jet INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2017, VOL. 63, NO. 3, PP. 323-328
Manuscript received June 27, 2017; revised July, 2017. DOI: 10.1515/eletel-2017-0043

Feasibility of FPGA to HPC computation migration of plasma impurities diagnostic algorithms

Pawel Linczuk, Rafal D. Krawczyk, Wojciech Zabolotny, Andrzej Wojenski, Piotr Kolasinski, Krzysztof T. Pozniak, Grzegorz Kasprowicz, Maryna Chernyshova and Tomasz Czarski

***Abstract*—We present a feasibility study of fast events parameters estimation algorithms regarding their execution time. It is the first stage of procedure used on data gathered from gas electron multiplier (GEM) detector for diagnostic of plasma impurities. Measured execution times are estimates of achievable times for future and more complex algorithms. The work covers usage of Intel Xeon and Intel Xeon Phi - high-performance computing (HPC) devices as a possible replacement for FPGA with highlighted advantages and disadvantages. Results show that less than 10 ms feedback loop can be obtained with the usage of** 25% **hardware resources in Intel Xeon or** 10% **resources in Intel Xeon Phi which leaves space for future increase of algorithms complexity. Moreover, this work contains a simplified overview of basic problems in actual measurement systems for diagnostic of plasma impurities, and emerging trends in developed solutions.**

***Index Terms*—plasma diagnostic, GEM system, feedback loop, Intel Xeon, Intel Xeon Phi, high-performance computing (HPC)**

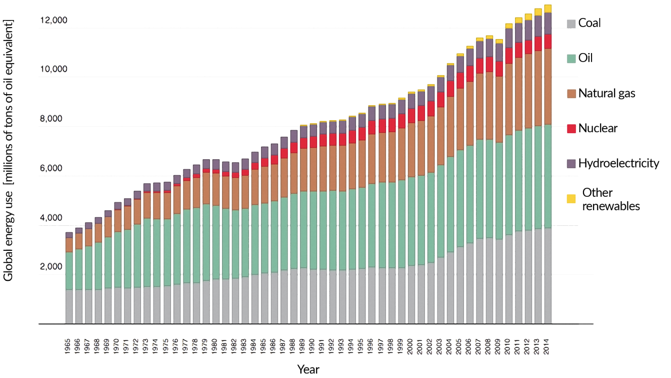

Fig. 1. Accumulated energy usage over years with appointed component fuels[1].

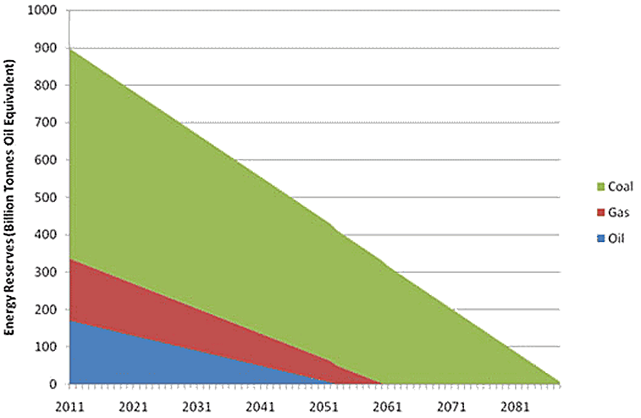

Fig. 2. Estimated time of earth deposits of main fossil fuels[2].

I. Introduction

MANKIND civilization development is significantly based on introduction and deployment of new technologies. They provide better tools and devices utilization and transportation efficiency. However, most of them rely on fossil fuels (about 85%, presented in figure 1) which deposits can exhaust soon.

If their usage does not stop growing (presented in figure 1), Earth deposits of this materials will end in a matter of about 100 years (presented in figure 2). Renewal of deposits would require million years.

The fossil fuels generate a high amount of CO_2 that is detrimental to the environment. Their availability is also highly dependent on geographical location. Described problems lead to research for other less constrained and harmful sources of energy.

Much of mankind effort concentrates on renewable energy, which consists of energy extraction from water, wind, light, biomass, etc. Those solutions are acceptable in terms of environmental pollution, but occupy much space (e.g., solar power plants), interfere with the landscape (e.g., wind power plants) or cannot continuously produce power (e.g., due to lack of wind).

Nuclear energy can solve partially the mentioned above problems. The side effect of its usage is a big amount of radioactive waste, which half-life can reach millions of years. Aside from that, nuclear fuel is also a fossil fuel, and its deposits are limited. It is possible to reuse the nuclear waste again as a fuel, but such solutions are much less efficient. Due to nuclear plants malfunctions in the past, it is a controversial and on some level dangerous solution.

This scientific work was partly supported by Polish Ministry of Science and Higher Education within the framework of the scientific financial resources in the years 2014-2017 allocated for the realization of the international co-financed project.

Pawel Linczuk, Rafal D. Krawczyk, Wojciech Zabolotny, Andrzej Wojenski, Piotr Kolasinski, Krzysztof T. Pozniak and Grzegorz Kasprowicz are with University of Technology, Institute of Electronic System, Warsaw, Poland (e-mail: p.linczuk@stud.elka.pw.edu.pl).

Pawel Linczuk, Maryna Chernyshova and Tomasz Czarski are with Institute of Plasma Physics and Laser Microfusion, Warsaw, Poland.

Based on described solutions, on the current level of technology, mankind do not have a way to obtain energy without negative side effects.

One of the proposed solutions is obtaining energy from thermonuclear fusion. It is based currently on deuterium and tritium fusion into helium, described in figure 3. Radioactively activated materials, used for thermonuclear fusion devices, remain radioactive for less than 100 years, what overcomes radioactive waste disposal. Due to reaction details, malfunctions are not so dangerous and result in slowing down and stopping the whole process. Fuel deposits are sufficient for millions of years and are not constrained geographically, what provides a solution of the previously described problems.

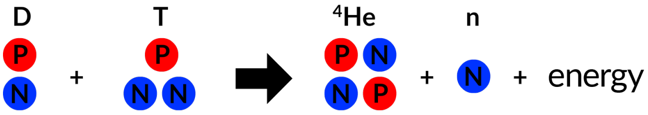

Fig. 3. Simplified deuterium and tritium thermonuclear fusion. In order to enable this reaction Lawson criteria must be fulfilled.

However, thermonuclear fusion is a complex process. One of developed areas is magnetic confinement of plasma. This can be implemented by usage of tokamak device.

To maximize experiment value and efficiency, fast-computation and high-throughput measurement systems have to be used. While the experiment is successful, new clean energy source will emerge in everyday life.

II. Technical problems

The increase of the throughput and accuracy of measurement systems relies on the usage of more and more complex components. In order to reach needed computational power, the newest technologies, techniques and devices have to be used in development.

The previously developed system relied heavily on FPGA calculations due to its highly parallel characteristic with high throughput combined with low latency. Both signal acquisition and data processing were done with this technology. However, such a solution is difficult to be improved due to implementation time and FPGA architecture limitations. The processing part of measurement systems needs to perform complex, mostly numerical algorithms, where the mentioned technology may be replaced with more convenient solutions. Currently, the High-Performance Computing (HPC) segment relies mostly on General-Purpose Computing on Graphics Processing Units (GPGPU), multi-core CPUs and many-core processors/coprocessors (e.g. Intel Xeon Phi). Those computational capabilities offered by those solutions, along with fast communication channels can meet latency limits needed in the described systems. However, there is a tradeoff between FPGA with very low latency, and HPC with lower implementation time and higher amount of available resources.

Time constraints emerge from the need to control dynamically changing experiments such as hot plasma diagnostics. Previous works [3] consisted of fast and low-level implementation of impulse parameters estimation on Intel devices (Xeon and Xeon Phi) along with execution time optimization (presented in figure 8). Implementation of the required algorithm in the GEM measurement system, as a part of hot plasma diagnostics, with near real-time feedback (assumed maximal 10 ms of Round-trip time - RTT), requires maximal hardware utilization. This article concentrates on a feasibility study of usage Intel HPC devices as computational part of the described architecture.

III. GEM system

Currently, the developed system [4] is built to detect plasma impurities in a tokamak during an experiment. Plasma, however, trapped in magnetic confinement, generates its own magnetic field. Superposition of both results in unwanted plasma movement in space. During contact with tokamak housing plasma gathers its atoms e.g., tungsten which is chosen as divertor coverage material in the WEST experiment. By usage of GEM detector the x-ray radiation generated by impurities in plasma can be measured. The measurement requires energy histogramming which is done in software. The architectural design of the system is shown in figure 4.

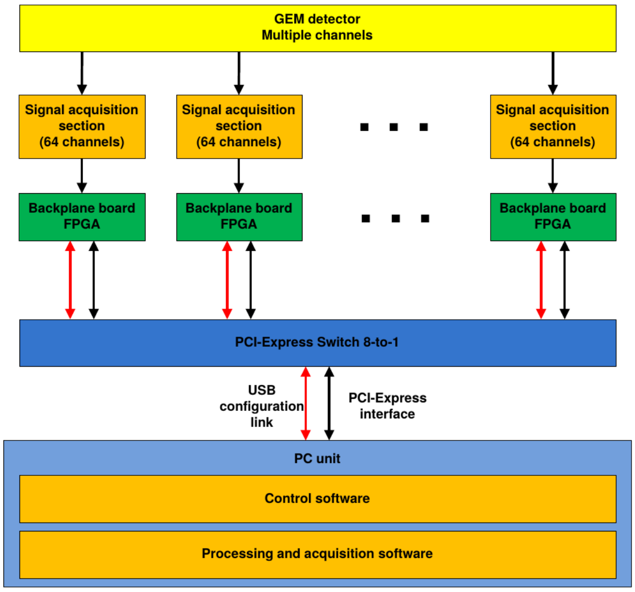
GEM detector
Multiple channels
Signal acquisition section (64 channels)
Signal acquisition section (64 channels)
Signal acquisition section (64 channels)
Backplane board FPGA
Backplane board FPGA
Backplane board FPGA
PCI-Express Switch 8-to-1
USB configuration link
PCI-Express interface
PC unit
Control software
Processing and acquisition software

Fig. 4. Architectural overview of developed system[4].

The logical structure of the GEM measurement system is shown in figure 5. Its purpose is to calculate pulse parameters (charge, time, duration, channel) from raw input signal samples. The next step is identification of clusters [5] (finding pulses generated by a single photon on the detector readout and combining them into a single event). Both mentioned steps are represented as Q part in figure 5 and are later used to calculate the histogram of energy distribution (H part in figure 5). Those algorithms create the ”calculations part” and are described in [6]. In the proposed system the acquisition part is done in FPGA (DR in figure 5). In the previous implementation

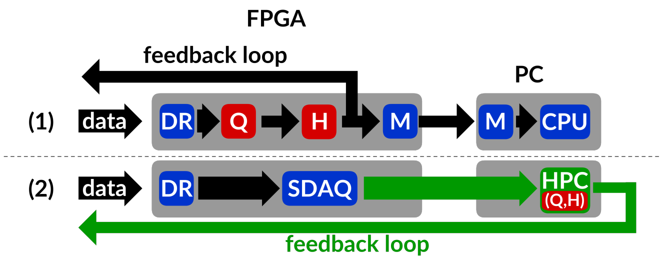

Fig. 5. Legacy system with optimization proposition. Migration of calculation part create additional latency and communication overhead. Along with such change new algorithmic possibilities emerge.

the ”calculations part” was solely done in FPGA along with feedback. Such solution was sufficiently efficient in terms of time constraints but faced limitations described in section II. Most works during the optimization were done on the feedback part, where the previous system used an inefficient buffering (part M in figure 5) to store the measurements data and process them offline, regardless of experiment duration.

IV. System architecture

The proposed solution migrates the processing part from FPGA to PC (presented in figure 5). To efficiently process data in the computational part the acquired data have to be serialized (Serial Data Acquisition - SDAQ part in figure 5).

The calculation of pulse parameters is an embarrassingly parallel problem, which can be easily divided between available computation threads. The optimal value of *m* and chunk sizes (presented in figures 10 and 12) have to be found in order to fully utilize hardware.

The actual system throughput can be calculated approximately as:

$$th = ch \cdot b \cdot d \cdot f_{eff}$$

$$th = 64 \cdot 16 \cdot 40 \cdot 500,000 \approx 19,07\ Gbit/s \approx 2,38GB/s$$

where *ch* is the number of acquisition channels, *b* - the size of a single sample in bits, *d* - the size of acquired sample frames, f_{eff} - the empirical value of maximum currently possible acquisition rate. Currently the throughput is about 32 milion pulses per second.

The communication with PC should minimize the buffer usage and use efficient data copy techniques. Currently, the communication is implemented with PCIe Gen 2.0 due to the requirements of the Xeon Phi used in research. However, the theoretically available bandwidth is sufficient. At 32 Gbit/s throughput[8], 8 lanes will be utilized in 60%[9], which is acceptable. Figure 9 shows usage of a single Direct Memory Access (DMA) core implementation which utilizes fully 8x of 16x available lines. Similar setup, as described in [9], will be considered in the improved system.

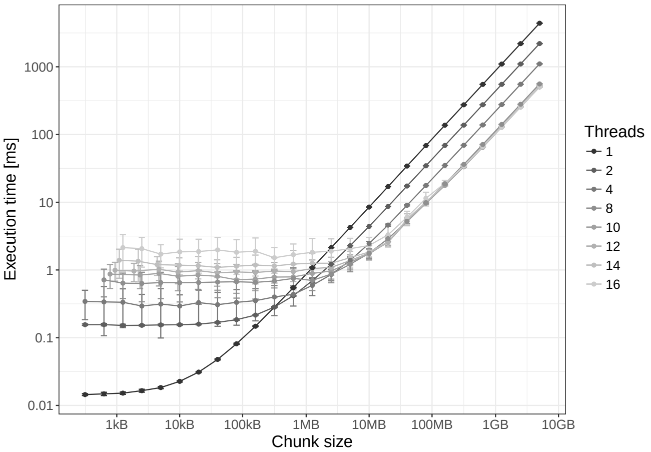

Fig. 6. Execution time comparison in terms of chunk size and number of threads used in Xeon CPU.

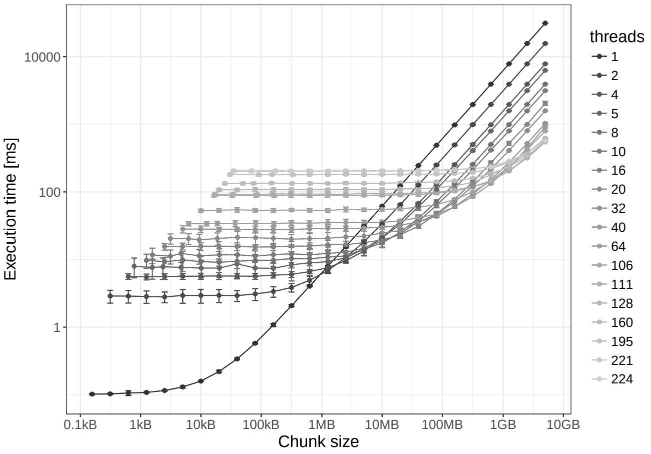

Fig. 7. Execution time comparison in terms of chunk size and number of threads used in Xeon Phi coprocessor (with ECC disabled).

V. Implementation

The feasibility study evaluated the algorithm prepared and implemented during previous works [3]. The data used in the test was a single valid pulse, along with six noise pulses, hardware-acquired [3] (presented in figure 11), and multiplicated to meet the chunk size of a given test.

The research was done using one Xeon E5-2630 v4 CPU[10] (consisting of 10 cores, able to run 20 threads) and one Intel Xeon Phi 31S1P[11] (Knights Corner, 57 cores, 228 fully-fledged hardware threads) as coprocessor. The motherboard used was the Intel’s S2600CW model[12].

The tests consisted of the execution of code using OpenMP 4.0, as part of Intel Parallel Studio, with a different number of threads. The work was divided between cores with *balanced* granulation or adequate type [13].

The Intel toolchain was used, included in Intel Parallel Studio XE 2016 with MPSS 3.7.2, run on CentOS 7.2 GNU/Linux distribution with stock kernel 3.10.0-327 version.

Execution times were measured by the use of *clock_gettime* function and heavily profiled with the use of VTune Amplifier

Xeon Phi KNL generation cards/processors supports PCIe Gen 3.0[7]

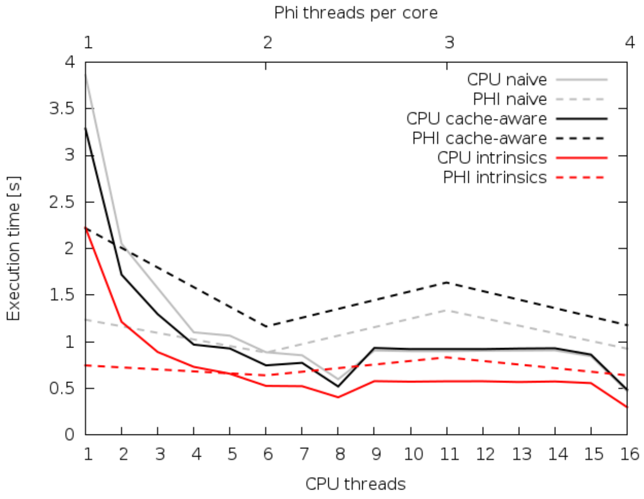

Fig. 8. Speedup comparison of different implementations[3]. Amount of test data reached 6GB, which were about 80 mln pulses. Xeon Phi results are presented with thread per core utilization and Xeon with number of used threads. In figure both intrinsics versions reached best results with Xeon full utilized CPU reaching time of 300ms. Measured time consists only of calculation time.

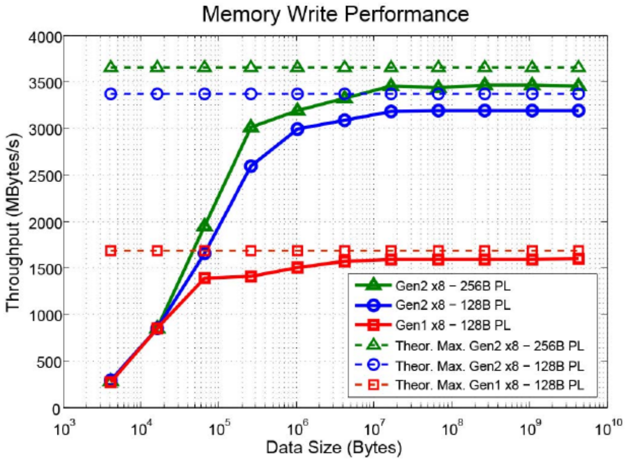

Fig. 9. Throughput comparision obtained by single DMA controller[9] in FPGA to PC by PCIe x8 communication lanes.

and usage of the intrinsics code. Intrinsics are the functions translated by compiler almost one to one to low level assembly code in order to maximize efficiency of algorithms and make possibility of use features not available otherwise.

VI. Choosing optimal thread number

Based on the figures 6 and 7, the number of threads for the fastest execution time can be calculated for various chunk sizes. However, the speedup percentage can vary between different chunk sizes, so the calculation of the appropriate number of threads should not only rely on given tables I and II but also on a speed-up percentage for the neighboring interval.

The research below does not include the memory access latency, which can be observed during future implementation testing. The used algorithm is concentrated on avoiding memory cache issues (eg. cache coherence, false sharing) with the usage of non-temporal read and write operations. A real-time kernel workaround should be considered for the implementation of the feasibility study in the full feedback loop. The purpose of this research is to estimate the minimal latency induced by threads during calculations and to determine directions of further research.

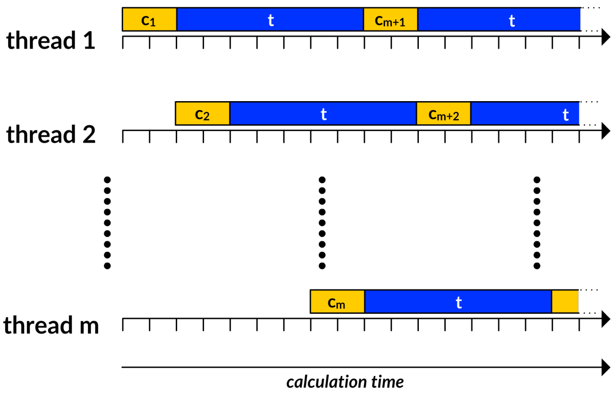

Fig. 10. Calculations described in figure 12 in time dimension. c describe time of sending data chunk and t as calculation time. Approriate m must be chosen to make threads fully utilized.

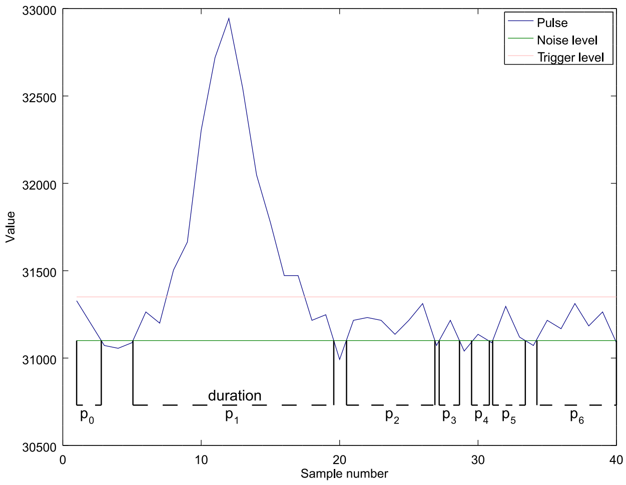

Fig. 11. Test pulse used during research[3] with highlighted trigger, noise level and described 7 pulses, only one of which (p_1) is valid. Output of algorithm should return only p_1 parameters estimation.

Table I
Xeon optimal chunk size

Data amount [kB]	Thread number
0-321	1
321-750	2
750-2,715	4
2,715-8,500	8
8,500-15,000	10
15,000-25,400	12
25,400 and above	10

Table II
XEON PHI OPTIMAL CHUNK SIZE

Data amount [MB]	Thread number
0-0.91	1
0.91-1.74	2
1.74-6.04	4
6.04-14.75	8
14.75-63.63	16
63.63-157	32
157-824	40
824-1,300	106
1,300 and above	111

VII. CALCULATION EFFICIENCY

In order to optimize communication to computation ratio, the calculations must be done in parallel. The data path is presented in figure 12. Based on results from [9] presented in figure 9, to fulfill the throughput requirements, it is necessary to use chunks not smaller than 200kB. The chosen size induces smaller latency than the larger ones, and ensures better CPU utilization, leaving more unused execution time available for future algorithm extensions. Regarding architecture proposed in figure 12, the execution periods of individual threads will be overlapped, as shown in figure 10.

Table III
OPTIMAL TIMES FOR 200 KB SIZE OF CHUNKS WITH SPEEDUP TO MORE THREADS NUMBER PER DEVICE

Type	Thread number	Computation time	Speedup
Xeon CPU	1	0.23ms	157%
Xeon Phi	1	1.33ms	262%

Using the data presented in figure 9, we may determine the speed of chunk processing. With 200 kB chunks, it is possible to achieve 3000 MB/s of data transfer. As 3000MB=15360·200 kB, the processing time of a single package is 1s/15360=0.065 ms. The execution times are presented in Table III.

With such overlapping, based on execution times, number of threads needed to run in feedback mode can be calculated (see figure 10). With those results, it is possible to estimate the idle time of available resources. The results are presented in Table IV.

Table IV
NUMBER OF THREADS NEEDED TO WORK IN FEEDBACK MODE WITH DEVICE USAGE

Type	Threads in feedback	Device usage
Xeon CPU	4	from 25% 2/core up to 50% 1/core
Xeon Phi	21	from 10% 4/core up to 19% 2/core

The currently used S2600CW motherboard is compatible with Xeon Phi 31S1P[14]. It provides 2 CPU sockets and can be used in a typical configuration with 2 Xeon Phi devices. In this configuration, the estimated device usage needed to fulfill time requirements, will be 12.5% to 25% for CPUs and 5% to 10% for Xeon Phi, which results in 9% to 18% average system usage. The further tests may be done using other motherboards with 4 and more CPUs but Xeon Phi compatibility have to be considered.

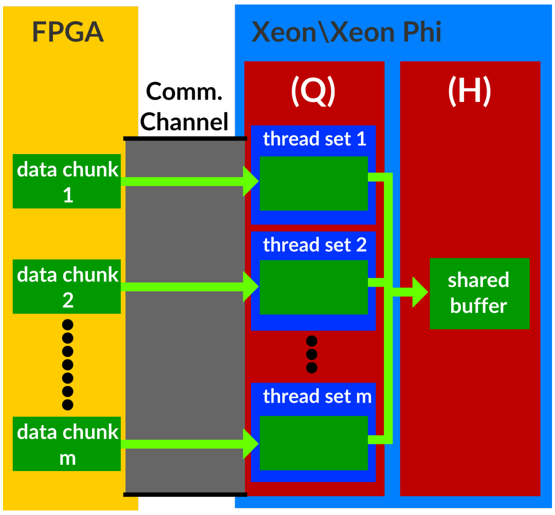

Fig. 12. Overview of system communication and data flow. Variable *m* should be chosen to fulfill real-time requirements described in figure 10. Scope of this work consists only of Q part.

VIII. CONCLUSIONS

The paper proposes the architecture, logical decomposition of the problem and efficiency analysis of the formerly implemented algorithm of pulse parameters estimation on Intel HPC devices. The comparison of work execution time depending on the number of threads and the size of input data chunk was performed. The estimated usage of CPU in proposed solution is up to 50% for a single Intel Xeon CPU and up to 20% for a single Intel Xeon Phi. With a possibility to use up to 4 devices on a single PC a lot of computational resources may be left to implement other parts of pulse parameters estimation algorithm and histogram calculation. Such solution may be used as a replacement of FPGA in a processing part of measurement systems, especially those based on GEM detectors.

The method may be used to achieve the desired balance between the RTT time, throughput and computational time for a different number of channels for algorithms of different complexity.

REFERENCES

[1] B. Plumer, “Have we hit ”the end of the fossil fuel era”? Not even close.” http://www.vox.com/2015/12/14/10121638/fossil-fuel-dominance, 2015.

[2] “The End Of Fossil Fuels,” https://www.ecotricity.co.uk/our-green-energy/energy-independence/the-end-of-fossil-fuels.

[3] P. Linczuk, R. D. Krawczyk, K. T. Pozniak, G. Kasprowicz, A. Wojenski, M. Chernyshova, and T. Czarski, “Algorithm for fast event parameters estimation on GEM acquired data,” *SPIE*, vol. 10013, 2016.

[4] A. Wojenski, K. T. Pozniak, G. Kasprowicz, P. Kolasinski, R. Krawczyk, W. Zabolotny, M. Chernyshova, T. Czarski, and K. Malinowski, "Fpga-based gem detector signal acquisition for sxr spectroscopy system," *Journal of Instrumentation*, vol. 11, no. 11, p. C11035, 2016. [Online]. Available: http://stacks.iop.org/1748-0221/11/i=11/a=C11035

[5] T. Czarski, M. Chernyshova, K. Malinowski, K. T. Pozniak, G. Kasprowicz, P. Kolasinski, R. Krawczyk, A. Wojenski, and W. Zabolotny, "The cluster charge identification in the gem detector for fusion plasma imaging by soft x-ray diagnostics," *Review of Scientific Instruments*, vol. 87, no. 11, p. 11E336, 2016. [Online]. Available: http://aip.scitation.org/doi/abs/10.1063/1.4961559

[6] T. Czarski, K. T. Pozniak, M. Chernyshova, K. Malinowski, G. Kasprowicz, P. Kolasinski, R. Krawczyk, A. Wojenski, and W. Zabolotny, "On line separation of overlapped signals from multi-time photons for the GEM based detection system," *SPIE*, vol. 9662, 2015.

[7] "Xeon Phi 7250 KNL generation processor," http://ark.intel.com/products/94035/Intel-Xeon-Phi-Processor-7250-16GB-1_40-GHz-68-core.

[8] J. Lawley, "Understanding Performance of PCI Express Systems," http://www.xilinx.com/support/documentation/white_papers/wp350.pdf.

[9] L. Rota, M. Caselle, S. Chilingaryan, A. Kopmann, and M. Weber, "A pcie dma architecture for multi-gigabyte per second data transmission," *IEEE Transactions on Nuclear Science*, vol. 62, no. 3, pp. 972–976, June 2015.

[10] "Intel Xeon Processor E5-2630 v3," http://ark.intel.com/products/83356/Intel-Xeon-Processor-E5-2630-v3-20M-Cache-2_40-GHz.

[11] "Intel Xeon Phi Coprocessor 31S1P," http://ark.intel.com/products/79539/Intel-Xeon-Phi-Coprocessor-31S1P-8GB-1_100-GHz-57-core.

[12] "Intel Server Board S2600CW Family," http://www.intel.com/content/www/us/en/motherboards/server-motherboards/server-board-s2600cw.html.

[13] "OpenMP* Thread Affinity Control," https://software.intel.com/en-us/articles/openmp-thread-affinity-control.

[14] "Intel Dual-Socket Server Boards," http://ark.intel.com/products/family/43716/Dual-Socket-Server-Boards.