**Federal State Autonomous Educational Institution of Higher Education "Moscow Institute of Physics and Technology
(National Research University)"**

**APPROVED
Head of the Phystech School of Applied Mathematics and Informatics
A.M. Raygorodskiy**

Work program of the course (training module)

course:	Parallel and Disrtibuted Computing I/Параллельные и распределённые вычисления I
major:	Information Science and Computer Engineering
specialization:	Computer Science/Информатика
	Phystech School of Applied Mathematics and Informatics
	Chair of Algorithms and Programming Technologies
term:	3
qualification:	Bachelor

Semester, form of interim assessment: 5 (fall) - Grading test

Academic hours: 60 AH in total, including:
lectures: 30 AH.
seminars: 0 AH.
laboratory practical: 30 AH.

Independent work: 120 AH.

In total: 180 AH, credits in total: 4

Author of the program: V.V. Yakovlev, candidate of physics and mathematical sciences, head of chair

The program was discussed at the Chair of Algorithms and Programming Technologies 21.05.2020

Annotation

In this course, students will become familiar with:

- such approaches to the development of parallel programs as MPI, OpenMP;
- computing on GPUs using CUDA.
- a device for distributed queues and storages (ZooKeeper, Kafka, Cassandra);
- the main frameworks for processing big data and their internal structure.

The course is practice-oriented. Programming is assumed both at seminars and as part of homework.

1. Study objective

Purpose of the course

To acquaint students with the basics of multiprocessor computing systems and give practical experience in working with such systems. The course consists of two modules, dedicated to parallel and distributed systems, respectively. The first module examines systems in almost "ideal" conditions, where computing nodes and connections between them are reliable and fast. The second module examines ways to build reliable systems from unreliable components.

Tasks of the course

During the course, students gain practical skills in working with both parallel and distributed systems.

2. List of the planned results of the course (training module), correlated with the planned results of the mastering the educational program

Mastering the discipline is aimed at the formation of the following competencies:

Code and the name of the competence	Competency indicators
Gen.Pro.C-1 Apply fundamental knowledge acquired in the physical and mathematical fields and/or natural sciences and use it in professional settings	Gen.Pro.C-1.1 Analyze the task in hand, outline the ways to complete it

3. List of the planned results of the course (training module)

As a result of studying the course the student should:

know:

- Types and classification of multiprocessor computing systems;
- principles of building distributed data warehouses;
- principles of building distributed ecosystems (Hadoop/Spark);
- the difference between (single-) server databases and distributed databases;
- model of asynchronous computation and relationship with the degree of transaction isolation;
- the Fisher-Lynch-Paterson theorem (FLP-theorem);
- know the basic principles of Paxos/Raft;
- know time synchronization algorithms (NTP, Cristian's Algorithm)
- standard distributed computing tasks (Multicasts, Failure Detectors, Membership, Consensus, RSM)

be able to:

- Use the library for parallel computations OpenMP;
- use the library for parallel computations MPI;
- use the distributed file system HDFS;
- use a distributed computing framework Hadoop;
- use the distributed data storage Hive;
- be able to use the primitives of distributed computing Lamport Timestamps, Vector Clocks
- solve the problem of consensus in a synchronous system;
- use Paxos/Raft algorithms.

master:

- Skills in working with multiprocessor computing systems (parallel and distributed computing systems in particular)
- horizons in choosing an architectural solution to the task.

4. Content of the course (training module), structured by topics (sections), indicating the number of allocated academic hours and types of training sessions

4.1. The sections of the course (training module) and the complexity of the types of training sessions

№	Topic (section) of the course	Types of training sessions, including independent work			
		Lectures	Seminars	Laboratory practical	Independent work
1	Distributed computing theory (transaction isolation levels, CRDT, CAP, FLP, Paxos, Raft)	15		15	60
2	GPU computing. CUDA technology	15		15	60
AH in total		30		30	120
Exam preparation		0 AH.			
Total complexity		180 AH., credits in total 4			

4.2. Content of the course (training module), structured by topics (sections)

Semester: 5 (Fall)

1. Distributed computing theory (transaction isolation levels, CRDT, CAP, FLP, Paxos, Raft)

- Classification of multiprocessor computing systems, failure models;
- Difference between parallel computing and distributed computing;
- Databases: ACID, isolation levels. Distributed databases: CAP, CRDT. AP and CP systems;
- Algorithms of time synchronization (NTP, Cristian's Algorithm);
- standard distributed computing tasks (Multicasts, Failure Detectors, Membership, Consensus, RSM);
- Fisher-Lynch-Paterson theorem (FLP-theorem);
- Basic principles of Paxos / Raft;

2. GPU computing. CUDA technology

GPU architecture.
Device (graphics processing unit) and Host. Data exchange between them.
Optimization of computing on GPUs.

5. Description of the material and technical facilities that are necessary for the implementation of the educational process of the course (training module)

Classroom equipped with a multimedia projector and screen.

6. List of the main and additional literature, that is necessary for the course (training module) mastering

Main literature

Additional literature

7. List of web resources that are necessary for the course (training module) mastering

Not used

8. List of information technologies used for implementation of the educational process, including a list of software and information reference systems (if necessary)

For practical training:
Computer class. Each computer must have Internet access and software for connecting to remote servers.

9. Guidelines for students to master the course

Successful mastering of the course requires a lot of independent student work. The course program contains the minimum required time for a student to work on a topic. Independent work includes:

- study of educational material (based on lecture notes, educational and scientific literature),
preparing answers to questions for self-study,
proof of individual statements, properties;
- preparation for practical training, 6 individual homework assignments.

Interim control of knowledge is carried out in the form of written surveys (mini-tests) on theory.

SUPPLEMENT

Assessment funds for course (training module)

major: Information Science and Computer Engineering
specialization: Computer Science/Информатика
Phystech School of Applied Mathematics and Informatics
Chair of Algorithms and Programming Technologies
term: 3
qualification: Bachelor

Semester, form of interim assessment: 5 (fall) - Grading test

Author: V.V. Yakovlev, candidate of physics and mathematical sciences, head of chair

1. Competencies formed during the process of studying the course

Code and the name of the competence	Competency indicators
Gen.Pro.C-1 Apply fundamental knowledge acquired in the physical and mathematical fields and/or natural sciences and use it in professional settings	Gen.Pro.C-1.1 Analyze the task in hand, outline the ways to complete it

2. Competency assessment indicators

As a result of studying the course the student should:

know:

- Types and classification of multiprocessor computing systems;
- principles of building distributed data warehouses;
- principles of building distributed ecosystems (Hadoop/Spark);
- the difference between (single-) server databases and distributed databases;
- model of asynchronous computation and relationship with the degree of transaction isolation;
- the Fisher-Lynch-Paterson theorem (FLP-theorem);
- know the basic principles of Paxos/Raft;
- know time synchronization algorithms (NTP, Cristian's Algorithm)
- standard distributed computing tasks (Multicasts, Failure Detectors, Membership, Consensus, RSM)

be able to:

- Use the library for parallel computations OpenMP;
- use the library for parallel computations MPI;
- use the distributed file system HDFS;
- use a distributed computing framework Hadoop;
- use the distributed data storage Hive;
- be able to use the primitives of distributed computing Lamport Timestamps, Vector Clocks
- solve the problem of consensus in a synchronous system;
- use Paxos/Raft algorithms.

master:

- Skills in working with multiprocessor computing systems (parallel and distributed computing systems in particular)
- horizons in choosing an architectural solution to the task.

3. List of typical control tasks used to evaluate knowledge and skills

Not provided.

4. Evaluation criteria

1. Distributed computing tools.
2. Description of the HDFS distributed file system, fault tolerance and scalability.
3. HDFS pros and cons.
4. Secondary Namenode assignment in HDFS.
5. The process of reading and writing data in HDFS.
6. Hadoop cluster structure. Server roles. How fault tolerance is provided.
7. The process of executing a task on a cluster. Distributed cache.
8. Formal model of MapReduce computation.
9. Scopes of Writable and WritableComparable.
10. Types of comparators in MapReduce.
11. What stages of the MapReduce application should be implemented by the developer.
12. MapReduce: Combiner, Comparator, Partitioner.

excellent

10 comprehensive, systematized, deep knowledge of the curriculum of the discipline and the ability to confidently apply them in practice when solving specific problems, free and correct justification of decisions made;

9 systematic, deep knowledge of the curriculum of the discipline and the ability to confidently apply them in practice when solving specific problems, the correct justification of decisions made;

8 deep knowledge of the curriculum of the discipline and the ability to apply them in practice when solving specific problems, the correct justification of decisions made;

good

7 firmly knows the material, correctly and essentially sets out it, knows how to apply the knowledge gained in practice, but admits some inaccuracies in the answer or in solving problems;

6 knows the material, correctly presents it, knows how to apply the acquired knowledge in practice, but admits some inaccuracies in the answer or in solving problems;

5 knows the basic material, correctly presents it, knows how to apply the knowledge gained in practice, but admits inaccuracy in the answer or in solving problems;

satisfactorily

4 fragmented, fragmented nature of knowledge, insufficiently correct wording of basic concepts, violation of logical sequence in the presentation of program material, but at the same time he owns the main sections of the curriculum necessary for further training and can apply the acquired knowledge in the standard situation;

3 the nature of knowledge is sufficient for further training and can apply the acquired knowledge on the model in a standard situation;

unsatisfactory

2 does not know most of the main content of the curriculum of the discipline, makes gross errors in the wording of the basic concepts of the discipline and does not know how to correctly use the knowledge gained in solving typical practical problems.

1 does not know the wording of the basic concepts of the discipline and does not know how to use the knowledge gained in solving typical practical problems.

5. Methodological materials defining the procedures for the assessment of knowledge, skills, abilities and/or experience

Differentiated classification is carried out taking into account current academic performance and the results of the completion of term paper. If necessary, in the process of interviewing a student, a selective survey is conducted on the knowledge of control questions, and typical tasks are offered.