Using cluster parallel computing in the content of information-didactic system

Meruert Serik, Murat Nauruzbaevich Bakiyev, Gulmira Beketovna Balgozhina

L.N. Gumilyov Eurasian National University, Pushkin Street, 11, Astana, 010008, Kazakhstan

Abstract. The article examines questions the developed by the authors, information - didactic system, into the content which are incorporated into the learning process special courses, including a course on parallel computing. And also gives the experimental examples for the implementation of the cluster environment Matlab, also reflected the performance of computing tasks using customized by authors cluster parallel computing.
[Serik M., Bakiyev M.N., Balgozhina G.B. **Using cluster parallel computing in the content of information-didactic system.** *Life Sci J* 2014;11(11s):537-539] (ISSN:1097-8135). http://www.lifesciencesite.com. 122

Introduction

Working in the field of new information technologies in the educational system, the main aspect of our work is to continuously improve the quality of education [1].

In high educational process of the Republic used the developed by us the information - didactic system.

Information - didactic system – is systematically organized collection of information, technical, training and methodological support, inextricably linked with the student as the subject of the educational process [2], which reflects the content of the specialty disciplines of computer science. This article discusses the issues of the contents of this information - didactic system.[3]

In light of the implementation of this system in the learning process specialty "Computer Science" introduced course "Parallel Programming".

The content of special course includes issues such as configuring a cluster, parallel jobs, parallel programming modes, the use of GPU, etc. Improving the training of future professionals in the aspect of parallel programming - current issue today.[4]

Main part

Note that the set of problems that require for their solutions use of powerful computing resources is constantly growing. he given circumstance due to the fact that there have been significant changes in the organization of scientific research. Solutions to complex problems require large computational resources.

In the transition to multi-core architecture for parallel programming becomes relevant tool today.[5]

Using parallel computing enables to expand the scope and research to address the economic, physical, mathematical and other sectoral objectives of parallelization.

For Parallel computing in the educational process we set up cluster parallel computing, which consist of existing in computer classes available computers.

Cluster - a group of computers connected in a local area network and that will work as a single computing resource. [6]

The solution of problems on the traditional sequential computers takes too much time, so the use clusters of parallel computing is optimal possibility of obtaining new effective scientific results. And teaching students to parallelization algorithms is the requirement of today.[7]

Can be distinguished the following basic requirements for cluster management:

- Installation on each computer service "MATLAB Distributed Computing Server".

- English version of the operating system Windows, i.e. will not be here problems with built-in security group, which is in Russian and English languages and different [8].

- Computers of the same type, otherwise the cluster will have a no uniform structure. In this case will be different: a CPU, RAM, speed plots local network;

- Performance, i.e. throughput;

- Convenience, i.e. management system of cluster should be flexible and highly customizable and easy to use;

- Reliability (fault tolerance), i.e. hardware platform and software must work quickly; failover cluster should continue to perform tasks. If execution stops, the system should recover the intermediate state and start with him, rather than the beginning.[9]

Hardware and software requirements for installing the cluster.

Us based on Windows Server was configured Matlab (R2011b) cluster.

For this we must consider the following hardware requirements (in our case), each processor should be at least 2 cores.[10]

For the experiment, we selected 3 computers with processors: Pentium (R) Dual-Core CPU 2.80GHz, Pentium (R) Dual-Core CPU3.50GHz and Core i7 CPU 2.20GHz (GPU NVIDIA, additional computing power GPU Computing).

In our case 3 computers available 12 cores (2 +2 +8). Thus, based on them we have created 12 work processes (including characteristics of the platform can create 16 workflows).

This provides a good opportunity to parallel computing [11, 12].

Memory in this case is respectively: 2 Gb, 2 Gb and 8 Gb.

Basic software requirements: Installing Matlab R2011b and Parallel Computing Toolbox on each computer; initialize and start the service Matlab Distributed Computing Server on each computer.

On all computers operating system was installed with one platform. In our case, running Windows Server 2003 and MATLABR2011b.
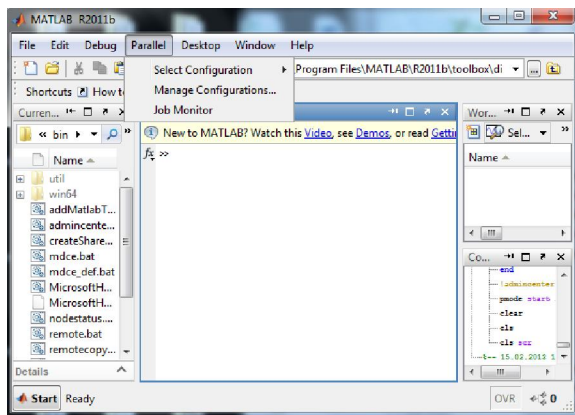
The main window of the parallel MATLAB (Figure 1):

Figure 1. The main window of the parallel MATLAB

All computers must be on the same local network to increase the speed of data transfer between virtual laboratories (Figure 2).

To initialize and start the service Matlab Distributed Computing Server should run the commands: !mdceinstall, !mdcestart. Then, in the management of parallel computing in MatLab should start up scheduler: !start jobmanager –name jm –v [13].
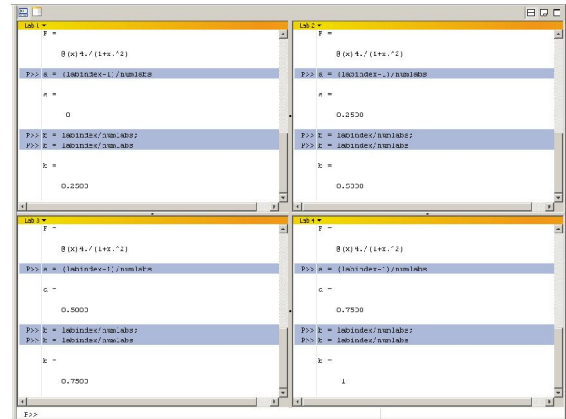

Figure 2 - Parallel command window

Next to accomplish the tasks in the management of parallel computations in parallel MatLab must run scheduler to start the workflow system.

In this case, as mentioned above, the number of working processes is twelve.

The aim of our work is to identify the advantage of parallel matrix multiplication in comparison with the serial multiplication of matrices.

As an example we used the matrix multiplication A (n * n) and the vector Z (n). To perform this task, follow the following steps:

repeat the matrix multiplication algorithm dimension n.

to compile the program serial and parallel matrix multiplication.

check at runtime for different values of n.

You can control the execution of its program through the launch window systems as Matlab (version R2011b and above) using JobMonitor (using the menu Parallel), or from the command line using system commands.[14]

Explicit definition pool if you run in the command window using the Matlab matlabpool [15]. The time and the number of selected processes which are available to the user depend on the predetermined configuration of the parallel. Currently, in our case would be the configuration name mpc, a number of processes - 4 (2 cores per processor). For the experiment, we selected two computer: Pentium (R) Dual-Core CPU 2.8 GHz and Pentium (R) Dual-Core CPU 3.5 GHz.

Conclusions

As a result of codes, i.e. serial and parallel matrix multiplication for different n values were obtained by the following time (Table 1):

Table 1. Data for comparison of serial and parallel matrix multiplication with the number of workers to 4 (in this case coincides with the number of workers cores)

n	Sequential multiplication	Parallel multiplication
300	3.617402	1.319166
1000	147.087110	48.551099

The next step is to use a Core i7 CPU 2.20 GHz with GPU NVIDIA, i.e. with additional computing power GPU Computing.

When you run the m-file with the number of workers matr2.m equal to eight and for different values of n obtained improved results:

Starting matlabpool using the 'local' configuration ... connected to 8 labs.

Elapsed time is 25.433397 seconds.

Sending a stop signal to all the labs ... stopped.

The following table shows a comparison of serial and parallel matrix multiplication with the number of workers equal to 8 (Table 2).

Table 2. Data for comparison of serial and parallel matrix multiplication in an amount equal to 8 workers

n	Sequential multiplication	Parallel multiplication
300	1.396600	0.787286
1000	60.864665	25.433397

The results show that the higher the value n, the less time is spent in a parallel version of multiplication over sequential variant.

Corresponding Author:
Dr. Serik Meruert
L.N. Gumilyov Eurasian National University
Pushkin Street, 11, Astana, 010008, Kazakhstan

References
1. Serik, M., A.K.Alshanov, N.F. Musina K.R. Yesmakhanova, 2010. The realization of open remote metods of education. Comparative education, Teacher Training, Education Policy, School Leadership and Social Inclusion. 8: 463-471.
2. Serik, M., A.K.Alshanov, N.F. Musina K.R. Yesmakhanova, 2010. Ineducation-didactic system as means of improvement of quality of education. Voronezh, pp: 471-472
3. Serik, M., G.B. Balgozhina, 2014. About the android platform, to create educational mobile applications. E-learning in the university and the school.
4. Llamas-Nistal, M., M.J. Fernández-Iglesias, L.E. Anido-Rifon, 2003. Computers and Education: Towards a Lifelong Learning Society. Springer, pp: 287.
5. Hollowell, J., 2011. Moodle as a Curriculum and Information Management System. Packt Publishing Ltd, pp: 308.
6. Voevodin, Vl.V.,V.V. Voevodin, 1999. Analytical Methods and Software Tools for Enhancing Scalability of Parallel Applications. Proc. of Intl. Conf. HiPer'99, Norway, pp: 489-493.
7. Cole, J., H. Foster, 2008. Using Moodle: Teaching with the Popular Open Source Course Management System. O'Reilly Media, Inc., pp: 266.
8. Akhter, Sh., J. Roberts, 2006. Multi-Core Programming. Increasing Performance through Software Multi-threading. Intel Press, pp: 344.
9. Buzzetto-More, N., 2010. The E-portfolio Paradigm: Informing, Educating, Assessing, and Managing with E-portfolios. Informing Science, pp: 328.
10. Serik, M., N.T. Chyndaliev, N.F. Musina, 2012. Work with system commands. USA: International Center for Education and Technology, pp: 130-131.
11. Serik, M., N.T. Chyndaliev and N.F. Musina, 2012. Teaching methods of the BIOS. USA: International Center for Education and Technology, pp: 131-132.
12. Serik, M., G.B. Balgozhina, 2014. Instant messaging application for Smartphone. Life Science Journal; 11(1s): pp: 258-262.
13. Olenev, H.H., R.V. Pechenkin, A.M. Chernecov, 2007. Parallel programming in MATLAB and its applications. Dorodnicyn computing centre of the Russian academy of sciences. Moscow, pp: 120.
14. Distance Education, 2011. General Books LLC, pp: 96.
15. Asanovic, K., 2008. The Parallel Computing Laboratory at U.C.Berkeley: A Research Agenda Based on the Berkeley View. Techn. Rep. UCB/EECS.

7/9/2014