THE SPEEDUP ANALYSIS IN GEM DETECTOR BASED ACQUISITION SYSTEM ALGORITHMS WITH CPU AND PCIE CARDS*

Rafal D. Krawczyk, Pawel Linczuk, Piotr Kolasinski
Andrzej Wojenski, Grzegorz Kasprowicz, Krzysztof Pozniak
Ryszard Romaniuk, Wojciech Zabolotny, Pawel Zienkiewicz

Institute of Electronics Systems, Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warszawa, Poland

Tomasz Czarski, Maryna Chernyshova

Institute of Plasma Physics and Laser Microfusion
Hery 23, 01-497 Warszawa, Poland

(Received July 11, 2016)

The demand for fast software post-processing algorithm in GEM detector based acquisition systems resulted in investigating potential algorithmic, hardware and software solutions to achieve highest throughput and lowest latencies in post-processing of the acquired data. The overview of solutions using Intel CPU and PCIe Intel and NVIDIA cards is presented. The feasibility analysis in terms of using the devices in the implemented systems is introduced including the trends of hardware development.

DOI:10.5506/APhysPolBSupp.9.257

1. Introduction

The GEM detector based acquisition systems are the compound multimodule electronic devices whose function is to provide quick acquisition, storing and processing of the data from the detector. The systems are described in [1, 2]. The scheme of main elements of the system is discussed in [3, 4]. The primary requirement for the systems of such kind is to provide highest throughput possible when processing the data in the plasma physics

* Presented at the NICA Days 2015 Conference associated with WPCF 2015: XI Workshop on Particle Correlations and Femtoscopy, Warszawa, Poland, November 3–7, 2015.

experiments. In the legacy system, which is currently used in various tokamak experiments [2], the data is acquired and then processed off-line. This part is done by algorithms called the post-processing, consisting of sorting and merging the acquired data, and then creating histograms. There is considered also the implementation of tomography algorithms using the calculated histograms in further development. The post-processing is so far performed in MATLAB. There is currently implemented an improved version of the system that allows acquiring the data with higher frequencies and higher resolution in comparison with the previous version of the system [5]. Subsequently, there are demands for increasing the spatial and temporal resolution of the systems along with their functionality understood as performing complex data analysis as fast as possible [6]. The primary objective is to adapt the algorithms to use an appropriate hardware resulting in increased throughput.

2. The optimization of post-processing algorithms

There were introduced mechanisms to improve the efficiency of aggregating and transmitting the data [4, 7]. The algorithms are discussed in detail in [8, 9]. Regarding the post-processing algorithms, first step of optimization was using the mex functions instead of MATLAB scripts. It resulted in drastic increase of performance, which was discussed in [9]. However, further speedup was in demand. Therefore, more aggressive methods of optimization were needed. In [10], there was given an overview of contemporary technologies and algorithmic approaches that could potentially be used with post-processing algorithms. There were, however, still potential possibilities to investigate. Investigation needs to be done to compare actual scientific and HPC solutions [11]. There were several categories to consider. For GPGPU family, the NVIDIA cards with CUDA technology was chosen, Intel Xeon Phi was chosen as multicore coprocessor. For the CPUs, the Intel Xeon family was chosen as representative.

3. The overview of hardware and software support for enhancing the performance of computations

At the present stage of the development of the system, the post-processing is performed on the PC. In such approach, the throughput of computations can be maximized when utilizing all cores of multi-core Intel Xeon Processors and by using the PCIe cards such as NVIDIA or Intel Xeon Phi (also called MIC). Either of these solutions allow high scalability of implementations. NVIDIA allows a wide choice of GPGPUs along with computation — dedicated TESLA cards: K20, K40 and K80. When using Intel coprocessors, one should consider both Knights Corner units (3100, 5100 and 7100

series) as well as announced Knights Landing architecture, whose units will be available as both PCIe cards and self-booting sockets. The latter solution has several advantages over PCIe card, such as more available PCIe lanes (36 as opposed to 16) and binary compatibility with Intel Xeon Processor [12]. Moreover, several major improvements over Knights Corner solution are claimed that can significantly increase performance.

The Intel Xeon Phi when used with Intel Xeon processors allows to perform computations both on processor and coprocessor in parallel. The provided off-load mechanism enables to send the data to coprocessor using the processor and to overlap the computations on both units. To achieve this, one must provide efficient means of dividing computation between CPU and MIC. Such mechanisms are available for OpenMP and Intel Cilk Plus. The equivalent mechanism is also provided by NVIDIA, by implementing host side (CPU) code and device side (GPGPU) kernels in CUDA [13]. The CUDA allows to overlap computations between CPU and GPGPU. The CUDA-based GPGPUs have different capabilities when compared with Intel Xeon Phi due to differences of architectures of both units. Its efficiency allows for higher speedups in a selection of highly parallel applications [14]. Architecture of NVIDIAs is a SIMT (NVIDIA SIMD variant) device (as opposed to MIMD Intel Xeon Phi). To fully utilize the computing capabilities of NVIDIA cards, the implementation should have a sufficient number of threads to utilize execution in warps and to hide latencies of memory access. The recursion is limited in comparison with Intel cards. Possibility of utilizing a pipeline within a single card can be limited and divided, and conquer algorithms with different data sets and multiple branches can strongly hamper the efficiency.

When using PCIe cards, to achieve significant speedup, algorithm must have a high amount of computations with sufficient amount of data to process in order to hide latency of data transfers over PCIe. Moreover, in typical optimization, the on-device memory hierarchy must be considered. PCIe cards are not universal solutions. Insufficiently comprehensive problems or developers hardware unfamiliarity can render implementation inefficient. With preliminary testing performed on Intel Xeon Phi 31s1p, it was easily observable. While investigating common patterns of embarrassingly parallel algorithms, such as parallelizing for loop with single SAXPY operation, the comparable execution times were achieved on either units (when both CPU and MIC units were utilizing parallelism and hardware-oriented program with compiler optimization options were used). When the data was reused, however, above 50 times speedup was achieved for stencil algorithm, confirming the results given in [15] (due to high cache reuse). When investigating various examples of using the devices in HEP experiments, in most cases when optimization techniques were used, the efficiency of optimized

algorithms for single Intel Xeon Phi were comparable with dual-socket Intel Xeon CPUs [16]. When considering either of MIC or Xeon CPU for their solution, one must analyze the architectural differences of cores of the devices *e.g.* their number (higher in MIC), clock rate (higher in Intel Xeon), size of cache and its latency. Moreover, one must consider the latencies which are involved with transferring the data between CPU and MIC or between CPU and GPGPU via PCIe. For Knights Corner Intel Xeon Phi, the bandwidth of PCIe 2.0 $\times$ 16 is 8 GB/s in either direction. For Knights Landing coprocessor PCIe card, the PCIe 3.0 $\times$ 16 allows for transferring data at theoretical rate of 16 GB/s in either direction. The memory is restricted to on-board DRAM global memory (up to 16 GB for Intel Xeon Phi and up to 24 GB for NVIDIA cards at the time of writing this paper) [17, 18]. To utilize maximum of computing capabilities of Intel Xeon Phi, the algorithm must be highly thread-parallelizable and highly vectorizable to make most of 512-bit vector register. To maximize performance of Intel Xeon Phi, one must use at least two hardware threads per core [15]. Moreover, once read, data that resides in cache should be reused before being evicted. As a result, due memory access patterns must be followed. To utilize vector unit as much as possible, alignment of data, tiling and padding must be imposed. Adequate data structures to allow accessing continuous blocks of data is advised. Similar problems are faced when using NVIDIA cards. The transfer via PCIe is also necessary. Moreover, the SIMT architecture imposes using large number of threads, which can significantly reduce efficiency of some algorithms, especially when accessing irregular data in the GEM detector based acquisition systems.

One should also consider the parallel capabilities of Intel Xeon processors. Currently available Intel CPUs offer parallelism with a mechanisms of hardware optimization of sequential algorithms [15]. Contemporary processors have up to 18 cores and up to 36 logical threads. The 256-bit vector registers are available [19]. Xeon processors are a compromise for applications where introducing parallelism can render limited speedup. The parallelism of Xeon processors can be appropriate, especially when computation on the card and copying data from and to CPU can render poor results due to low compute to data transfer ratio, poor data reuse or not parallelizable algorithm. Moreover, the MIMD architecture of CPU (just like the architecture of MIC) allows to run different programs on each core, thereby allowing to easily implement pipeline or extending the possibilities of dividing work among the cores than in the case of GPGPU.

Actual HPC solutions provide a way to create pipelines over PCIe cards, where calculations from HEP hardware ends on CPU with omitting its RAM with RDMA Transfers. NVIDIA offers a GPUDirect technology [20] and for Xeon Phi, there is the OFED implementation. This allows to lower the la-

tency of whole software post-processing but exact implementation and usage must be further investigated for the specific issue. NVIDIA GPGPUs, Intel coprocessors and Intel processors have been investigated as representatives — other devices, such as AMD processors or AMD GPU cards and have not been the object of this study.

4. Feasibility analysis of using CPU and PCIe cards in algorithms of the GEM detector based acquisition systems

Post-processing algorithms have been analyzed in terms of assessing the achievable speedup when parallelized. Whereas speedup has been achieved for histogramming [10] when parallelizing on CPU, the merging and sorting is poorly parallelizable. When regarding merging and sorting, the available solutions [13, 21] impose increasing number of computations and do not fit irregular data types of the system. Either of the problem is memory-bound with low ratio of computation *versus* memory access. However, several methods to increase the memory reuse are to consider. First of all, one can merge both algorithms, thus enhancing data locality as the reorganized words can be directly used in histogramming. Second of all, there is a possibility of treating the events occurring for single board and for several boards independently and, following this pattern, to divide the problem into sub-problems. Also one should consider of tomography algorithms to be implemented. The tomographic reconstruction will be performed after collecting information on histograms and using the data for 3D reconstruction. Algorithms of such kind focus on solving linear equations are parallelizable, and have numerous methods and patterns of introducing parallelism.

5. Conclusions

The preliminary tests allowed to assess details of different contemporary technologies in terms of increasing computation throughput thereby achieving higher speedup when considering post-processing algorithms in GEM detector based acquisition systems. The presented issues are strongly associated with High Performance Computing (HPC). The further study is necessary to assess which among presented technologies allows to maximize the performance of selection of algorithms. Moreover, the research can lead to conclusions whether or not use the already-announced Knights Landing Intel processors architecture that will be available as either a coprocessor card or an alternative or supplement for a CPU. Investigating performance on NVIDIA card or other GPGPU is to consider in further study. Depending on the achieved speedup, the scalability, memory access rates and other factors of importance when considering PCIe cards or CPUs, due device with higher computation capability can be chosen to achieve further speedup.

REFERENCES

[1] G. Kasprowicz *et al.*, *Proc. SPIE* **8454**, 84540M (2012).

[2] M. Chernyshova *et al.*, *JINST* **9**, C03003 (2014).

[3] T. Czarski *et al.*, *Proc. SPIE* **9662**, 96622W (2015).

[4] P. Kolasinski *et al.*, *Proc. SPIE* **9662**, 96622J (2015).

[5] A. Wojeński *et al.*, *Nucl. Instrum. Methods Phys. Res. B* **364**, 49 (2015).

[6] A. Wojeński *et al.*, *Proc. SPIE* **9662**, 96622K (2015).

[7] P. Kolasinski *et al.*, *Proc. SPIE* **9290**, 92902H (2014).

[8] T. Czarski *et al.*, *JINST* **10**, P10013 (2015).

[9] R.D. Krawczyk *et al.*, *Proc. SPIE* **9662**, 96622M (2015).

[10] R.D. Krawczyk *et al.*, *Proc. SPIE* **9662**, 96622N (2015).

[11] `https://www.top500.org/`, Top500 List — November 2015.

[12] A. Sodani, Intel Xeon Phi Processor Knights Landing Architectural Overview.

[13] S. Cook, *CUDA Programming: A Developer's Guide to Parallel Computing with GPUs*, Elsevier, 2013.

[14] NVIDIA Official Site: Intel Xeon Phi: Just the Facts, `http://www.nvidia.com/object/justthefacts.html`

[15] J. Jeffers, J. Reinders, *Intel Xeon Phi Coprocessor High-Performance Programming*, Elsevier, 2013.

[16] A. Nowak *et al.*, *J. Phys.: Conf. Ser.* **513**, 052024 (2014).

[17] NVIDIA Official Site: Tesla Server GPUs, `http://www.nvidia.co.uk/object/tesla-server-gpus-uk.html`

[18] Intel Official Site: Intel® Xeon Phi™ Coprocessors, `http://ark.intel.com/products/family/71840/Intel-Xeon-Phi-Coprocessors`

[19] Intel Official Site: Intel Xeon E7 v3 family overview, `http://ark.intel.com/pl/products/family/78585/Intel-Xeon-Processor-E7-v3-Family`

[20] NVIDIA Official Site: NVIDIA GPUDirect, `https://developer.nvidia.com/gpudirect`

[21] M. McCool, J. Reinders, A. Robison, *Structured Parallel Programming. Patterns for Efficient Computation*, Elsevier, 2012.