The Ministry of Education and Science of the Russian Federation

Lobachevsky State University of Nizhni Novgorod

Computing Mathematics and Cybernetics faculty

The competitiveness enhancement program
of the Lobachevsky State University of Nizhni Novgorod
among the world's research and education centers

Strategic initiative

Achieving leading positions in the field of supercomputer technology and high-performance computing

Computational mathematics and cybernetics faculty

APPROVED

Dean of the Computational mathematics
and cybernetics faculty

_________________ V. Gergel

"_____"___________________2014.

Course program

PARALLEL PROGRAMMING FOR MULTIPROCESSOR DISTRIBUTED MEMORY SYSTEMS

Bachelor program "Fundamental Informatics and Information Technologies"

Specialization

General part

Professional block

Б.3.Б.12

Degree

Bachelor

Education form

Full-time attendance

Nizhni Novgorod
2014

1. Course objective

The training course covers parallel programming technology designed for development of high-performance implementation of time-consuming algorithms to be executed on the parallel computation systems of cluster architecture.

Lecture sections propose studying of Message Passing Interface technology fundamental principles, structure of MPI library, types of communications between processes, derived data types, virtual topologies, blocking and nonblocking communications in point-to-point and collective modes.

During laboratory practice phases of parallel software development will be reviewed. Among those are: development of serial implementation, as a comparative example, parallel version development, its analysis. Training process is based on the test problems which don't require specific knowledge from particular application domains, except the information from the training course.

The main objective of the course is to study the base concepts of MPI and its practical usage in development of parallel programs for the distributed memory systems.

The following tasks are stated:

1. Mastering the MPI parallel programming technology for distributed memory systems.
2. Consideration of the general approaches to algorithm implementation using MPI.
3. Discussing (within the laboratory works) the examples of efficient implementation of the examined algorithms.

2. Course position in the bachelor program

The course is developed for the 3-d year students and is given in the 5-th semester. This course belongs to the compulsory disciplines of the professional block.

It is expected that students have some basic knowledge in the structured and modular programming. C/C++ is considered as the basic programming language although many of the ideas can be implemented successfully using other programming languages which support parallel programming (e.g. Fortran).

The course is based on the materials developed in UNN with the support of Microsoft in 2005-2006 years. (http://www.hpcc.unn.ru/mskurs/cs338_pp_index.htm). In 2014, a partial modification for the course materials was made as a part of the "5-100 project" by Alexander Sysoyev. The changes mainly consist in the detailed description of MPI-2 and MPI-3 standard extensions. In addition lecture materials were restructured according to practical experience of its usage on the Computational Mathematics and Cybernetics faculty educational programs and advanced trainings held in Moscow, Saint-Petersburg, Nizhny Novgorod, Perm, Belgorod and other cities. In addition, the laboratory works were translated into English.

3. Learning outcomes and requirements

In the framework of this course, the following **competencies** are formed:

- Possessing the general culture of thinking, the ability to perceive, compile and analyze information (General Competency 1 - GC1). Students will be able to:
    - construct oral and written arguments in a logical and clear manner.
- The ability of intellectual, cultural, moral, physical, and professional self-development and self-improvement (GC 2). Students will be able to:
    - constantly improve their professional and cultural level.
- The ability to understand and apply in practice the theory of information as a fundamental scientific basis of information technology (Professional Competency 1 – PC 1). Students will be able to:
    - understand the content side of the information process, know the techniques for sending, receiving, processing, analyzing and storing data.
- The ability to understand, develop and apply modern information technology (PC 4). Students will be able to:
    - understand the concepts and implement the functionality of the following core technologies:
        - at the level of technological literacy:
            - computer systems architecture;
        - at the level of in-depth knowledge:
            - basic programming;
            - parallel and distributed computing.
    - develop and use professionally the software for supporting information systems and processes, to be able to use modern instrumental computing equipment.
- The ability to conduct research (PC 5). Students will be able to:
    - develop new algorithmic and methodological and technological solutions;
    - collect, process and interpret the data of modern research necessary to develop approaches, decisions and conclusions on appropriate scientific and professional issues.

- The ability to conduct analytical activities (PC 8). Students will be able to:
    - analyze and select modern technologies and methodologies for implementing an information system.

As the result of education graduates from the course:

Learners who successfully completed the training course

Will know usage of MPI technologies for development of parallel programs for computing systems with distributed memory.

Will Be Able To implement parallel algorithms for solution of standard problems of computational mathematics (matrix calculations, sorting, graph processing).

Can use system software (compiler, MPI library) for development of parallel programs for distributed memory systems.

4. Course outline

The course consists of 1 credit, 36 hours, including 8 lecture hours and 8 practice hours. Practice classes can be held as lab works (students carry out assignments step-by-step under supervision) or master class (supervisor demonstrates and explains step-by-step solutions). 20 hours are allocated for individual work. The authors encourage additional work.

4.1. Course outline

Course outline is as follows:

#	Module	Semester	Week	Module type				Assessment
				Lecture	Seminar	Lab	Individual work	
1	The Fundamentals of MPI	5	1-2	2	–	–	2	Test
2	Collective and Point-to-Point Communications		3-4	2	–	–	2	Test
3	Derived Data Types, Communicators and Virtual Topologies		5-6	2	–	–	2	Test
4	MPI Extensions		7-8	2	–	–	2	Test
5	Parallel Algorithms of Matrix-Vector Multiplication		9-10	–	–	2	3	Test
6	Parallel Algorithms of Matrix Multiplication		11-12	–	–	2	3	Test
7	Parallel Methods of Solving the Linear Equation Systems		13-14	–	–	2	3	Test
8	Parallel Algorithms of Graph Processing		15-16	–	–	2	3	Test
	TOTAL:			**8**	–	**8**	**20**	**Final assessment form – exam**

4.2. Course description

Course content is as follows:

1. The Fundamentals of MPI

Introduction. MPI: Basic concepts and definitions. The fundamentals of MPI. MPI program initialization and termination. Determining the number and the rank of the processes. Message send/receive operations. Evaluating of MPI program execution time. The first MPI parallel program. Introduction into collective data communication.

2. Collective and Point-to-Point Communications

Collective communications: data broadcasting, reduction operations, example (calculating the constant π), scattering and gathering, example (calculating the inner product), all to all communications, computation synchronization. Communications between two processes: communication modes, nonblocking communications, simultaneous sending and receiving.

3. Derived Data Types, Communicators and Virtual Topologies

Derived data types in MPI: type map, the methods of constructing, declaring and deleting, data packing. Groups of processes and communicators: managing groups, managing communicators. Virtual topologies: Cartesian topologies (grids), graph topologies.

4. MPI Extensions

Nonblocking collective operations: general description, data broadcasting, reduction operations, scattering and gathering, all to all communications, computation synchronization. Process creation and management: general description, the dynamic process model, process management, establishing connections.

5. Parallel Algorithms of Matrix-Vector Multiplication

Matrix-vector multiplication problem statement. Serial algorithm of matrix-vector multiplication and its implementation. Analysis of information dependencies. Computation organization. Parallel algorithm and its implementation.

6. Parallel Algorithms of Matrix Multiplication

Matrix multiplication problem statement. Serial algorithm of matrix multiplication and its implementation. Analysis of information dependencies. Computation organization. Parallel algorithm and its implementation.

7. Parallel Algorithms of Solving the Linear Equation Systems

The problem statement of solving the linear equation systems. Gauss algorithm. Gaussian elimination stage, back substitution stage. Serial implementation. Analysis of information dependencies. Parallel algorithm. The pivot row selection. Parallel program implementation.

8. Parallel Algorithms of Graph Processing

The shortest path problem statement. Serial Floyd algorithm and its implementation. Analysis of information dependencies. Parallel Floyd algorithm and its implementation

5. Learning technologies

During course we use the following learning technologies: lectures, lab works, individual work, assessment techniques. Powerpoint presentations for all lectures and practical lessons are used.

6. Individual work and assessment techniques

Individual work consists of mastering theoretical and practical material according to the given references, solving practical problems, and answering on the given questions. Individual work can be done in both classes and at home.

Control of individual work is performed by electronic tests. In the end of the program there is a final test.

6.1 Assessment forms

Monitoring of progress in studies is performed by tests in class that consist of assignments from the list of questions and practical problems (given below).

The final attestation is done based on the results of the final test. This test includes questions from all sections of the course.

6.2 Individual work: Practical problems

Module 1

1. Develop a program for finding the minimum (maximum) value of the vector elements.
2. Develop a program for computing the inner product of two vectors.
3. Develop a sample program for collective operation MPI_Bcast.
4. Develop a sample program for collective operation MPI_Reduce.
5. Develop an implementation of collective operation MPI_Bcast using point-to-point communications. Compare the execution time of the developed version to the function MPI_Bcast.

Module 2

6. Develop a sample program for each collective operation available in MPI.
7. Develop the implementations of collective operations using point-to-point communications. Carry out the computational experiments and compare the execution time of the developed programs to the functions of MPI for collective operations.
8. Develop a program, carry out the experiments and compare the results for different algorithms of data gathering, processing and broadcasting (the function MPI_Allreduce()).

Module 3

9. Develop a sample program for each method of constructing the derived data types available in MPI.

10. Develop a sample program using data packing and unpacking functions. Carry out the experiments and compare the results to the results obtained in case of the use of the derived data types.
11. Develop the derived data types for the rows, columns and diagonals of matrices.
12. Develop a sample program for the Cartesian topology.
13. Develop a sample program for a graph topology.
14. Develop subprograms for creating a set of additional virtual topologies (a star, a tree, etc.).

Module 4

15. Develop a sample program for each method of nonblocking collective operations.
16. Develop a sample program using additional process in MPI program. Possible scheme to implement is "master-workers".

Module 5

17. Study the parallel algorithm of matrix-vector multiplication based on column-wise block-striped matrix partitioning. Develop a program implementation of this algorithm.
18. Study the parallel algorithm of matrix-vector multiplication based on chessboard block matrix partitioning. Develop a program implementation of this algorithm.

Module 6

19. Modify the developed Fox algorithm implementation using the derived MPI data type for broadcasting and gathering matrix blocks.
20. Study the parallel algorithm of matrix multiplication based on block striped matrix partitioning. Develop a program implementation of this algorithm.
21. Study the Cannon parallel algorithm of matrix multiplication based on chessboard block matrix partitioning. Develop a program implementation of this algorithm.

Module 7

22. Study the conjugate gradient method of solving the linear equation systems.
23. Develop the serial and the parallel variants of the method.

Module 8

24. Study the Prim algorithm for finding the minimum spanning tree. Develop the program, which implements this algorithm.
25. Study the Dejkstra method for solving the problem of finding the shortest path from one of the graph vertices to the other. Develop the program, which implements this algorithm.

6.4 Assessment criteria

Perfect	Correct answers on >95% of the number of test questions
Excellent	Correct answers on 80-95% of the number of test questions

Very good	Correct answers on 70-79% of the number of test questions
Good	Correct answers on 60-69% of the number of test questions
Satisfactory	Correct answers on 50-59% of the number of test questions
Unsatisfactory	Correct answers on 25-49% of the number of test questions
Bad	Correct answers on <25% of the number of test questions

7. References

1. Quinn, M.J. (2004). Parallel Programming in C with MPI and OpenMP. – New York, NY: McGraw-Hill.
2. Grama, A., Gupta, A., Kumar V. (2003, 2nd edn.). Introduction to Parallel Computing. – Harlow, England: Addison-Wesley.
3. Pacheco, P. (1996). Parallel Programming with MPI. - Morgan Kaufmann.
4. Andrews, G.R. (2000). Foundations of Multithreaded, Parallel, and Distributed Programming.. – Reading, MA: Addison-Wesley.
5. Snir, M., Otto, S., Huss-Lederman, S., Walker, D., Dongarra, J. (1996). MPI: The Complete Reference. – MIT Press, Boston, 1996.
6. Group, W., Lusk, E., Skjellum, A. (1999). Using MPI – 2nd Edition: Portable Parallel Programming with the Message Passing Interface (Scientific and Engineering Computation). – MIT Press.
7. Group, W., Lusk, E., Thakur, R. (1999). Using MPI-2: Advanced Features of the Message Passing Interface (Scientific and Engineering Computation). – MIT Press.
8. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein C. (2009). Introduction to Algorithms, 3rd Edition. – The MIT Press.
9. Barker, M. (Ed.) (2000). Cluster Computing White Paper at http://arxiv.org/ftp/cs/papers/0004/0004014.pdf.
10. MPI: A Message-Passing Interface Standard. Version 3.0. – Message Passing Interface Forum. September 21, 2012 – http://www.mpi-forum.org/docs/mpi-3.0/mpi30-report.pdf.

8 Course support

The following software and hardware are used during course study:

Hardware

“Lobachevsky” supercomputer with the Intel Xeon Phi coprocessors is used.

Software

Microsoft Visual Studio 2005 or newer, MPI implementation (MPICH, MS MPI, OpenMPI, Intel MPI) is used.

9 Author

English version of the course is developed by A. Sysoyev.

Senior lecturer ____ Alexander Sysoyev

Course program is discussed by Software Department members.

«______» _________________ 2014; Document # _______

The head of the Software Department, prof. ____________________________ Roman Strongin

Course program is approved by methodical commission of Computational Mathematics and Cybernetics Faculty of UNN,

«______» _________________ 2014; Document # _______

The head of the commission __________________________________ Natalia Shestakova