Towards parallel track reconstruction with Intel ArBB

I. Kisel[1], *I. Kulakov*[2,3], *H. Pabst*[4], *and M. Zyzak*[2,3]

[1]GSI, Darmstadt, Germany; [2]Goethe University, Frankfurt, Germany; [3]University of Kyiv, Ukraine; [4]Intel, Germany

The Intel Array Building Blocks (ArBB) software [1] is a data-parallel programming environment designed to effectively utilize the power of existing and upcoming throughput-oriented features on modern processor architectures, including Intel's multi-core and many-core platforms. Intel Array Building Blocks provides a generalized vector parallel programming solution that frees application developers from dependencies on particular low-level parallelism mechanisms or hardware architectures. It is comprised of a combination of standard C++ library interface and powerful runtime. It produces scalable, portable, and deterministic parallel implementations from a single high-level source description. It allows to parallelize on both data and task levels in an easy way.

The track reconstruction is a task, which needs a lot of computational power. At the same time all modern high energy physics experiments operate with huge data rates and require fast reconstruction procedures. In order to operate effectively in such difficult conditions the full utilization of CPU is required. Parallel programing is considered now as the only way to utilize the full power of CPU, since all modern CPUs have more than one core and contain a SIMD unit. All the charateristics of ArBB make it perfectly suitable for fast track reconstruction and fitting tasks.

The SIMDized Kalman filter track fitter [2] has been modified using ArBB. Results obtained with the ArBB version have been compared with results of the SIMD version [3]. Tests for time and track fit quality of the algorithm have been performed on a 8-cores computer (GSI lxir039) with two Xeon X5550 processors at 2.7 GHz and 8 MB L3 cache. The computer has 16 logical cores due to the hyper-threading technology.

For time tests two types of calculations using 1 logical core only as well as all 16 logical cores have been performed. The former is to compare the level of the code vectorization using SIMD instructions, the latter in order to compare parallelization level between cores and utilization of the full potential of the CPU. The ArBB version shows practically the same time results, as the SIMD one (see Table).

Table 1: Fitting time per track of the SIMD and ArBB versions executed on one core and all cores

	SIMD		ArBB	
Cores	1	16	1	16
Time, μs	0.42	0.05	0.43	0.06

Residuals and pulls (residuals normalized on the estimated errors) of the estimated track parameters have been calculated at the production vertex in order to test the track fit quality. In Figure 1 residuals and pulls for the track postition x, the track slope in the XZ plane t_x and the inverse particle momentum, signed according to charge, q/p are shown. Obtained results are the same as in the SIMD version.

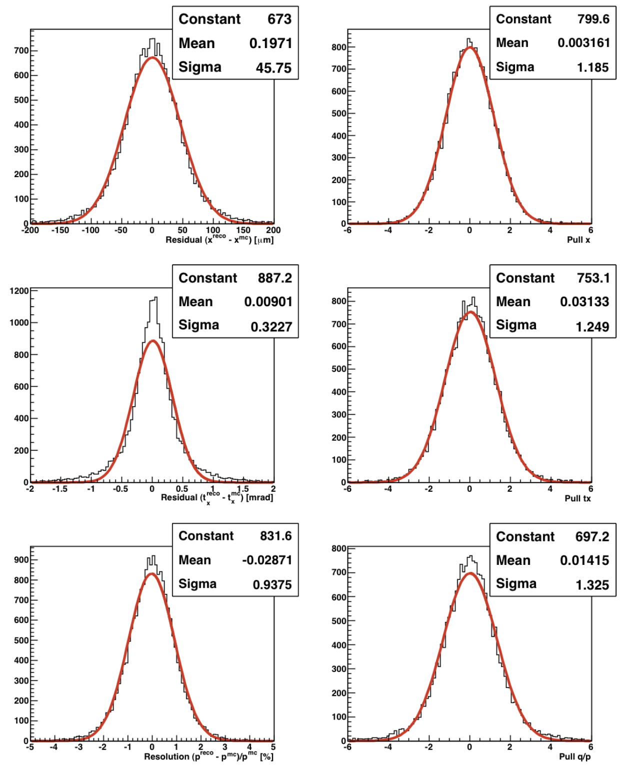

Figure 1: Residuals and pulls of the estimated track parameters calculated with the ArBB version of the Kalman filter track fitter

Subsequently we consider modifications of the Cellular Automaton (CA) based track reconstruction algorithm using ArBB.

References

[1] http://software.intel.com/en-us/articles/intel-array-building-blocks.

[2] S. Gorbunov *et al.*, Comp. Phys. Comm. **178** (2008) 374

[3] I. Kisel, M. Kretz and I. Kulakov, *CBM Progress Report 2009*, Darmstadt 2010, p. 75