

The parallel framework for the partial wave analysis

Tokareva V.A., Denisenko I.I.

DLNP JINR

July 10, 2016

Partial wave analysis

- Hadron spectra are crucial for understanding QCD in the strong coupling regime;
- Partial wave analysis (PWA) is the main tool to extract masses, widths and quantum numbers of hadronic resonances used by LHCb, COMPASS, BESIII, etc.;
- For currently collected datasets PWA applications are often restricted by high demand to CPU resources.

Parallel PWA

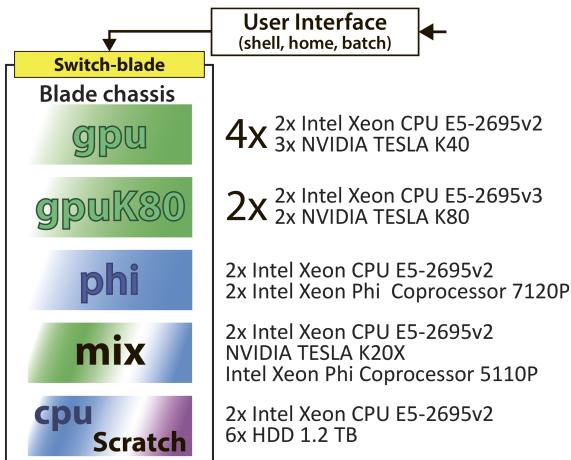
- In the typical set-up PWA can be naturally parallelized. The function minimized in the fit is

$$S = -\ln L = -\sum_i \frac{\sigma^{\text{event}_i}}{\sigma^{\text{norm}}}$$

where σ^{event_i} is the differential cross section for data event i and σ^{norm} is normalization cross section obtained using MC integration.

- Computations for each event are independent.

The structure of the heterogeneous cluster HybriLIT

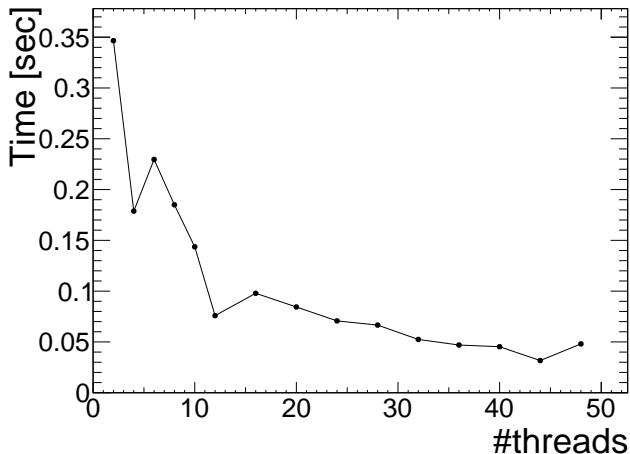


Parallel realizations

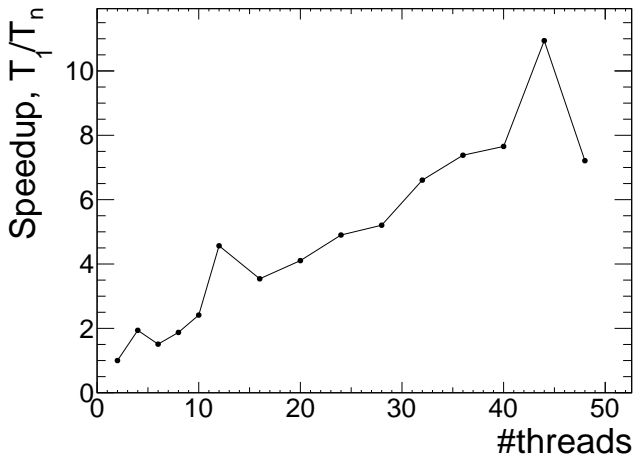
Next algorithm realizations were implemented:

- OpenMP realization;
- MPI realization;
- OpenMP realization with coprocessor extensions for Intel Xeon Phi coprocessors.

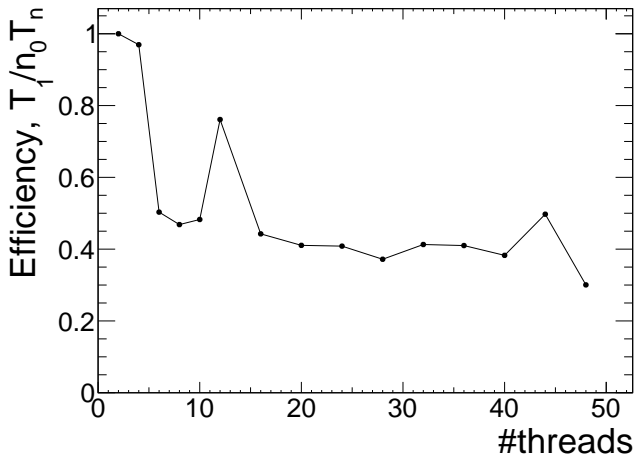
Calculation time for PWA minimization function vs. the number of threads, OpenMP realization



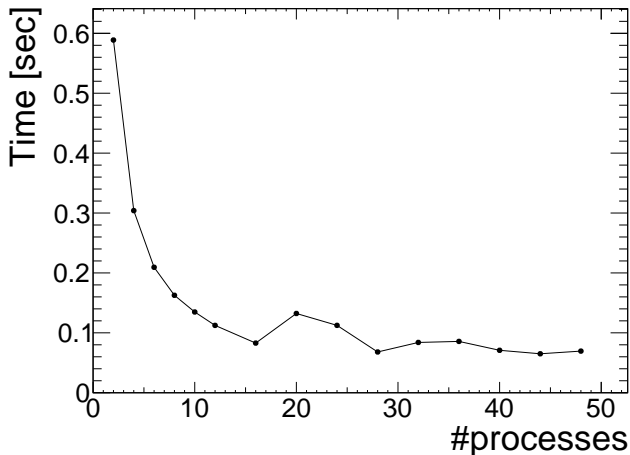
Speedup for PWA minimization function vs. the number of threads, OpenMP realization



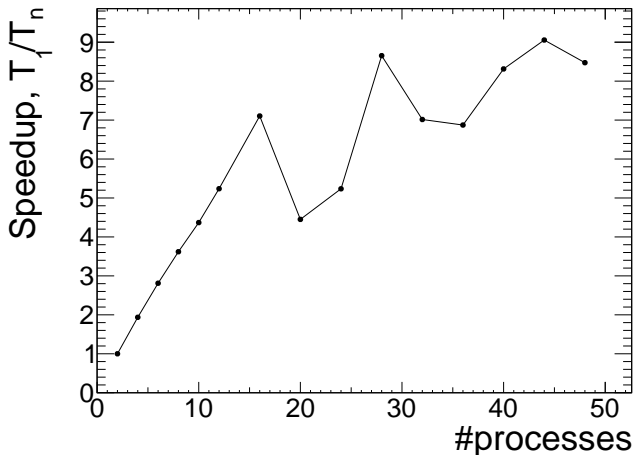
Efficiency for PWA minimization function vs. the number of threads, OpenMP realization



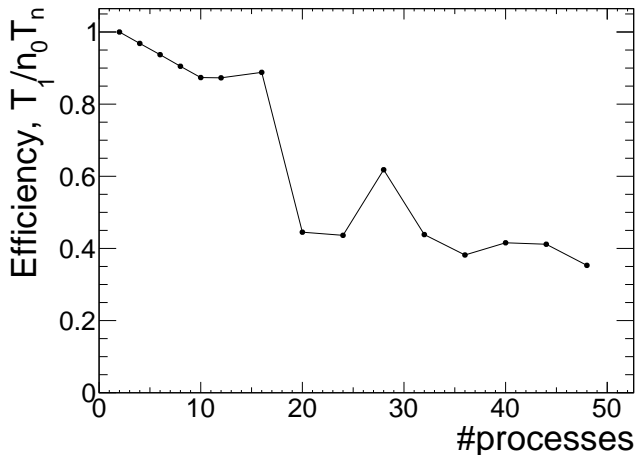
Calculation time for PWA minimization function vs. the number of processes, MPI realization



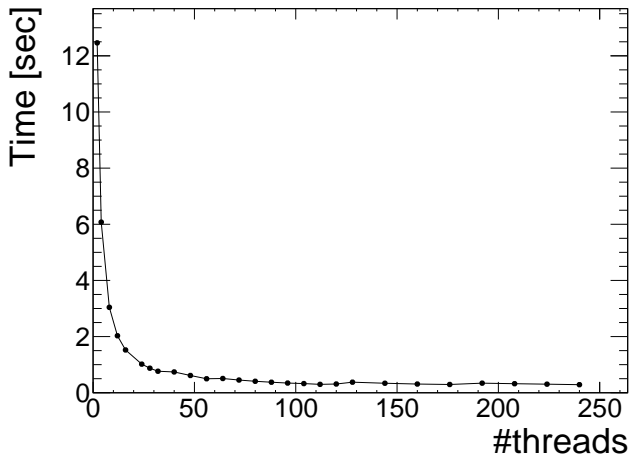
Speedup for PWA minimization function vs. the number of processes, MPI realization



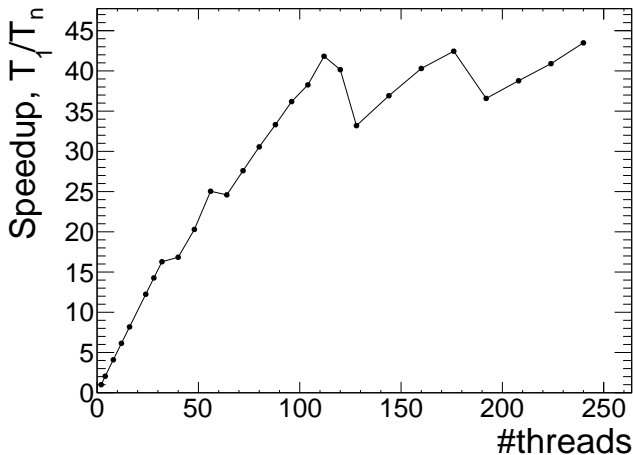
Efficiency for PWA minimization function vs. the number of processes, MPI realization



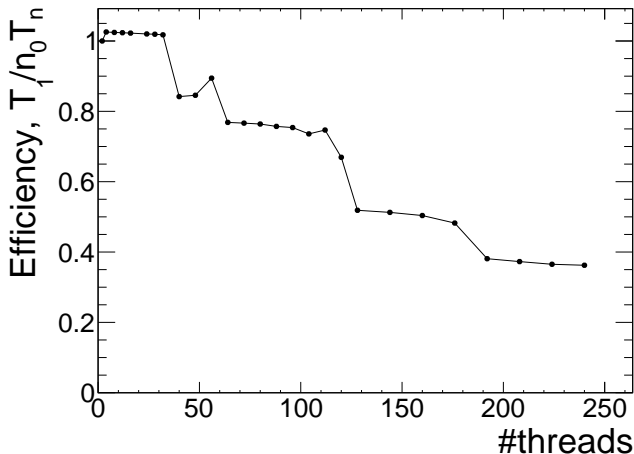
Calculation time for PWA minimization function vs. the number of threads, Xeon Phi realization



Speedup for PWA minimization function vs. the number of threads, Xeon Phi realization



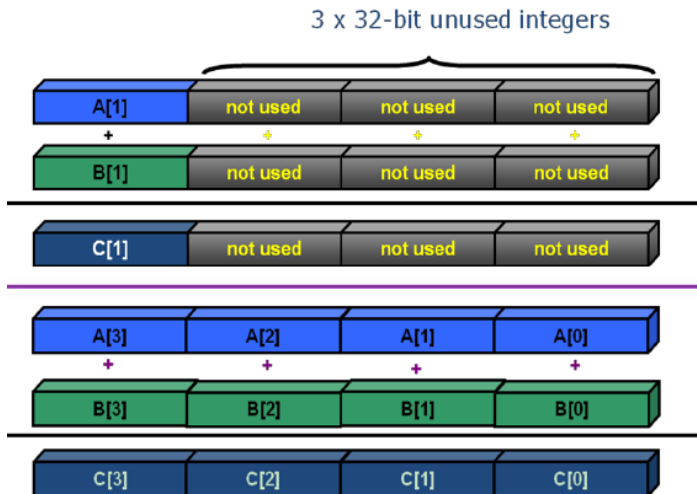
Efficiency for PWA minimization function vs. the number of threads, Xeon Phi realization



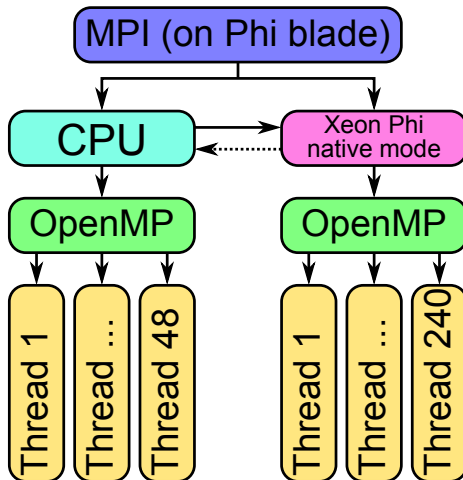
Main results

Technology	Minimum calc. time	Maxumum calc. time	Speedup
OpenMP	0.35	0.03	11
MPI	0.59	0.06	9
OpenMP on Xeon Phi	12.4	0.28	44

Outlook: vectorization



Outlook: advanced architecture



Outlook: user interface

A user-friendly interface is to be implemented

Thanks for attention!

tokareva@jinr.ru

Useful materials

- 1 BES-III experiment, web source — <http://bes3.ihep.ac.cn/>
- 2 Official OpenMP web source — <http://www.openmp.org/>
- 3 D. Gregor, M. Troyer. Boost.MPI, web source — <http://boost.cowic.de/rc/pdf/mpi.pdf>
- 4 Building a Native Application for Intel Xeon Phi Coprocessors, web source — <https://software.intel.com/en-us/articles/building-a-native-application-for-intel-xeon-phi-coprocessors>
- 5 Heterogeneous cluster HybriLIT, official site — <http://hybrilit.jinr.ru/>