

**Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»**

УТВЕРЖДЕНО

**Директор высшей школы
программной инженерии
А.В. Малеев**

	Рабочая программа дисциплины (модуля)
по дисциплине:	Разработка распределенных систем
по направлению:	Программная инженерия
профиль подготовки:	Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии
курс:	3
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 5 (осенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 48 час.

Всего часов: 108, всего зач. ед.: 3

Программу составил: А.В. Созыкин, канд. техн. наук, доцент

Программа обсуждена на заседании высшей школы программной инженерии 13.05.2024

Аннотация

В этом курсе студенты познакомятся с:

- такие подходы к разработке параллельных программ, как MPI, OpenMP;
- Вычисления на графических процессорах с использованием CUDA.
- устройство распределенных очередей и хранилищ (ZooKeeper, Kafka, Cassandra);
- основные фреймворки для обработки больших данных и их внутренняя структура.

Курс ориентирован на практику. Программирование предполагается как на семинарах, так и в качестве домашнего задания.

1. Цели и задачи

Цель дисциплины

Познакомить студентов с основами многопроцессорных вычислительных систем и дать практический опыт работы с такими системами. Курс состоит из двух модулей, посвященных параллельным и распределенным системам соответственно. Первый модуль исследует системы в почти «идеальных» условиях, когда вычислительные узлы и соединения между ними надежны и быстры. Второй модуль исследует способы построения надежных систем из ненадежных компонентов.

Задачи дисциплины

В ходе курса студенты получают практические навыки работы как с параллельными, так и с распределенными системами.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.3 Рассматривает различные варианты решения задачи, оценивает их преимущества и недостатки
	УК-1.4 Грамотно, логично, аргументированно формирует собственные суждения и оценки
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1 Формулирует совокупность взаимосвязанных задач в рамках поставленной цели работы, обеспечивающих ее достижение. Определяет ожидаемые результаты решения поставленных задач
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.3 Знает методы тестирования программного кода на ошибки и способен проводить тестирование на различных уровнях (модульное, интеграционное, системное)
	ОПК-6.2 Умеет применять языки программирования для решения прикладных задач
	ОПК-6.4 Имеет навыки программирования и тестирования программных продуктов
ОПК-8 Способен осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий	ОПК-8.3 Умеет оптимизировать и проводить рефакторинг существующего кода для улучшения производительности и поддержки
	ОПК-8.1 Понимает принципы, по которым работают базы данных, и умеет создавать структуру данных, оптимизированную для эффективного хранения и обработки информации
	ОПК-8.2 Умеет применять технологии машинного обучения в различных прикладных областях

ПК-3 Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение	ПК-3.3 Умеет излагать основные принципы построения и виды архитектуры программного обеспечения, методы и средства проектирования программного обеспечения, методологии разработки программного обеспечения и технологии программирования
	ПК-3.1 Различает синтаксис языков программирования, особенности программирования на этих языках, стандартные библиотеки языков программирования
	ПК-3.2 Умеет выбирать языки программирования для написания программного кода с учетом технического задания
ПК-5 Способен проектировать, разрабатывать, внедрять, сопровождать и снимать с эксплуатации информационные системы	ПК-5.4 Имеет практический опыт разработки, внедрения и адаптации прикладного программного обеспечения информационных систем в различных программных средах
	ПК-5.3 Умеет разрабатывать прототип информационных систем в соответствии с требованиями и проводить его тестирование для проверки корректности архитектурных решений
	ПК-5.2 Умеет определять оптимальные методы и инструменты разработки, внедрения и адаптации прикладного программного обеспечения информационных систем
	ПК-5.1 Умеет описывать архитектуру, устройство и функционирование информационных систем

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- типы и классификация многопроцессорных вычислительных систем;
- принципы построения распределенных хранилищ данных;
- принципы построения распределенных экосистем (Hadoop / Spark);
- разница между (одно-) серверными базами данных и распределенными базами данных;
- модель асинхронных вычислений и взаимосвязь со степенью изолированности транзакции;
- теорема Фишера-Линча-Патерсона (FLP-теорема);
- знать основные принципы Paxos / Raft;
- знать алгоритмы синхронизации времени (NTP, алгоритм Кристиана)
- стандартные задачи распределенных вычислений (многоадресные рассылки, детекторы отказов, членство, консенсус, RSM)

уметь:

- использовать библиотеку для параллельных вычислений OpenMP;
- использовать библиотеку для параллельных вычислений MPI;
- использовать распределенную файловую систему HDFS;
- использовать среду распределенных вычислений Hadoop;
- использовать распределенное хранилище данных Hive;
- уметь использовать примитивы распределенных вычислений Lamport Timestamps, Vector Clocks
- решить проблему консенсуса в синхронной системе;
- использовать алгоритмы Paxos / Raft.

владеть:

- навыки работы с многопроцессорными вычислительными системами (в частности, с параллельными и распределенными вычислительными системами)
- горизонты в выборе архитектурного решения поставленной задачи.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Теория распределенных вычислений (уровни изоляции транзакций, CRDT, CAP, FLP, Paxos, Raft)	15	15		24
2	Вычисления на графическом процессоре. CUDA-технология	15	15		24
Итого часов		30	30		48
Подготовка к экзамену		0 час.			
Общая трудоёмкость		108 час., 3 зач.ед.			

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 5 (Осенний)

1. Теория распределенных вычислений (уровни изоляции транзакций, CRDT, CAP, FLP, Paxos, Raft)

- Классификация многопроцессорных вычислительных систем, модели отказов;
- Разница между параллельными вычислениями и распределенными вычислениями;
- Базы данных: ACID, уровни изоляции. Распределенные базы данных: CAP, CRDT. Системы AP и CP;
- Алгоритмы синхронизации времени (NTP, Cristian's Algorithm);
- стандартные задачи распределенных вычислений (многоадресные рассылки, детекторы отказов, членство, консенсус, RSM);
- Теорема Фишера-Линча-Патерсона (FLP-теорема);
- Основные принципы Paxos / Raft;

2. Вычисления на GPU. Технология CUDA

Архитектура GPU.

Устройство (графический процессор) и Хост. Обмен данными между ними.

Оптимизация вычислений на GPU.

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Учебный класс оборудован мультимедийным проектором и экраном.

6.Перечень рекомендуемой литературы

Основная литература

1. Теория и практика параллельных вычислений, учебное пособие / В. П. Гергель. — Москва, ИНТУИТ, 2016.— URL: <https://ibooks.ru/bookshelf/363224/reading> (дата обращения: 25.11.2020). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)

Дополнительная литература

Рекомендуемая литература для самостоятельного изучения:

- Батищев, Д. С., & Михелев, В. М. (2016). Инфраструктура высокопроизводительной компьютерной системы для реализации облачных сервисов хранения и анализа данных персональной медицины

7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Не предусмотрено

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

Компьютерный класс. На каждом компьютере должен быть доступ в Интернет и программное обеспечение для подключения к удаленным серверам.

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Успешное освоение курса требует от студентов большой самостоятельной работы. Программа курса содержит минимально необходимое время для работы студента над темой. Самостоятельная работа включает:

- изучение учебного материала (на основе конспектов лекций, учебной и научной литературы), подготовка ответов на вопросы для самостоятельного изучения, доказательство отдельных заявлений, свойств;
- подготовка к практическим занятиям, 6 индивидуальных домашних заданий.

Промежуточный контроль знаний осуществляется в виде письменных опросов (мини-тестов) по теории.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению: Программная инженерия
профиль подготовки: Разработка программно-информационных систем
высшая школа программной инженерии
высшая школа программной инженерии
курс: 3
квалификация: бакалавр

Семестр, формы промежуточной аттестации: 5 (осенний) - Дифференцированный зачет

Разработчик: А.В. Созыкин, канд. техн. наук, доцент

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.3 Рассматривает различные варианты решения задачи, оценивает их преимущества и недостатки
	УК-1.4 Грамотно, логично, аргументированно формирует собственные суждения и оценки
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1 Формулирует совокупность взаимосвязанных задач в рамках поставленной цели работы, обеспечивающих ее достижение. Определяет ожидаемые результаты решения поставленных задач
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.3 Знает методы тестирования программного кода на ошибки и способен проводить тестирование на различных уровнях (модульное, интеграционное, системное)
	ОПК-6.2 Умеет применять языки программирования для решения прикладных задач
	ОПК-6.4 Имеет навыки программирования и тестирования программных продуктов
ОПК-8 Способен осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий	ОПК-8.3 Умеет оптимизировать и проводить рефакторинг существующего кода для улучшения производительности и поддержки
	ОПК-8.1 Понимает принципы, по которым работают базы данных, и умеет создавать структуру данных, оптимизированную для эффективного хранения и обработки информации
	ОПК-8.2 Умеет применять технологии машинного обучения в различных прикладных областях
ПК-3 Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение	ПК-3.3 Умеет излагать основные принципы построения и виды архитектуры программного обеспечения, методы и средства проектирования программного обеспечения, методологии разработки программного обеспечения и технологии программирования
	ПК-3.1 Различает синтаксис языков программирования, особенности программирования на этих языках, стандартные библиотеки языков программирования
	ПК-3.2 Умеет выбирать языки программирования для написания программного кода с учетом технического задания
ПК-5 Способен проектировать, разрабатывать, внедрять, сопровождать и снимать с эксплуатации информационные системы	ПК-5.4 Имеет практический опыт разработки, внедрения и адаптации прикладного программного обеспечения информационных систем в различных программных средах
	ПК-5.3 Умеет разрабатывать прототип информационных систем в соответствии с требованиями и проводить его тестирование для проверки корректности архитектурных решений
	ПК-5.2 Умеет определять оптимальные методы и инструменты разработки, внедрения и адаптации прикладного программного обеспечения информационных систем

2. Показатели оценивания компетенций

В результате изучения дисциплины «Разработка распределенных систем» обучающийся должен:

знать:

- типы и классификация многопроцессорных вычислительных систем;
- принципы построения распределенных хранилищ данных;
- принципы построения распределенных экосистем (Hadoop / Spark);
- разница между (одно-) серверными базами данных и распределенными базами данных;
- модель асинхронных вычислений и взаимосвязь со степенью изолированности транзакции;
- теорема Фишера-Линча-Патерсона (FLP-теорема);
- знать основные принципы Paxos / Raft;
- знать алгоритмы синхронизации времени (NTP, алгоритм Кристиана)
- стандартные задачи распределенных вычислений (многоадресные рассылки, детекторы отказов, членство, консенсус, RSM)

уметь:

- использовать библиотеку для параллельных вычислений OpenMP;
- использовать библиотеку для параллельных вычислений MPI;
- использовать распределенную файловую систему HDFS;
- использовать среду распределенных вычислений Hadoop;
- использовать распределенное хранилище данных Hive;
- уметь использовать примитивы распределенных вычислений Lamport Timestamps, Vector Clocks
- решить проблему консенсуса в синхронной системе;
- использовать алгоритмы Paxos / Raft.

владеть:

- навыки работы с многопроцессорными вычислительными системами (в частности, с параллельными и распределенными вычислительными системами)
- горизонты в выборе архитектурного решения поставленной задачи.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

Не предусмотрено

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

Инструменты распределенных вычислений.

Описание распределенной файловой системы HDFS, отказоустойчивости и масштабируемости.

Плюсы и минусы HDFS.

Назначение вторичного Namenode в HDFS.

Процесс чтения и записи данных в HDFS.

Кластерная структура Hadoop. Роли сервера. Как обеспечивается отказоустойчивость.

Процесс выполнения задачи на кластере. Распределенный кеш.

Формальная модель вычисления MapReduce.

Объемы Writable и WritableComparable.

Типы компараторов в MapReduce.

Какие этапы приложения MapReduce должен реализовать разработчик.

MapReduce: комбайнер, компаратор, разделитель.

Пример билета.

Инструменты распределенных вычислений.

MapReduce: комбайнер, компаратор, разделитель.

Критерии оценивания

превосходно

10 всестороннее, систематизированное, глубокое знание учебного плана дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принимаемых решений;

9 систематическое, глубокое знание учебного плана дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принимаемых решений;

8 глубокое знание учебного плана дисциплины и умение применять их на практике при решении конкретных задач, правильное обоснование принимаемых решений;

хороший

7 твердо знает материал, правильно и содержательно излагает его, умеет применять полученные знания на практике, но допускает неточности в ответе или решении задач;

6 знает материал, правильно его преподносит, умеет применять полученные знания на практике, но допускает неточности в ответе или решении задач;

5 знает основной материал, правильно излагает его, умеет применять полученные знания на практике, но допускает неточности в ответе или решении задач;

удовлетворительно

4 фрагментарный, фрагментарный характер знаний, недостаточно правильная формулировка основных понятий, нарушение логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения, и может применять полученные знания в стандартной ситуации;

3 характер знаний достаточен для дальнейшего обучения и может применить полученные знания на модели в стандартной ситуации;

неудовлетворительный

2 не знает большей части основного содержания учебного плана дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не знает, как правильно использовать полученные знания при решении типовых практических задач.

1 не знает формулировки основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Дифференцированная классификация проводится с учетом текущей успеваемости и результатов выполнения курсовой работы. При необходимости в процессе интервьюирования студента проводится выборочный опрос на знание контрольных вопросов, и предлагаются типовые задания.