

## Setting Up and Implementation of the Parallel Computing Cluster in Higher Education

<https://doi.org/10.3991/ijet.v14i06.9736>

Meruert Serik<sup>(✉)</sup>, Nursaule Karelkhan  
L.N. Gumilyov Eurasian National University, Astana, Kazakhstan  
serik\_meruert@mail.ru

Jaroslav Kultan  
University of Economics, Bratislava, Slovakia

Zhandos Zulpykhar  
L.N. Gumilyov Eurasian National University, Kazakhstan

**Abstract**—In this article, we describe in detail the setting up and implementation of the parallel computing cluster for education in the Matlab environment and how we solved the problems arising on this way. We also describe the comparative analysis of parallel computing cluster by the example of matrix multiplication by a vector with large dimensions. First calculations were performed on one computer, and then on a parallel computing cluster. In the experiment, we proved the effectiveness of parallel computing and the necessity of the setting up of the parallel computing cluster. We hope that the creation of a parallel computing cluster for education will help in teaching the subject of parallel computing at higher schools that do not have sufficient hardware resources. This paper presents unique setting up and implementation of the parallel computing cluster for teaching and learning of the parallel computing course and a wide variety of information sources from which instructors can choose.

**Keywords**—Education, parallel computing, parallel computing cluster, Matlab

### 1 Introduction

Today, there are also a large number of important practical problems associated with computational experiments, the solution of which requires the use of huge computer power.

The most important part of setting up and implementing high-performance parallel computing cluster is the training of qualified specialists in the field of education and information technologies. Therefore, in order to develop education in this area, it is necessary to provide relevant equipment to higher education. In this regard, the education cluster EUBA was setup at the department of Applied Informatics at the University of Economics in Bratislava for implementation in the educational process.

The cluster we call the education cluster EUBA similar to the Beowulf cluster, has some differences.

The name Beowulf originally referred to a specific computer built in 1994 by Thomas Sterling and Donald Becker at NASA [1].

Beowulf is a computer cluster of what is normally identical, commodity-grade computers networked into a small local area network with libraries and programs installed which allow processing to be shared among them. The result is a high-performance parallel computing cluster from inexpensive personal computer hardware. No particular piece of software defines a cluster as a Beowulf. Beowulf clusters normally run a Unix-like operating system, such as BSD, Linux, or Solaris, normally built from free and open source software. Commonly used parallel processing libraries include Message Passing Interface (MPI) and Parallel Virtual Machine (PVM). Both of these permit the programmer to divide a task among a group of networked computers, and collect the results of processing. Examples of MPI software include Open MPI or MPICH. There are additional MPI implementations available (Beowulf cluster. In Wikipedia). Difference, the education cluster EUBA was set in the Windows operating system and in Matlab environment.

**The purpose of the research** is determining the practical and methodological foundations for improving the training of students in the aspect of studying courses on parallel computing, as well as their practical implementation in the learning process.

**The subject of our research** is the introduction into the content of the educational process of the setting up and implementation parallel computing cluster.

In the course of our research, sources on the topic were analyzed. The experience of other scientists from different countries was studied, namely the setting up and implementation of a parallel computing cluster for education. In works, “Barrier to parallel processing courses in computer education and solutions”[3], “Teaching parallel programming using Java”[4], “Improvement of students' training in parallel and cloud computing”[5], authors use of parallel computing in the learning process is considered.

In the work, “Cluster computing in the classroom and integration with computing curricula 2001”, the authors share their experiences in teaching cluster computing and the topics chosen, depending on course objectives, perspective themes of training, offered course components for teaching in a parallel computing cluster [6].

In the article, “Teaching High-Performance Computing on a High-Performance Cluster”, authors present their experiences of how a state of the art midsize Linux cluster, bought and operated on a department level primarily for education and algorithm development purposes, can be used for teaching a large variety of HPC aspects such as basics of parallel algorithms, classical tuning, or hardware aware programming. Special focus is put on the effects of such an approach on the intensity and sustainability of learning [7].

Virtual machines (VMs) installed on available computer lab resources can be used to simulate high-performance cluster computing environments. In article, “Virtual clusters for parallel and distributed education”, authors describes two such virtual clusters in use at small colleges, demonstrates their effectiveness for parallel computing education, and provides information about how to obtain the VMs for use

in an educational lab setting. They have used these clusters to introduce parallelism into several courses in their undergraduate curriculum [8].

The authors of work “The realization of small cluster parallel computing environment for college education” wrote about the message passing interface (MPI) is taken to build a small cluster of Linux-based systems with the number of ordinary PC and establish a parallel development environment with lower investment. Meanwhile it is verified and proves to be reliable. System takes the advantages of low-cost hardware to provide a practical parallel programming environment on clusters for general research institutes and research schools [9].

The author of paper, Research on cloud computing and its application in big data processing of distance higher education, study the parallel k-means clustering algorithm based on cloud computing platform Hadoop, and give the design and strategy of the algorithm [10].

In the article, the application of coarse-grained parallel genetic algorithm with Hadoop in university intelligent course-timetabling system, which is based on the cloud-computing platform with Hadoop, presents an improved method fusing Coarse-grained parallel genetic algorithm (CGPGA) and Map/Reduce programming model, to solve the university course-timetabling problem. The simulation experiment results show that, compared with the traditional genetic algorithm, CGPGA not only improves the success rate but also reduces the conflict rate in solving the university course-timetabling problem [11].

In work “Developing a hands-on course around building and testing high performance computing clusters” the authors describe a successful approach to designing and implementing a high performance computing (HPC) class focused on creating competency in building, configuring, programming, troubleshooting, and benchmarking HPC clusters [12].

In work, “Application of teaching quality assessment based on parallel genetic support vector algorithm in the cloud computing teaching system” authors compare the parallel algorithm and the serial algorithm on the Hadoop platform. The results of experiments show that the GA-SVM based on map reduce is suitable for teaching quality assessment under the environment of big data. As a new technology, cloud computing provides a broad space for the development of a new model in the aspects of hardware environment construction, software resource development, network teaching implementation and personal knowledge management. In order to effectively deal with the challenges of big data processing in the field of education, this paper proposes a GA-SVM teaching quality assessment algorithm which is based on MapReduce [13].

In the scientific and pedagogical literature of the university of economics in Bratislava, studies of the problems of parallel computing are presented in the papers “Parallelization of instance methods of a remote object of a distributed .net application mechatronic”[14], “Parallel programming”[15].

Analysis of scientific literature and internet resources show that courses on parallel computing were introduced in higher education in foreign universities, at the same time. This paper presents unique setting up and implementation of the parallel

computing cluster for teaching and learning of the parallel computing course and a wide variety of information sources from which instructors can choose.

## 2 Method

The parallel computing cluster was configured in the Windows 10 operating system and in the Matlab interactive environment for teaching and learning of the parallel computing course.

Large-scale simulations and data processing tasks that support engineering and scientific activities such as mathematical modeling, algorithm development, and testing can take an unreasonably long time to complete or require a lot of computers memory. You can speed up these tasks by taking advantage of high-performance computing resources, such as multicore computers, GPUs, computer clusters, and grid and cloud computing services. Math Works parallel computing products let you use these resources from Matlab and Simulink without making major changes to your computing environment and workflows [16].

For organization of the parallel computing cluster for education we use single instruction, multiple data. The educational cluster consists of 3 computers (Figure 1):

- Master (for managing the cluster and used to interacting with users)
- Nodes (a group of computers)
- Network Switch

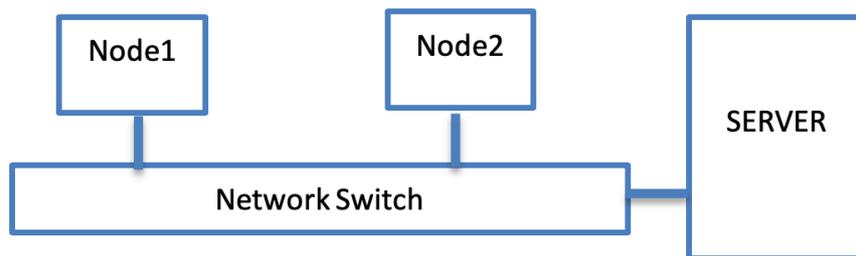


Fig. 1. Educational cluster

When configuring the cluster, we must select the same type of computers by the characteristic. The same characteristics of computers simplifies the setting of parameters. Initially, we tried to configure and apply a parallel computing cluster for performance on weak computers, but, unfortunately, the parallel calculation was allowed slower. Therefore, we chose more powerful computers. Technical Parameters of educational cluster:

- Master: Intel Pentium (R) Dual-Core E2180 CPU 2.0 GHz, RAM 4 Gb
- 2-Nodes: Intel Pentium (R) Dual-Core E2180 CPU 2.0 GHz, RAM 4 Gb
- Network (Mercury8-port 10/100Mbps)

To configure the cluster, the following work was performed.

- Install Windows 10, x64. (If you have another version of Windows installed, that's ok, the main thing is that on all computers there are one typical operating systems. Prior to that, I had experience on the server, I installed Windows Server on the remaining Windows 10. The cluster worked fine.) All computers must have open access without a password.
- Combine computers into one WorkGroup.
- Install Matlab on each computer.
- Disable the firewall on all computers.
- On each computer run Matlab open folder: D:/ProgramFiles/matlab/2015b/toolbox/bin, and run commands (Figure 2):

```
!mdce install  
!mdce start
```

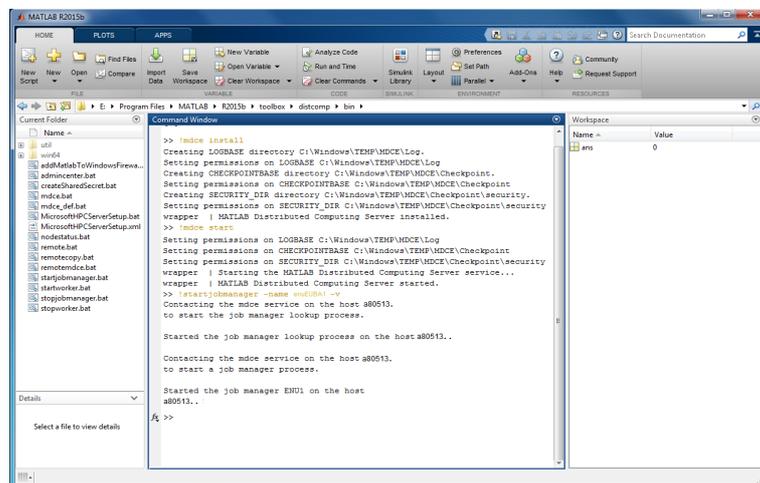


Fig. 2. Mdce service is running

The **mdce** service ensures that all other processes are running and that it is possible to communicate with them. Once the mdce service is running, you can use the **nodestatus** command to obtain information about the mdce service and all the processes it maintains. **mdce install** installs the mdce service in the Microsoft® Windows Service Control Manager. This causes the service to automatically start when the Windows operating system boots up. The service must be installed before it is started. **mdce uninstall** uninstalls the mdce service from the Windows Service Control Manager. Note that if you wish to install mdce service as a different user, you must first uninstall the service and then reinstall as the new user. **mdce start** starts the mdce service. This creates the required logging and checking directories, and then starts the service as specified in the mdce defaults file. **Startjobmanager** starts a job manager process and the associated job manager lookup process under the mdce service, which maintains them after that. The job manager handles the storage of jobs and the distribution of tasks contained in jobs to MATLAB® workers that are

registered with it. The mdce service must already be running on the specified computer [17].

- Next for Server command: `startjobmanager -name EUBA1 -v`
- Next for Node1 command: `startjobmanager -name EUBA2 -v`
- Next for Node2 command: `startjobmanager -name EUBA3 -v`
- Next select commands Parallel-> Discover Cluster on the Server computer (Figure 3). The master computer to find the node computers

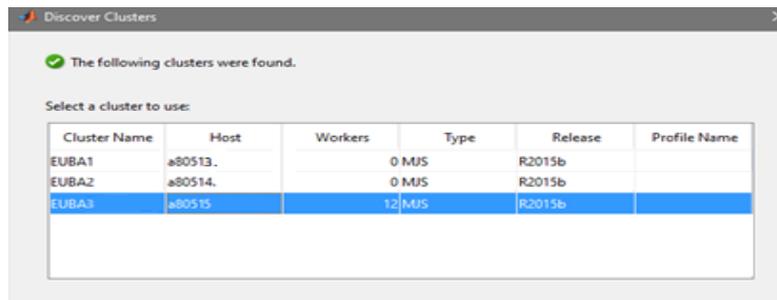


Fig. 3. Discover Cluster

- Run Admincenter, with command: `admin center`
- Choose command Add or Find (Figure 5). We wrote the following IP addresses:
  - 193.87.114.193 (IP address of Node1, a80513)
  - 193.87.114.174 (IP address of Node2, a80514)
  - 193.87.114.180 (IP address of Node3, a80515)

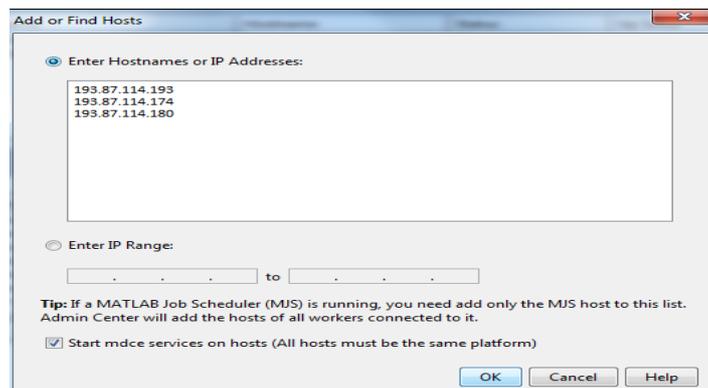


Fig. 4. Window Admin center (command Add or Find)

- The EUBA educational cluster is configured. The educational cluster consists of 6 cores of processors and 12 workers (can be increased to 18 workers).

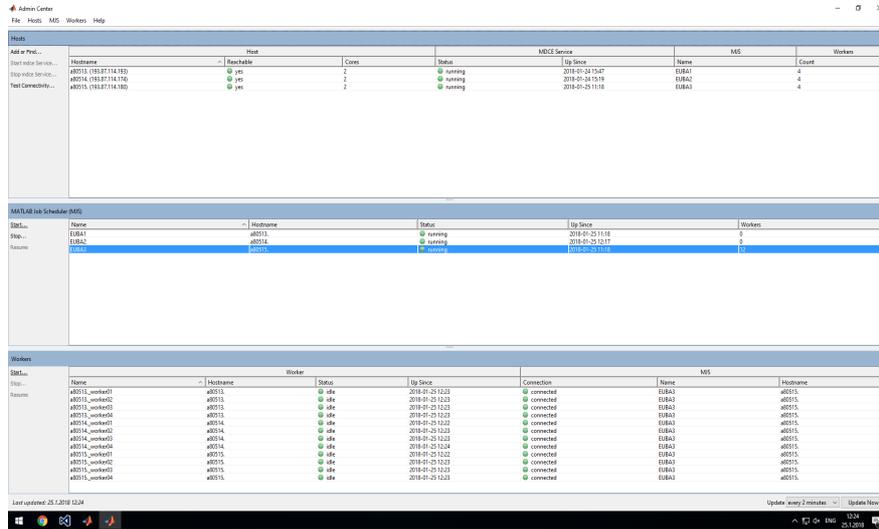


Fig. 5. Characteristics of educational cluster EUBA

Run command: Parallel ->EUBA, Result (Figure 6) Starting parallel pool connected to 12 workers.

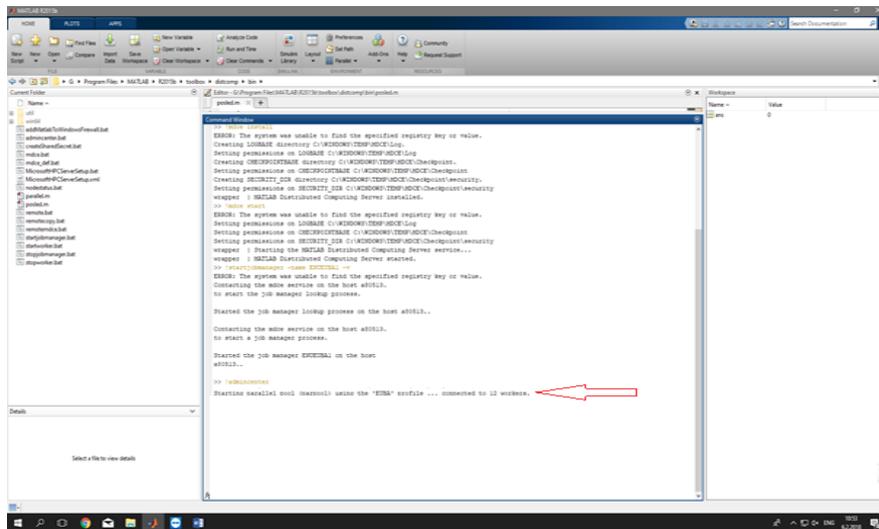


Fig. 6. Educational cluster EUBA

After configuring the cluster in the established university network, there was no parallel computing efficiency. Therefore, we set up a separate modem (Mercury8-port 10/100Mbps) to connect the computers of the parallel computing cluster, the result was effective.

### 3 Results of Research

To analyze the execution of parallel programs, a matrix-vector program of multiplication was performed. The dimensions of the matrix were from 100x100, 1000x1000 to 20000x20000, respectively, from 100, 1000 to 20000[18]. First, the calculations were performed on one computer without using parallel computing, then using parallel computing on a local and a third time using parallel computing on the educational cluster EUBA.

Traditionally	Parallel computing
<pre>clear; n=2000; A = rand(n,n); b = linspace(20,40,n); C=b'; tic for i=1:n x = zeros(1, n); S=0; for j=1:n S=S+A(i,j)*C(j); x(i)=S; end disp(x(i)) end toc</pre>	<pre>clear; n=2000; A = rand(n,n); b = linspace(20,40,n); C=b'; tic parfor i=1:n x = zeros(1, n); S=0; for j=1:n S=S+A(i,j)*C(j); x(i)=S; end disp(x(i)) end toc</pre>

The results of these calculations are as follows (Table 1).

**Table 1.** Computational experimental results.

n	Traditionally	Parallel computing in Local	Acceleration Indicators Local	Parallel computing in Cluster	Acceleration Indicators CLUSTER
10	0,000804	0,21	0,003828571	0,12	0,0067
100	0,002	0,2929	0,006828269	0,14	0,014285714
1000	0,115583	0,4137	0,279388446	0,48	0,240797917
2000	0,478318	0,65025	0,735590927	1,87	0,255785027
3000	0,86236	0,99081	0,870358595	4,23	0,203867612
4000	2,357672	1,811867	1,301238998	7,54	0,312688594
5000	3,333	2,054286	1,622461527	11,75	0,283659574
6000	6,6731	3,2569	2,048911542	16,94	0,39392562
7000	8,9795	4,455687	2,015289674	23,302	0,385353189
8000	12,023849	5,809064	2,069842749	30,0523	0,400097463
9000	16,88805	7,981354	2,115937972	39,0384	0,432600977
10000	22,262362	20,2041	1,101873481	48,306	0,460861218
11000	29,745054	54,0784	0,550035763	55,235	0,538518222
12000	35,458	86,2458	0,411127266	57,258	0,619267177
13000	49,7181032	125,9124	0,394862644	69,235	0,718106495
14000	78,2254	245,124	0,31912583	102,255	0,765003178
15000	296,2145	457,25	0,647817387	350,35	0,845481661

16000	324,259	time out		372,54	0,870400494
17000	399,255	time out		421,125	0,948067676
18000	468,258	time out		433,245	1,080815705
19000	874,153	time out		459,254	1,903419459
20000	1457,214	time out		501,9245	2,903253378
21000	time out			645,123	

We had shared memory (Figure 7); we want to configure the distributed memory (Figure 8). Distributed Memory Like shared memory systems, distributed memory systems vary widely but share a common characteristic. Distributed memory systems require a communication network to connect inter-processor memory [19].

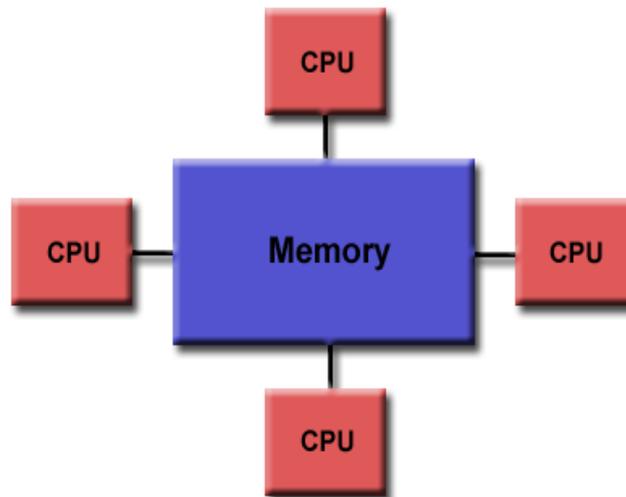


Fig. 7. Shared Memory

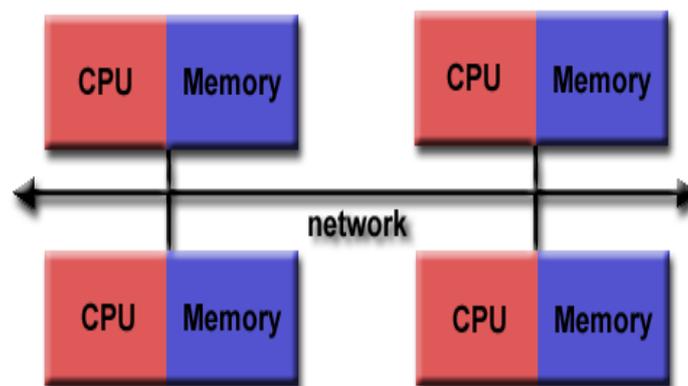


Fig. 8. Distributed Memory

## 4 Discussion

Further, after setting up the cluster from Table 1, we see that the speed of parallel computing on the cluster is slowly increasing, and from 18000 the parallel computing speed is higher. Initial slow increases are associated with the data connection between the processors. But, we see that parallel computing on a cluster of big matrices is much more successful (Figure 9).

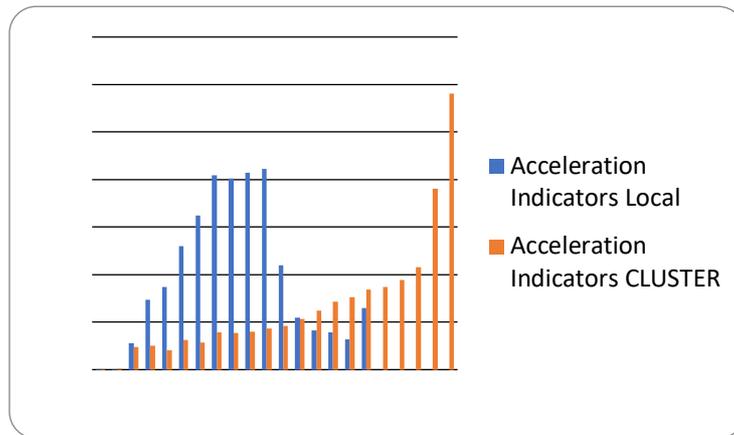


Fig. 9. Acceleration indicators

## 5 Conclusion and Recommendation

In the experiment, we showed that parallel computing will be effective at large computing. Also parallel computing in local requires a large RAM. Computers with the similar characteristics simplify the settings. In addition, when configuring a cluster, we must select powerful computers. In addition, for setting up of the parallel computing cluster, we need a fast network because the network speed is interconnected with the calculation. Therefore, the programmer must know all the subtleties of parallel computing. We hope that the creation of a parallel computing cluster for education will help in teaching the subject of parallel computing.

## 6 Acknowledgement

The parallel computing cluster EUBA had been approved and implemented in the learning process in the department of Applied Informatics at the University of Economics in Bratislava.

## 7 References

- [1] Sterling, T., Savarese, D., Becker, D.J, Dorband, J. E., Ranawake, U.A., & Packer, C.V., (1995). "BEOWULF: A parallel workstation for scientific computation", International Conference on Parallel Processing, Vol. 1: Architecture, pages 11-14, Boca Raton, USA, August 1995.
- [2] Beowulf cluster. In Wikipedia. Retrieved August 17, 2018 from [https://en.wikipedia.org/wiki/Beowulf\\_cluster](https://en.wikipedia.org/wiki/Beowulf_cluster)
- [3] Tseng, Y. (2010). Barrier to parallel processing courses in computer education and solutions. In 4th International Technology, Education and Development Conference (INTED), Valencia: SPAIN. pp. 568-575.
- [4] Shafi, A., Akhtar, A., Javed, A., & Carpenter, B. (2014). Teaching Parallel Programming Using Java. 2014 Workshop on Education for High Performance Computing, 56-63. Doi: <https://doi.org/10.1109/EduHPC.2014.7>
- [5] Serik M., Karelkhan N., Sadvakassova A., & Sarzhanova G.(2017). Improvement of students' training in parallel and cloud computing. *Espacios*, Vol. 38 (Nº 60) pp 3-13. <http://www.revistaespacios.com/a17v38n60/17386003.html>
- [6] Apon, A., Mache, J., Buyya, R., & Jin, H. (2004). Cluster Computing in the Classroom and Integration With Computing Curricula 2001. *IEEE Transactions on Education*, 47(2), 188-195. <https://doi.org/10.1109/TE.2004.824842>
- [7] Bernreuther, M., Brenk, M., Bungartz, H., Mundani, R., & Muntean, I. L. (2005). Teaching High-Performance Computing on a High-Performance Cluster. Lecture Notes in Computer Science Computational Science – ICCS, 1-9. [https://doi.org/10.1007/11428848\\_1](https://doi.org/10.1007/11428848_1)
- [8] Shoop, E., Brown, R., Biggers, E., Kane, M., Lin, D., & Warner, M. (2012). Virtual clusters for parallel and distributed education. *Proceedings of the 43rd ACM technical symposium on Computer Science Education - SIGCSE '12*, 517-522. <https://doi.org/10.1145/2157136.2157287>
- [9] Xu, M., & Su, Q. (2014). The realization of small cluster parallel computing environment for college education. *9th International Conference on Computer Science & Education*. <https://doi.org/10.1109/ICCSE.2014.6926584>
- [10] Zhang G., Li J, & Hao L. (2015). Research on Cloud Computing and Its Application in Big Data Processing of Distance Higher Education, *International Journal of Emerging Technologies in Learning*, 10(8), 55-58, DOI: <https://doi.org/10.3991/ijet.v10i8.5280>
- [11] Wu L. (2015). The application of coarse-grained parallel genetic algorithm with Hadoop in university intelligent course-timetabling system, *International Journal of Emerging Technologies in Learning*, 10(8), 11-15. doi: <https://doi.org/10.3991/ijeSt.v10i8.5206>
- [12] Frinkle, K., & Morris, M. (2015). Developing a Hands-On Course Around Buaround and Testing High Performance Computing Clusters. *Procedia Computer Science*, 51, 1907-1916. <https://doi.org/10.1016/j.procs.2015.05.455>
- [13] Hu, H. & Zheng, J.(2016). Application of Teaching Quality Assessment Based on Parallel Genetic Support Vector Algorithm in the Cloud Computing Teaching System. *International Journal of Emerging Technologies in Learning*. 11(8), 16-21. <https://doi.org/10.3991/ijet.v11i08.6040>
- [14] Kostal I. (2012) Parallelization of instance methods of a remote object of a distributed .net application, *Mechatronika 2012: proceedings of 15th international conference on mechatronics: Prague, December 5-7*, pp. 360-369.

- [15] Hanak J. (2013) Moderné paralelné programovanie, [Modern parallel programming] 2. vydanie. Bratislava: Vydavateľstvo Ekonom. [in Slovak].
- [16] Perform large-scale computations using multicore desktops, GPUs, clusters, grids, and clouds (2018) Retrieved Jan 17, 2018 from <https://www.mathworks.com/solutions/parallel-computing>
- [17] MATLAB Distributed Computing Server(2018), Retrieved May 14, 2018 from <https://www.mathworks.com/help/mdce>
- [18] Karelkhan N. (2017) Zhogary onimdi parallel esepteuler klasterin baptau [Setting up high-performance parallel computing cluster]. Proceedings of the XII International Scientific Conference for students and young scholars «Science and education - 2017», Astana, April 14, pp. 139-143. [in Kazak]
- [19] Introduction to Parallel Computing, (2018) Retrieved May 14, 2018 from <https://computing.llnl.gov/tutorials/parallel-comp>

## 8 Authors

**Meruert Serik** professor, doctor of pedagogical sciences, Department of Computer Sciences, Faculty of Computer Sciences, L.N. Gumilyov Eurasian National University, Astana, Kazakhstan. E-mail: [serik\\_meruerts@mail.ru](mailto:serik_meruerts@mail.ru)

Nursaule Karelkhan, doctoral student, Department of Computer Sciences, Faculty of Computer Sciences, L.N. Gumilyov Eurasian National University, Satpayev str 2, 010000 Astana, Kazakhstan. E-mail: [knursaule@mail.ru](mailto:knursaule@mail.ru)

**Ing. Jaroslav Kultán**, PhD, Department of Applied Informatics, Faculty of Economic Informatics, University of Economics in Bratislava, Bratislava, Slovakia, E-mail: [jkultán@gmail.com](mailto:jkultán@gmail.com)

**Zulpykhar Zhandos**, Candidate of pedagogical sciences, Department of Computer Sciences, Faculty of Computer Sciences, L.N. Gumilyov Eurasian National University, Astana, Kazakhstan. E-mail: [astzhan@gmail.com](mailto:astzhan@gmail.com)

Article submitted 2018-10-18. Resubmitted 2019-01-07. Final acceptance 2019-01-17. Final version published as submitted by the authors.