KF Particle Finder on Xeon Phi
Status

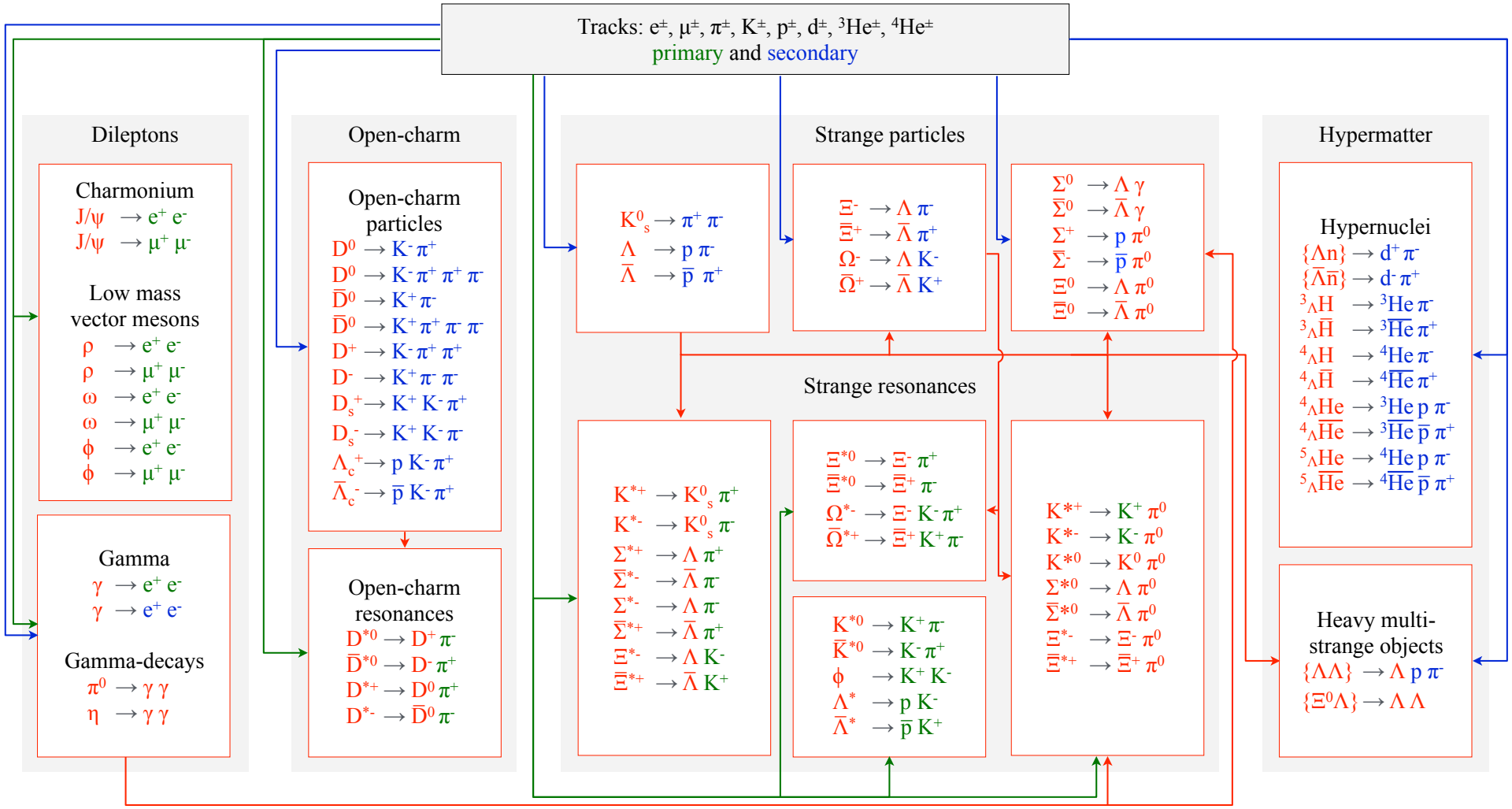
Ivan Kisel[1,2,3], Maksym Zyzak[1,2,3]

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

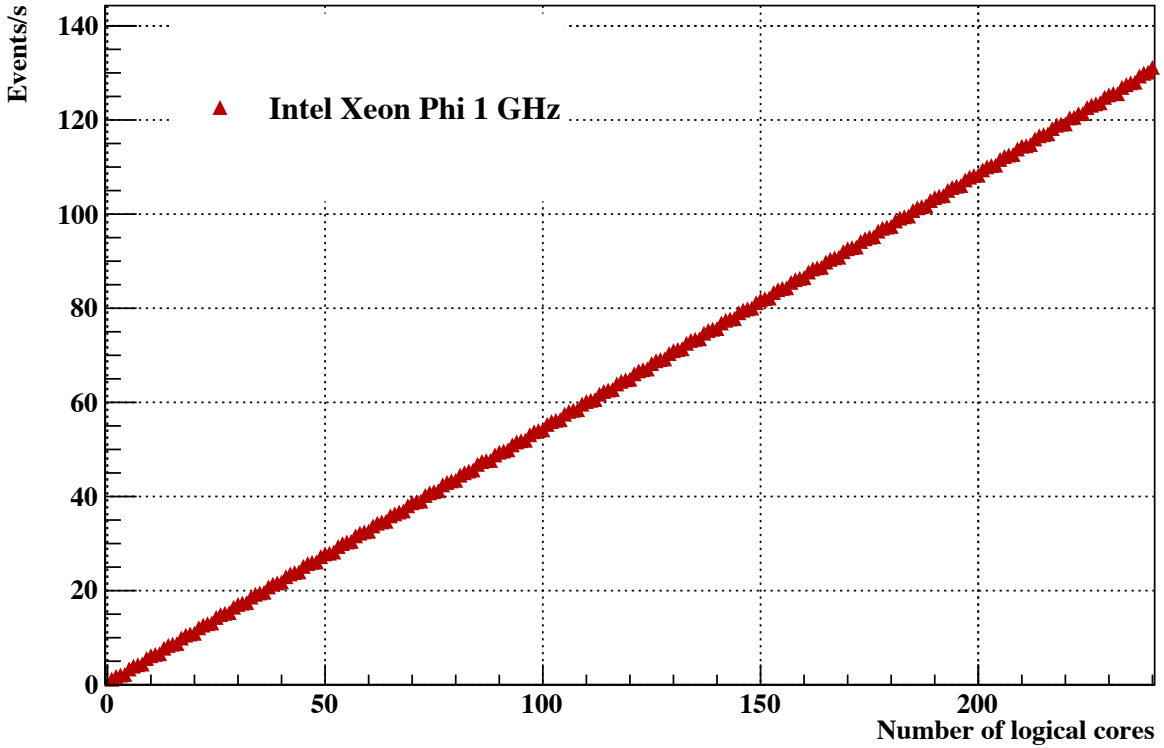
STAR HLT Meeting

BNL, Upton
13.01.2016

KF Particle Finder

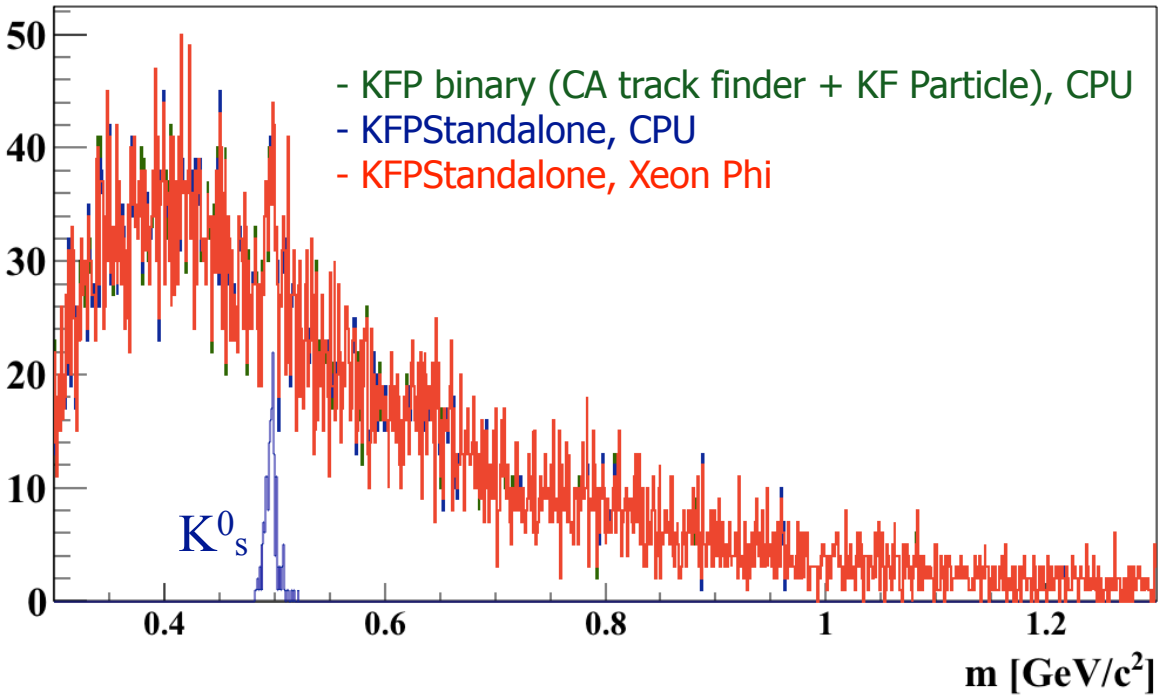


KF Particle Finder on the Xeon Phi



- The full version of KF Particle Finder is adapted for the Xeon Phi card for the native mode.
- 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.
- The throughput of the current version is 130 events/s, which allows to operate with an event rate up to 5 kHz having 40 Xeon Phi cards.

Histograms on the Xeon Phi



Tools for online histograms collection are created:

- Histograms are collected as arrays of bins after KF Particle Finder on the Xeon Phi.
- Histograms are saved into txt files by request.
- Files are read from /srv/michome/ by the KFPStandalonePlotHistograms binary, which plots them and stores to the file KFParticlePerformance.root.
- Histograms collected on the Xeon Phi and on the CPU are the same.
- Current set of histograms includes M, p, p_t, rapidity, χ^2/NDF, x, y, z, r, L for each particle.

Transfer data to the Xeon Phi

Three possibilities for data transfer assuming the native mode scenario:

- copying txt files to the Xeon Phi via scp;
- storing files directly to the Xeon Phi mounted to the host (/srv/michome/);
- offload data and store files directly on the Xeon Phi.

	scp	/srv/michome/	offload
store CPU time, ms	10 + transfer time	10	300

Currently saving files to /srv/michome/ is the fastest data transfer method

Summary and plans

Summary

- The KF Particle Finder package is adapted for running on the Xeon Phi in a native mode.
- The KF Particle Finder scales linearly on the Xeon Phi.
- Tools for histogram collection and plotting are created.
- Currently input tracks are decided to be stored into files to /srv/michome/.

Plans

- Write the KF Particle scheduler for the Xeon Phi, which will read the input data and distribute it between cores.
- Test performance of the scheduler together with the data transfer.
- Organize triggering based on the KF Particle Finder.
- Check physics performance.
- Speed up the code.