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

**Introduction to MPI**

Bachelor program "Fundamental Informatics and Information Technologies"

Specialization

**General part**

**Professional block**

**Elective disciplines**

**Б3.Б12**

Degree

**Bachelor**

Education form

**Full-time attendance**

Nizhni Novgorod

2014

## 1. Purposes of mastering the discipline

The need for solving complex applied problems requiring large-scale computing and essentially limited maximum performance of classical computers (von Neumann machines) gave rise to multiprocessor systems. The use of such systems helps to a large extent increase the computer performance at any existing level of hardware development. However, this requires parallel generalization of the traditional sequential pattern of problem solving. Thus, for multiprocessor systems, numerical methods should be developed as systems of parallel interacting processes allowing for running on independent processors. The applicable algorithmic languages and systemware must enable development of parallel programs and ensure synchronization and mutual exclusion of asynchronous processes etc.

Supercomputer technologies and high performance computing based on parallel systems are becoming an increasingly important technical development factor; they are becoming omnipresent.

The mentioned issues form the subject matter of this course. The purpose of this course is to study one of MPI technologies of parallel programming. This course also studies mathematical models, methods and technologies of parallel programming for multicore and multiprocessor computers to the extent ensuring a successful start in the field of parallel programming.

The proposed set of knowledge and skills forms a theoretical basis for methods of complex program development and includes such topics as purposes and objectives of parallel data processing , construction principles for parallel computing systems, modeling and analysis of parallel computations, development principles for parallel programs and algorithms, systems for parallel programming and parallel numerical algorithms for solving a number of computing mathematics problems.

## 2. Discipline in the concentration program

This course is intended for third year students and is taught in the 5$^{th}$ semester. This is a base discipline of the professional block.

Being originally graduate-oriented, this course is also intended for teachers and post-graduates in the field of parallel programming. This course involves basic knowledge and skills in structural, modular and (in some sections) object-oriented programming. The basic programming language is C/C++. Many ideas may be successfully used for another programming language that supports parallelism, especially Fortran.

## 3. Requirements to discipline mastery

Learning this discipline involves forming of the following competencies:

- ability to acquire, generalize and analyze the information (GC1) to the following extent:
    - ability to speak and write in a logical, reasoned and clear way;
- capacity for intellectual, cultural, moral, physical and professional development and self-improvement (GC2) to the following extent:
    - ability to persistently pursue objective subject to moral and legal norms and obligations;
    - capacity for continuous improvement of their professional and cultural level;
- ability to understand and put into practice the information theory as the fundamentals of information technologies (PC1)

- Readiness to join the professional community (PC2)
- Ability to understand and use the existing mathematical tools and laws of natural science (PC3)
- Ability to understand, develop and use state-of-the-art information technologies (PC4)
- R&D capabilities (PC5)
- Management capabilities (PC6)
- Design capabilities (PC7)
- Analytical capabilities (PC8)

As a result, students should:

**Know** parallel computation methods for solving problems of computing mathematics (matrix computations), basic approaches to parallel programming, parallel program development technologies, their basic functionality and basic principles of use.

**Be able** to construct parallel program execution models, evaluate parallel computing efficiency, analyze computation complexity and possibility of algorithm parallelization, use parallel program general development schemes to implement their own algorithms, evaluate main parameters of the resulting parallel programs such as speedup, efficiency and scalability. Be able to develop, debug and optimize parallel programs for traditional and heterogeneous clusters/HPC systems.

**Have** parallel program development and debugging skills for MPI-based clusters and multicore systems.

## 4. Discipline structure and contents

The total discipline intensity is 3 credits/108 hours. Form of testing for the 5th semester: examination.

### 4.1. Discipline structure

| No. | Section<br>Disciplines | Semester | Semester week | Types of academic work including students' independent work and its intensity (in hours) | | | | Forms of progress control (per semester week)<br>Mid-term examination (per semester) |
|---|---|---|---|---|---|---|---|---|
| | | | | Lecture | Practice | Lab. | Independent work. | |
| 1 | Introduction | 5 | 1 | 2 | - | - | 2 | |
| 2 | Basic Notions and Definitions | | 2 | 2 | - | - | 2 | |
| 3 | MPI-Based Parallel Programming Basic Data Transmission Operations | | 4 | 2 | - | 3 | 2 | |
| 4 | MPI-Based Parallel Programming Collective Data Transmission Operations | | 5 | 2 | - | 3 | 2 | |
| 5 | Principles of Parallel Method Development | | 6-7 | 2 | - | - | 2 | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 6-7 | Parallel Methods for Matrix-Vector Multiplication | | 8-9 | 4 | - | 4 | 4 | |
| 8-9 | Parallel Methods for Matrix Multiplication | | 10-11 | 4 | - | 4 | 4 | |
| 10 | Parallel Computations for Systems with Distributed Memory | | 12 | 2 | - | - | - | |
| 11 | Parallel Computation Modeling and Analysis | | 13-14 | 4 | - | 4 | 2 | |
| 12 | Estimation of Communication Complexity for Parallel Algorithms. | | 15-16 | 4 | - | 4 | 2 | |
| 10 | | | 17 | 2 | | | | Examination (36 hours) |
| | Final examination | | | | | | | Examination (36 hours) |
| | TOTAL: | | | 28 | - | 22 | 22 | |

**4.2. Contents of discipline sections**

The course consists of several sections with lectures and practical classes dedicated to studying MPI parallel programming models, methods and technologies. This course is oriented to the use of cluster systems and supercomputers based on traditional CPUs.

The course contains educational materials sufficient for a successful start in the field of parallel programming for state-of-the-art HPC systems. As part of this approach, lectures account for a significant part of this course and has the following contents:

*Lecture 1. Introduction* The lecture emphasizes the importance of parallel computations and describes the general course structure.

*Lecture 2. Basic Notions and Definitions* The lecture introduces the notion of parallel computations. It describes basic efficiency parameters and demonstrates applicability of such parameters by the example of the number sequence summation problem.

*Lecture 3. MPI-Based Parallel Programming.* Basic operations. The lecture gives basic notions and definitions related to MPI. It also lists the minimum set of functions required for parallel program development.

*Lecture 4. MPI-Based Parallel Programming*. Collective operations. The lecture describes an extended set of operations ensuring more efficient data exchange between computation processes. It also gives an application example for collective operations.

*Lecture 5. Principles of Parallel Method Development*. The lecture describes parallel method development stages and gives development examples.

*Lectures 6-7. Parallel Methods for Matrix-Vector Multiplication* The lecture is dedicated to the basic parallelization methods to solve the matrix-vector multiplication problem.

*Lectures 8-9. Parallel Methods for Matrix Multiplication*. The lecture is dedicated to the basic parallelization methods to solve the matrix multiplication problem.

*Lecture 10. Parallel Computation for Systems with Distributed Memory.* This lecture describes a classification of computer systems and introduces the notion of parallelism for distributed memory systems.

*Lecture 11. Parallel Computation Modeling and Analysis* The lecture is dedicated to the basic theory of parallel computation modeling and analysis. It introduces the notion of "operations-operands" graph and describes a number ways to estimate parallel method efficiency. Parallel computation modeling and analysis is illustrated by π computation and the finite difference method.

*Lecture 12. Estimation of Communication Complexity for Parallel Algorithms.* The lecture is dedicated to the issues of interprocessor communication time estimation. It compares the runtimes for models and experiments.

### 4.3. Practice topics

| Practice topic |
|---|
| Parallel Hello World. Launch of MPI programs. |
| Computation of scalar products of vectors with manual job assignment. Efficiency measurements. |
| Pi computation. Parallel sorting. |
| Solving matrix multiplication problem. |

## 5. Educational technology

Teaching is based on such educational technologies as lectures, practical classes, seminars (problem-oriented, design, discussion, training and organizational ones) and extramural independent work. This involves the use of project method, information technologies, testing, e-learning tools and web browsing. Lectures include MS Powerpoint presentations. These presentations are used for conducting classes.

## 6. Teaching and learning support of students' independent work. Grading tools for routine progress control and mid-term proficiency examinations.

Independent work consists in familiarization with theory based on textbooks and monographs indicated in the references list, solving practical problems, preparation for seminars, reports, presentations and e-tests in the course of learning, and answering self-test questions. Independent work may take place both in reference rooms and at home.

### 6.1. Practice

Each practical class involves the following activities:

1. Literature search and studying of the algorithm proposed for implementation.
2. Sequential algorithm implementation in C/C++. Testing, debugging, solving test problems, efficiency analysis.
3. Development of the parallel method version based on MPI. Testing, debugging, solving test problems, efficiency analysis.
4. Comparative analysis of performance/efficiency.
5. Activity report preparation.

**List of algorithms proposed for implementation:**

*Matrix multiplication*

1. Use of horizontal partitioning
2. Cannon's algorithm
3. Fox algorithm
4. DNS algorithm

*Matrix transposing*

*Solving linear system*

1. Gaussian method
2. Iterative methods (Jacobi method)
3. Iterative methods (Seidel method)

For each section, the set of algorithms may be modified/extended at the teacher's discretion.

## 6.2. Self test

1. How are the concepts "speedup" and "efficiency" defined?
2. Is it possible to attain superlinear speedup?
3. What is the contradictoriness of the speedup and efficiency characteristics?
4. What algorithm is a scalable one? Give examples of methods with various scalability levels
5. What are the main ways to ensure parallelism?
6. In what way can parallel computer systems differ?
7. What is the basis of Flynn's taxonomy?
8. What is the principle of subdividing multiprocessor systems into multiprocessors and multicomputers?
9. What system classes exist for multiprocessors?
10. What are advantages and disadvantages of symmetric multiprocessors?
11. What system classes exist for multicomputers?
12. What are advantages and disadvantages of cluster systems?
13. What data communication network topologies are most widely used to construct multiprocessor systems?
14. What are the specific features of data communication networks for clusters?
15. What are the main characteristics of the data communication networks?
16. What system platforms may be used for the purposes of building clusters?
17. How is the "operations-operands" model defined?
18. How is the schedule for the distribution of computations among processors defined?
19. How is the time of parallel algorithm execution defined?
20. What schedule is optimal?
21. How can the minimum possible time of problem solving be defined?
22. What is a paracomputer? What can this concept be useful for?
23. What estimates should be used as the characteristics of the sequential problem solving time?
24. How to define the minimum possible time of parallel problem solving according to "operands-operations" graph?
25. What dependences may be obtained for parallel problem solving time if the number of processor being used is increased or decreased?
26. What number of processors corresponds to the parallel algorithm execution time comparable in the order with the estimates of minimum possible time of problem solving?
27. How are the concepts "speedup" and "efficiency" defined?

28. Is it possible to attain superlinear speedup?
29. What is the contradictoriness of the speedup and efficiency characteristics?
30. How is the concept of computation cost defined?
31. What is the concept of the cost-optimal algorithm?
32. What does the problem of parallelizing a sequential algorithm of the numeric values summation lie in?
33. What is the essence of the summation cascade scheme? What is the aim of considering the modified version of the scheme?
34. What is the difference between the speedup and efficiency characteristics for the discussed versions of the summation cascade scheme?
35. What is the parallel algorithm of all the partial sums computation of a numeric value sequence?
36. How is Amdahl's law formulated? Which aspect of parallel computation does it allow to take into account?
37. What suppositions are used to ground the Gustafson-Barsis's law?
38. How is the isoefficiency function defined?
39. What minimum set of operations of sufficient for the organization of parallel computations in the distributed memory systems?
40. Why is it important to standardize message passing tools?
41. How can a parallel program be defined?
42. What is the difference between the concepts of "process" and "processor"?
43. What minimum set of MPI functions is necessary for parallel software development?
44. How are the messages being passed described?
45. How do we organize the reception of messages from specific processes?
46. How can one determine runtime for MPI-based software?
47. What is the difference between point-to-point and collective data transmission operations?
48. Which MPI function provides transmitting data from a process to all the processes?
49. What is meant by reduction?
50. In what cases should we apply barrier synchronization?
51. What collective data transmission operations are supported in MPI?
52. What is a communicator in MPI?
53. What can new communicators be created for?
54. What is a virtual topology in MPI?
55. What types of virtual topologies are supported in MPI?
56. What may virtual topologies appear to be useful for?

### 6.3. Grading criteria

Perfect - complete mastery of basic and additional subject matter without errors or mistakes, ability to solve non-conventional problems; competencies (parts of competencies) related to the discipline have been mastered in an integrated manner exceeding obligatory requirements. A sustainable system of competences has been formed, relation to mastering other competences is evident;

Excellent - complete mastery of basic and additional subject matter without errors or mistakes, competencies (parts of competencies) have been mastered completely at a high level; a sustainable system of competencies has been formed;

Very good - sufficient mastery of basic subject matter with insignificant errors, ability to solve standard problems, competencies (parts of competencies) have been mastered completely;

Good - mastery of basic materials with noticeable errors, competencies (parts of competencies) have been mastered for the most part;

Satisfactory - minimal mastery of the subject matter with errors and mistakes, ability to solve principal problems, competencies (parts of competencies) have been mastered at the minimum level required to achieve basic learning objectives;

Unsatisfactory - insufficient knowledge of the subject matter, additional training required, competencies (parts of competencies) related to this discipline have not been mastered in a manner sufficient for achieving basic learning objectives;

Poor - no mastery of the subject matter; respective competencies have not been mastered.

**7. Teaching, learning and information support**


1. Andrews, G. R. (2000). Foundations of Multithreaded, Parallel, and Distributed Programming.. – Reading, MA: Addison-Wesley.
2. Bertsekas, D.P., Tsitsiklis, J.N. (1989) Parallel and distributed Computation. Numerical Methods. - Prentice Hall, Englewood Cliffs, New Jersey.
3. Buyya, R. (Ed.) (1999). High Performance Cluster Computing. Volume1: Architectures and Systems. Volume 2: Programming and Applications. - Prentice Hall PTR, Prentice-Hall Inc.
4. Kahaner, D., Moler, C., Nash, S. (1988). Numerical Methods and Software. – Prentice Hall.
5. Foster, I. (1995). Designing and Building Parallel Programs: Concepts and Tools for Software Engineering. Reading, MA: Addison-Wesley.
6. Dongarra, J.J., Duff, L.S., Sorensen, D.C., Vorst, H.A.V. (1999). Numerical Linear Algebra for High Performance Computers (Software, Environments, Tools). Soc for Industrial & Applied Math/
7. Blackford, L. S., Choi, J., Cleary, A., D'Azevedo, E., Demmel, J., Dhillon, I., Dongarra, J. J., Hammarling, S., Henry, G., Petitet, A., Stanley, D. Walker, R.C. Whaley, K. (1997). Scalapack Users' Guide (Software, Environments, Tools). Soc for Industrial & Applied Math.
8. Wilkinson, B., Allen, M. (1999). Parallel programming. – Prenrice Hall.
9. Pacheco, P. (1996). Parallel Programming with MPI. - Morgan Kaufmann.
10. Gropp, W., Lusk, E., Skjellum, A. (1999a). Using MPI - 2nd Edition: Portable Parallel Programming with the Message Passing Interface (Scientific and Engineering Computation). - MIT Press.
11. Gropp, W., Lusk, E., Thakur, R. (1999b). Using MPI-2: Advanced Features of the Message Passing Interface (Scientific and Engineering Computation). - MIT Press.
12. Snir, M., Otto, S., Huss-Lederman, S., Walker, D., Dongarra, J. (1996). MPI: The Complete Reference. - MIT Press, Boston, 1996.
13. Chandra R., Dagum L., Kohr D., Maydan D., McDonald J., Melon R. Parallel Programming in OpenMP. – Morgan Kaufmann Publishers, 2000.
14. Grama, A., Gupta, A., Kumar V. Introduction to Parallel Computing. – Harlow, England: Addison-Wesley, 2003.
15. Quinn M. J. Parallel Programming in C with MPI and OpenMP. – New York, NY: McGraw-Hill, 2004.
16. Fengguang S., Tomov S., Dongarra J. Efficient Support for Matrix Computations on Heterogeneous Multi-core and Multi-GPU Architectures. – University of Tennessee

Computer Science Technical Report, UT-CS-11-668, (also Lawn 250), 2011. [http://icl.cs.utk.edu/news_pub/submissions/lawn250.pdf]

**8. Inventory**

To conduct classes in this discipline, the following software and hardware is required.

*Hardware*

For MPI classes, a computing cluster/mini cluster is desirable. In absence of cluster systems, several PCs in the local network can be used.

*Software*

For the purposes of this course, the following software is necessary:

- OC Microsoft Windows XP/7 or Linux
- Microsoft Visual Studio 2008/2010
- A MPI implementation (e. g. MPICH)

Author(s): Professor _____________ V.P. Gergel

Assistant _____________ Ye. A. Kozinov

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