Proc. of the Sixth International Conference on Advances in Computing, Electronics and Communication - ACEC 2017.

ISBN: 978-1-63248-138-2 doi: 10.15224/ 978-1-63248-138-2-10

Hybrid Parallel Implementation of Multiple Sequence Alignment Software ClustalW on Intel Xeon Phi

Plamenka Borovska, Veska Gancheva, Ivailo Georgiev

***Abstract*— This work is aimed to investigate and to improve the performance of multiple sequence alignment software ClustalW on the test platform EURORA at CINECA, for the case study of the influenza virus sequences. The objective is code optimization, porting, scaling and performance evaluation of parallel multiple sequence alignment software ClustalW for Intel Xeon Phi (the MIC architecture). For this purpose a parallel multithreaded optimization including OpenMP has been implemented and verified. The experimental results show that the hybrid parallel implementation utilizing MPI and OpenMP provides considerably better performance than the original code.**

***Keywords*— bioinformatics, ClustalW, computer cluster EURORA, hybrid programming, high performance computing, multiple sequence alignment, parallel programming, performance.**

I. Introduction

The fundamental scientific studies are in revolution era by the big files and flows of data. One of the fields of the fundamental science, strongly dependent from the development of big data, is the field of molecular and computational biology [1]. In the biological sciences there are very well established practices of collecting data in the public and generally accessible data bases, which are used by the scientists from all over the world, working on concrete subjects. The development of the bioinformatics stimulates in high extent the methods for processing and analyzes of collected data. The technological progress, as well the next generation sequencing, yielded to exponential grow of size and number of experimental data, and as a result the well-known methods and technologies became not applicable to the new challenges of the big flows of data. Many scientific research teams are doing prognostics for the significance of big data, and most analyses for the period till 2025 list astronomy, molecular and computational biology, medicine and meteorology as directions of fundamental science, strongly dependent and influenced from the development of the big files and flows of data [2].

The parallel implementation of methods and algorithms for analysis of biological data using high-performance computing is essential to accelerate the research and reduce the financial cost. Multiple sequence alignment (MSA) is an important method for biological sequences analysis and involves more than two biological sequences, generally of the protein, DNA, or RNA type [3]. This method is computationally difficult and is classified as a NP-hard problem [4]. ClustalW software has become the most popular algorithm and implements a progressive method for multiple sequence alignment [5]. ClustalW computes the best match for the selected sequences, and lines them up so that the identities, similarities and differences can be seen. The basic algorithm behind ClustalW proceeds in three stages: pairwise alignment (PA), guide tree (GT) and multiple alignment (MA). Pairwise alignment computes the optimal alignment cost for each pairs of sequences. A distance matrix is built up; its entries show the degree of divergence for each pair of sequences in evolution. Distance is calculated as the percentage of nonidentity residues between two sequences. An evolutionary guide tree is constructed of the distance matrix using the sequence similarity matrix and Neighbor-Joining algorithm [6]. The tree holds values for each sequence that represent its similarity to all other sequences. The algorithm aligns the sequences progressively according to the branching order in the guide tree by first aligning the most similar pair of sequences, then the next most similar pair and so on. ClustalW phases are relatively independent. Each of the phases produces intermediate data which is used as an input for the next one. The execution time is strongly dependent on the number of sequences as well as their size. ClustalW-MPI [7] is a distributed and parallel implementation for distributed computer clusters and for traditional parallel computers.

This research is aimed to investigate and improve the performance of multiple sequence alignment software ClustalW on the computer system EURORA (Intel Xeon Phi) [8] at CINECA, for the case study of the influenza virus sequences. The objective is porting, optimization, scaling, performance evaluation and profiling of parallel multiple sequence alignment software ClustalW. For this purpose a hybrid parallel (MPI+OpenMP) version has been implemented and verified as part of the work done in order to assess the efficiency of the algorithm.

The paper is structured as follows. Section II presents the experimental framework. Section III explains the experimental results based on parallel ClustalW MPI Implementation. The

Plamenka Borovska
Technical University of Sofia, Department of Informatics
Bulgaria

Veska Gancheva
Technical University of Sofia, Department of Programming and Computer Technologies
Bulgaria

Ivailo Georgiev
Bulgarian Academy of Science, Institute of Microbiology
Bulgaria

Proc. of the Sixth International Conference on Advances in Computing, Electronics and Communication - ACEC 2017.

ISBN: 978-1-63248-138-2 doi: 10.15224/ 978-1-63248-138-2-10

design and development of a hybrid MPI/OpenMP implementation is explained in Section IV. The experiments, performance evaluation and results analysis are discussed in Section V. Section VI presents ClustalW 2.1 porting and optimization on Intel Xeon Phi. We present the conclusion in Section VII.

II. Experimental Framework

The experimental framework of the investigations is based on the computer system Eurora (Intel Xeon Phi) at CINECA. EURORA is a cluster made of 65 nodes of different types [6]:

Compute Nodes: There are 64 16-core compute cards (nodes). Half of the nodes contain 2 Intel(R) Xeon(R) SandyBridge eight-core E5-2658 processors, with a clock rate of about 2 GHz, while the other half of the cards contain 2 Intel(R) Xeon(R) SandyBridge eight-core E5-2687W processors, with a clock of about 3 GHz. 58 compute nodes have 16GB of memory, but the safely allocatable memory on the node is 14 GB (see PBS resources: memory allocation). The remaining 6 nodes (with processors at 3 GHz clock rate) have 32 GB RAM. The EURORA cores are capable of 8 floating point operations per cycle. 32 compute cards have two nVIDIAK20 (Kepler) GPU cards and 32 compute cards have two Intel Xeon Phi accelerators.

Login nodes: The Login node has 2 Intel(R) Xeon(R) esa-core Westmere E5645 processors at 2.4 GHz Intel Xeon (Esa-Core Westmere) E5645 2.4 GHz.

All the nodes are interconnected through a custom Infiniband network, allowing for a low latency/high bandwidth interconnection.

Intel Xeon Phi is the first Intel Many Integrated Core (Intel MIC) architecture product. Each card consists of 60 physical cores (@1.1 Ghz) and each core is able to handle up to 4 thread using hyperthreading. Each core has one Vector Processing Unit able to deliver for each clock cycle: 8 Fused Multiply and Add (FMA) floating point operations in double precision and 16 Fused Multiply and Add (FMA) floating point operations in single precision. The Phi has a peak performance of 1056 GFlops in double precision and 2112 Gflops in single precision. Each Phi coprocessor has a RAM memory of 8 GB, and a peak bandwidth of 352 GB/s.

III. Parallel ClustalW MPI Implementation

Some experiments have been carried out utilizing parallel program implementation of ClustalW MPI. Influenza virus A/H1N1 sequences obtained from Genbank [9] have been used as experimental data. The objective of the experiments is to measure the parallel performance parameters. Similarity searching between RNA segments of various influenza viruses A has been carried out based on the parallel program implementation of ClustalW using MPI on EURORA system (Intel Xeon Phi) in native mode.

Detailed information about the duration of the three computational phases of the ClustalW-mpi for the case study of the influenza virus A/H1N1 are presented in Table I. The input file consists of 98 influenza virus protein sequences, each with length of 481 symbols.

TABLE I. PHASES DURATION TIME OF CLUSTALW-MPI ON INTEL XEON PHI ARCHITECTURE USING VARIOUS NUMBERS OF THREADS

Numbers of threads	Execution time, sec			
	PA	***GT***	***MA***	***Total***
1	1809,6	0,33	3204,55	5014,48
20	96,3	0,25	48,2	144,75
40	49,3	0,24	47,6	97,14
60	31,64	0,25	46,76	78,65
120	28,6	0,24	42,8	71,64
240	28,2	0,24	43,6	72,04

The speed-up is evaluated as a ratio of the execution time on 1 core to the execution time on 20, 40, 60, 120, and 240 threads respectively. The experimental results for the speed-up of on Intel Xeon Phi architecture using various numbers of threads with respect to 1 thread are shown on Fig. 1.

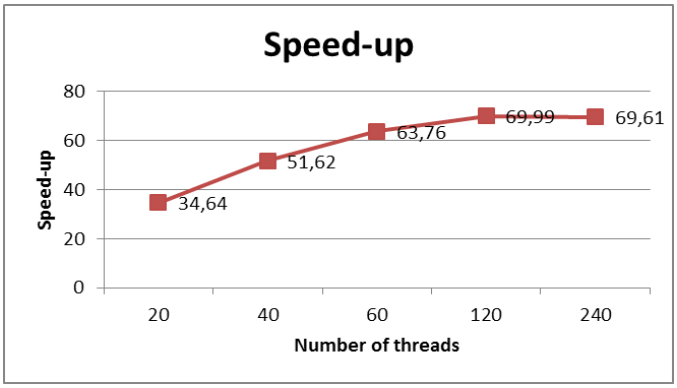

Figure 1. Speed-up during the execution of ClustalW-mpi on Intel Xeon Phi architecture using various numbers of threads

The experimental results and analyses of ClustalW-mpi software show that the times for parallel execution during the first stage (pairwise alignment) and the third stage (multiple alignment) decrease with increasing the number of threads, while the execution time of the second stage (guide tree construction) remains approximately constant.

IV. Hybrid Parallel ClustalW Implementation

The computational aspect of this project is to investigate the parallel performance in respect to the efficiency, scaling and profiling of parallel multiple alignment on EURORA system utilizing parallel OpenMP deployment and optimization of the MPI-based parallel implementation of ClustalW algorithm for Intel Xeon Phi architecture. The parallel hybrid MPI/OpenMP computational model of ClustalW algorithm for multiple sequence alignment is presented in Fig. 2.

Proc. of the Sixth International Conference on Advances in Computing, Electronics and Communication - ACEC 2017.

ISBN: 978-1-63248-138-2 doi: 10.15224/ 978-1-63248-138-2-10

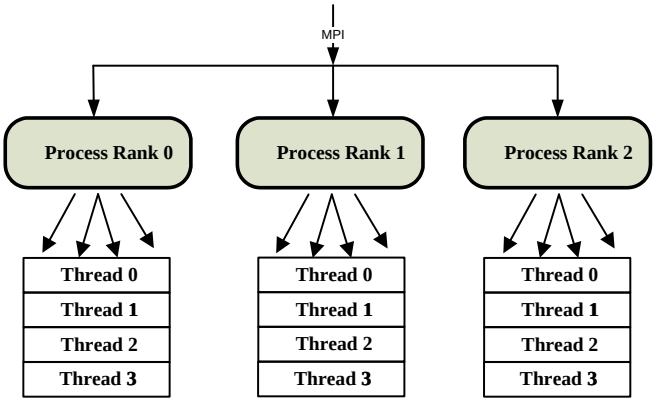

Figure 2. Hybrid OpenMP/MPI parallel computational model of ClustalW

The ClustalW code implementation on Intel Xeon Phi requires hybrid granularity of parallelization. Coarse granule computing for each node (multithreaded process) that runs multithreading (fine granule) of the cores within the computing node on Intel Xeon Phi [10] [11].

In order to choose an appropriate part of the code to be parallelized, we used **gprof** for profiling the program: how often each function is called and how long it takes to perform.

For this purpose, first the flag –**pg** in the Makefile is added at the end of the compilation and linking rows. Then the following command is used:

gprof clustalw-mpi gmon.out > gprof.output

The file gprof.output shows the profiling results in Fig. 3.

Each sample counts as 0.01 seconds.

% time	cumulative seconds	self seconds	calls	self Ks/call	total Ks/call	name
25.22	1280.69	1280.69	1910035	0.00	0.00	diff
21.72	2384.01	1103.31	3908	0.00	0.00	calc_mean
16.36	3214.72	830.72	1910035	0.00	0.00	forward_pass
15.24	3988.78	774.05	1910035	0.00	0.00	reverse_pass
13.95	4697.12	708.35	4204763923	0.00	0.00	calc_score
2.84	4841.31	144.18	1	0.14	0.14	calc_similarities
2.71	4979.07	137.77	1	0.14	0.14	nj_tree
1.04	5032.09	53.02	1910035	0.00	0.00	tracepath
0.21	5042.86	10.77	57027490	0.00	0.00	prfscore
0.12	5049.20	6.34	1954	0.00	0.00	calc_prf1
0.10	5054.17	4.97	1	0.00	1.11	read_tree
0.08	5058.19	4.01	1910035	0.00	0.00	count_gaps
0.07	5061.56	3.37	3908	0.00	0.00	calc_h_penalties
0.07	5064.92	3.37	1	0.00	0.01	aln_score
0.07	5068.27	3.35	3908	0.00	0.00	calc_p_penalties
0.05	5071.01	2.74	1954	0.00	0.00	prfalign
0.04	5073.03	2.02	1	0.00	3.65	pairalign
0.04	5074.84	1.80	1954	0.00	0.00	pdiff

Figure 3. Profiling results of ClustalW-mpi

The analysis of the ClustalW code shows that one of the time-consuming functions is **calc_similarities().**This function is executed only once, but the execution takes a relatively large time - 144.18 sec.

The function **calc_similarities()** in the file **calctree.c** is parallelized utilizing OpenMP. In each **For Loop** a parallel for OpenMP directive is placed. This results in having each OpenMP thread assigned to a different sequence every time until the work is completed.

Initial benchmark tests have shown that increasing the number of threads for the same number of MPI processes and nodes decreases the execution time significantly (Fig. 4). The function execution takes only 0.05 sec.

0.08	4899.39	3.69	3908	0.00	0.00	calc_h_penalties
0.07	4902.70	3.31	1	0.00	0.01	aln_score
0.06	4905.77	3.07	3908	0.00	0.00	calc_p_penalties
0.06	4908.71	2.94	1954	0.00	0.00	prfalign
0.04	4910.68	1.97	1	0.00	3.63	pairalign
0.04	4912.60	1.92	1954	0.00	0.00	pdiff
0.02	4913.44	0.84	89909402	0.00	0.00	open_penalty2
0.02	4914.19	0.74	60308256	0.00	0.00	open_penalty1
0.01	4914.80	0.61	58666896	0.00	0.00	ext_penalty1
0.01	4915.37	0.57	57869664	0.00	0.00	ext_penalty2
0.01	4915.81	0.44	1954	0.00	0.00	add_ggaps
0.01	4916.08	0.27	3908	0.00	0.00	calc_gap_coeff
0.00	4916.26	0.18	1	0.00	0.00	clustal_out
0.00	4916.36	0.10	1954	0.00	0.00	calc_prf2
0.00	4916.46	0.10				del
0.00	4916.54	0.08	1913953	0.00	0.00	info
0.00	4916.60	0.06	1866	0.00	0.00	mark_group2
0.00	4916.65	0.05	3	0.00	0.00	two_way_split
0.00	4916.70	0.05	1	0.00	0.00	calc_similarities
0.00	4916.75	0.05	1	0.00	1.15	malign
0.00	4916.79	0.04	472868	0.00	0.00	local penalty

Figure 4. Profiling results of ClustalW-mpi-openmp

v. Hybrid Parallel ClustalW Performance Evaluation

The objective of the experiments is to measure the parallel performance parameters of the hybrid ClustalW MPI+OpenMP parallel program implementation on EURORA system (Intel Xeon Phi) in native mode. Some experiments have been carried out utilizing hybrid parallel program implementation of ClustalW. Influenza virus A/H1N1sequences have been used as experimental data.

Detailed information about the duration of the three computational phases of the ClustalW-mpi-openmp for the case study of the influenza virus A/H1N1 is presented in Table II. The input file consists of 98 influenza virus protein sequences, each with length of 481 symbols.

The speed-up is computed as a ratio of the execution time on 1 core to the execution time on 20, 40, 60, 120, and 240 threads respectively. The experimental results for the speed-up using various numbers of threads with respect to 1 thread for the case of ClustalW-mpi-openmp are shown on Fig. 5.

TABLE II. PHASES DURATION TIME OF CLUSTALW-MPI-OPENMP ON INTEL XEON PHI ARCHITECTURE USING VARIOUS NUMBERS OF THREADS

Numbers of threads	Execution time, sec			
	PA	*GT*	*MA*	*Total*
1	1809,6	0,33	3204,55	5014,48
20	58,41	0,24	31	89,65
40	28,45	0,22	30,28	58,95
60	19,01	0,22	30,27	49,5
120	16,64	0,22	30,8	47,66
240	15,6	0,23	31,6	47,43

The experimental results and analyses of ClustalW software show that the time for parallel execution during the first stage (pairwise alignment) decreases with increasing the number of threads. The experimental results presented in TABLE III. show that the MPI+OpenMP program implementation achieves higher reduction in execution time as the first stage and the third stage. The total execution time is decreased by approximately 1.5 times.

Proc. of the Sixth International Conference on Advances in Computing, Electronics and Communication - ACEC 2017.

ISBN: 978-1-63248-138-2 doi: 10.15224/ 978-1-63248-138-2-10

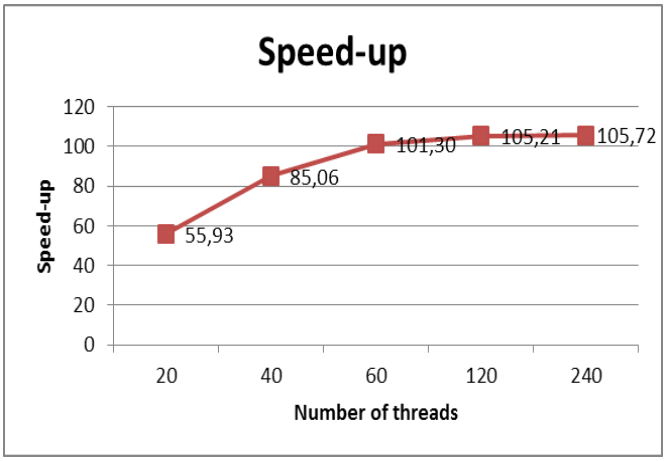

Figure 5. Speed-up during the execution of ClustalW-mpi-openmp on Intel Xeon Phi architecture using various numbers of threads

TABLE III. CLUSTALW-MPI-OPENMP EXECUTION TIME VS. CLUSTALW-MPI ON INTEL XEON PHI ARCHITECTURE USING VARIOUS NUMBERS OF THREADS

Numbers of threads	PA	MA	Total
20	1,65	1,55	1,61
40	1,73	1,57	1,65
60	1,66	1,54	1,59
120	1,72	1,39	1,50
240	1,81	1,38	1,52

VI. ClustalW 2.1 Code Porting and Optimization

The latest version of ClustalW code (version 2.1) has been partly optimized for Intel MIC accelerators. In the current implementation the first part (pairwise alignment) of the program was parallelized for improving the performance on Intel MIC accelerators.

The experimental results are presented in the **Error! Reference source not found.**Achieved speed-up for the whole code is shown on the **Error! Reference source not found.** and achieved speed-up for the parallel section of ClustaW 2.1 code is shown on the Figure 7.

TABLE IV. EXPERIMENTAL RESULTS FOR THE PARALLEL PART THE PARALLEL SECTION OF THE CODE

Numbers of threads	Execution time for Clustalw-2.1, sec			
	PA	*GT*	*MA*	*Total*
1	2837,4	1304,2	973,5	5115,1
20	740,5	1304,2	973,5	3018,2
40	527,8	1304,2	973,5	2805,5
60	471,3	1304,2	973,5	2749,0
120	442,1	1304,2	973,5	2719,8
240	440,8	1304,2	973,5	2718,5

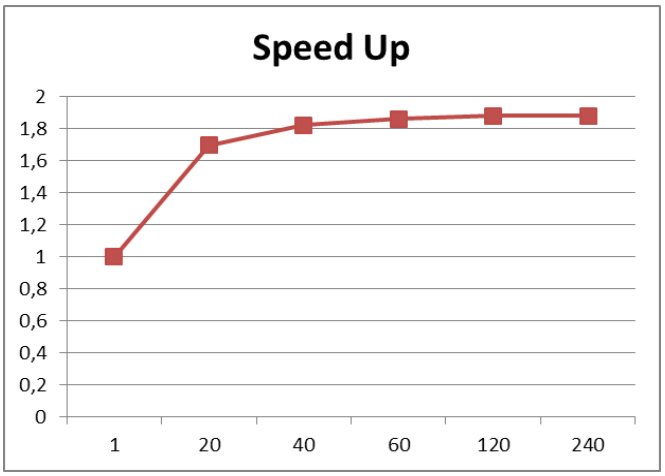

Figure 6. Speed-up during the execution of optimized version of ClustalW 2.1 on Intel Xeon Phi using various numbers of threads.

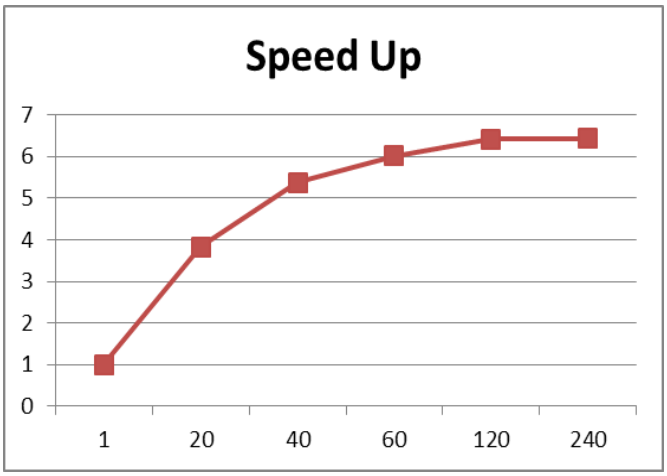

Figure 7. Achieved speed-up for the parallel part (pairwise alignment) of clustaw 2.1 code.

The experimental results for the parallel section of the code shows that the speed-up increases as well as reaching the core numbers limitation.

VII. Conclusion

The parallel software for multiple sequence alignment ClustalW has been optimized through multithreaded implementation. The ClustalW code is ported on the Intel MIC architecture (EURORA system). It is best suited to work in native Xeon Phi MPI mode in a combination of MPI tasks and threads. Parallel performance evaluation has been investigated experimentally. Parallel performance parameters such execution time and speed-up have been measured. The performance estimation and analyses show that the hybrid parallel program implementation utilizing MPI and OpenMP of ClustalW software for multiple sequence alignment scales well as the number of the cores increases up to 60 cores. On the other hand the performance of the MPI+OpenMP implementation is better than MPI only implementation, since

Proc. of the Sixth International Conference on Advances in Computing, Electronics and Communication - ACEC 2017.

ISBN: 978-1-63248-138-2 doi: 10.15224/ 978-1-63248-138-2-10

the total execution time is decreased by approximately 1.5 times. The optimized code is universal and can be apply to other similar research projects and experiments in the field of bioinformatics and will allow researchers to conduct their experiments on even more powerful supercomputers. They will be able to perform simulations with very large amounts of data.

Acknowledgment

This work was supported by the National Science Fund, Bulgarian Ministry of Education and Science, Grant DN 07/24. Thanks the project PRACE for the access to the EURORA system.

References

[1] European Big Data Value Partnership, Strategic Research and Innovation Agenda, January 2016.

[2] Stephens ZD, Lee SY, Faghri F, Campbell RH, Zhai C, Efron MJ, Iyer R, Schatz MC, Sinha S, Robinson GE.Big Data: Astronomical or Genomical?, PLoS Biol. 2015 Jul 7;13(7):e1002195. doi: 10.1371/journal.pbio.1002195, eCollection 2015.

[3] H., Carrillo and D. Lipman, "The Multiple Sequence Alignment Problem in Biology," SIAM Journal of Applied Mathematics, vol. 48, no. 5, 1988, pp. 1073-1082.

[4] S. Sze, Y. Lu, and Q. Yang, "A polynomial time solvable formulation of multiple sequence alignment," Journal of Computational Biology, vol. 13, no. 2, 2006, pp. 309–319, doi:10.1089/cmb.2006.13.309.

[5] J. Thompson, D. Higgins, T. Gibson, "ClustalW: Improving the Sensitivity of Progressive Multiple Sequence Alignment through Sequence Weighting, Position-Specific Gap Penalties and Weight Matrix Choice," Nucleic Acids Research, Vol. 22, No. 22, 1994, pp. 4673-4680.

[6] N. Saitou, M. Nei, "The Neighbor-Joining Method: A New Method for Reconstructing Phylogenetic Trees," Molecular Biology and Evolution, 4(4), 1987, pp. 406-425.

[7] Kuo-Bin Li, "ClustalW-MPI: ClustalW Analysis Using Distributed and Parallel Computing", Bioinformatics, 2003, 19(12), 1585--1586.

[8] EURORA – Configuration, http://www.hpc.cineca.it/content/eurora-user-guide#systemarchitecture, retrieved: 17.12.2013.

[9] GenBank, http://www.ncbi.nlm.nih.gov/Genbank/

[10] James Jeffers, James Reinders, Intel Xeon Phi Coprocessor High Performance Programming, ISBN-13: 978-0-12-410414-3, Morgan Kaufmann Publisher, Feb 2013.

[11] http://software.intel.com/en-us/articles/intel-xeon-phi-coprocessor-codename-knights-corner

About Authors:

Plamenka Borovska, Prof. PhD is a prominent scientist and University professor in the area of supercomputer architecture and applications, optimization algorithms, machine learning, in silico experimentation and bioinformatics. Prof. Borovska is author of 150 scientific publications and papers at international scientific conferences and journals. She holds a patent for "multiprocessor systems" recognized in the United States, England and Germany. She has participated in over 35 research projects at national and European level in the field of parallel processing, supercomputing architectures and applications: TEMPUS, ERASMUS, PRACE, ETN FETCH and the Programme for Development of Human resources. Under her scientific supervision 16 PhD thesis have been successfully defended.

Veska Gancheva, Asoc. Prof. PhD is expert with extensive experience in information technology, bioinformatics, in silico biological experiments parallel methods, algorithms and models, software technologies, supercomputing applications, cloud computing, management and analysis of data. She is author of 45 scientific publications and has participated in over 30 research projects at national and European level: ERASMUS, PRACE and the Operational Programme for Human Resources Development.

Ivailo Georgiev, PhD is expert in high-performance computing platforms, bioinformatics, integration and configuration of software, parallel methods, algorithms and models, supercomputing applications, in silico biological experiments, numerical methods, and artificial intelligence. He is author of 18 scientific publications and has participated in 10 research projects at national and European level - PRACE Second Implementation Phase Project, EU's Seventh Framework.