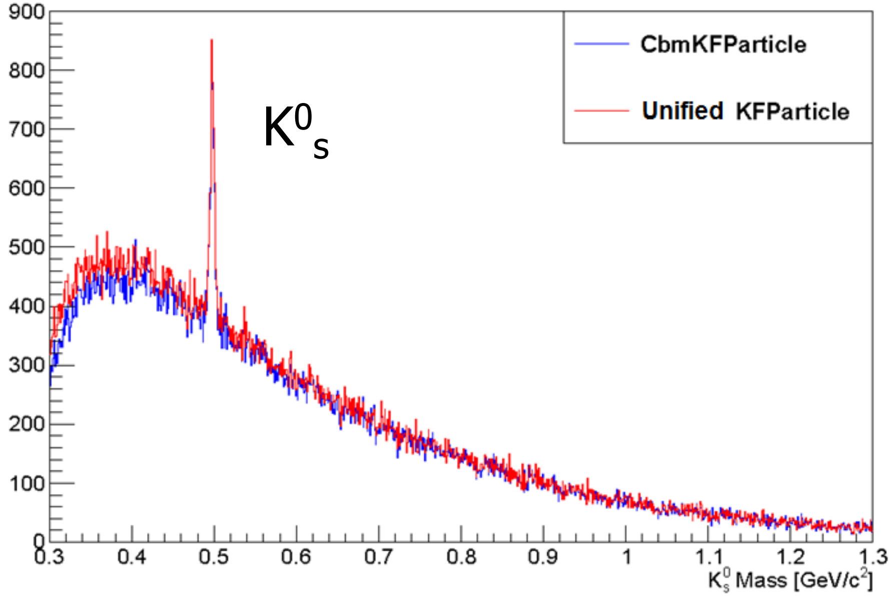
- The KF Particle Finder has been parallelized using Intel TBB.
- The KF Particle Finder shows linear scalability on many-core machines (the scalability on a computer with 80 cores is shown).

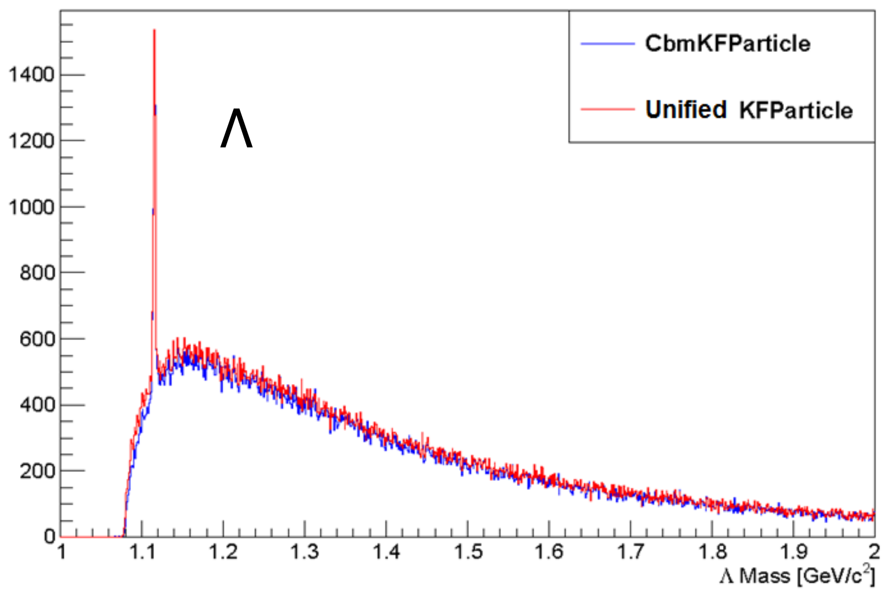


Given n threads each filled with 1000 events,
run them on specific n logical cores, 1 thread per 1 core.
AuAu mbias events at 25 AGeV

Unification of ALICE and CBM KFParticle

- The unified KFParticle package has been created and tested within the CBMRoot framework.
- The unified package has the functionality of both ALICE and CBM.
- The first tests have been done using CbmV0Analysis.





- The unified KFParticle and CbmKFParticle show similar results.
- Further tests of the package functionality will be done.

Summary and Plans

Summary

- The KFParticle package is a particle reconstruction package with a rich functionality. The functionality becomes more and more advanced.
- KFParticle has been SIMDized. SIMDized version shows the same results.
- The unified version of the KFParticle has been created.
- The particles finder has been developed based on the SIMDized KFParticle package. About 50 particles (decay channels) are included.
- The algorithm shows high speed (1.5 ms per mbias AuAu event at 25 AGeV) and efficiency, shows linear scalability on many-core systems.

Plans

- Increase the functionality of the package, create the KFParticle library.
- Implement statistical methods for the particle reconstruction and selection based on KFParticle.
- Add adaptive methods (DAF, PDAF, etc.) to KFParticle.
- Implement using parallel languages (ArBB and OpenCL), implement on GPUs.