

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

УТВЕРЖДЕНО

**Директор физтех-школы
прикладной математики и
информатики**

А.М. Райгородский

	Рабочая программа дисциплины (модуля)
по дисциплине:	Хранение и обработка больших объёмов данных
по направлению:	Прикладная математика и информатика
профиль подготовки:	Прикладная математика и информатика Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
курс:	1
квалификация:	магистр

Семестр, формы промежуточной аттестации: 2 (весенний) - Экзамен

Аудиторных часов: 30 всего, в том числе:

лекции: 0 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 30 час.

Подготовка к экзамену: 30 час.

Всего часов: 90, всего зач. ед.: 2

Программу составили:

О.Н. Ивченко, заведующий кафедрой

П.В. Мезенцев, ассистент

А.А. Горохов, канд. физ.-мат. наук, доцент

П.И. Ахтямов, ассистент

А.Н. Штохов, ассистент

А.И. Выборнов, ассистент

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 02.04.2024

Аннотация

Курс "Хранение и обработка больших объемов данных" представляет собой введение в современные технологии, методы и инструменты, используемые для работы с большими объемами данных. В ходе курса студенты изучают основные концепции и подходы к обработке данных, анализу больших объемов информации, а также методы хранения и управления данными в условиях больших нагрузок. Студенты также знакомятся с распределенными системами хранения данных, технологиями облачных вычислений, методами обработки потоков данных и инструментами для работы с большими объемами информации. Курс охватывает такие темы, как базы данных NoSQL, параллельные вычисления, анализ данных, машинное обучение и другие современные подходы к работе с данными.

1. Цели и задачи

Цель дисциплины

- овладение алгоритмами, парадигмами и инструментами для пакетной и потоковой обработки больших объёмов данных.

Задачи дисциплины

- приобретение студентами навыков проектирования архитектур, применения специализированных инструментов и разработки программных систем для работы с большими объемами данных.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-2 Способен управлять проектом на всех этапах его жизненного цикла	УК-2.1 Формулирует в рамках обозначенной проблемы цель, задачи, актуальность, значимость (научную, практическую, методическую и иную в зависимости от типа проекта), ожидаемые результаты и возможные сферы их применения
	УК-2.2 Способен прогнозировать результат деятельности и планировать последовательность шагов для достижения данного результата. Формирует план-график реализации проекта в целом и план контроля его выполнения
	УК-2.3 Способен организовать и координировать работу участников проекта, обеспечивать работу команды необходимыми ресурсами
	УК-2.4 Представляет публично результаты проекта (или отдельных его этапов) в форме отчетов, статей, выступлений на научно-практических конференциях, семинарах и т.п.
ОПК-6 Способен выбирать и (или) разрабатывать подходы к решению типовых и новых задач в области информатики и вычислительной техники, учитывая особенности и ограничения различных методов решения	ОПК-6.1 Способен анализировать задачу, планировать пути решения, предлагать и комбинировать способы решения
	ОПК-6.2 Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем
	ОПК-6.3 Способен использовать исследовательские методы при решении новых задач, применяя знания из различных областей науки (техники)
	ОПК-6.4 Владеет аналитическими и вычислительными методами решения, понимает и учитывает на практике границы применимости получаемых решений

ОПК-6.5 Способен самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- типы хранилищ больших объёмов данных;
- подходы к потоковой и пакетной обработке данных;
- принципы трансляции высокоуровневых языков программирования (SQL-подобных и функциональных) в последовательность задач на Hadoop кластере.

уметь:

- пользоваться распределенной файловой системой;
- запускать задачи на Hadoop кластере;
- писать задачи для запуска на Hadoop кластере с помощью нативного Java-интерфейса;
- писать задачи для запуска на Hadoop кластере с помощью любого другого языка программирования (с помощью инструментария Hadoop streaming);
- пользоваться высокоуровневыми языками программирования для BigData для обработки большого объема данных на вычислительном кластере;
- решать задачи статистики, задачи поиска и индексации, задачи машинного обучения на Hadoop кластере.

владеть:

- навыками работы с большими объемами данных и кругозором 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	Распределённые файловые системы (GFS, HDFS)		2		5
2	Парадигма MapReduce		6		5
3	Управление ресурсами Hadoop-кластера. YARN		2		4
4	SQL over BigData: Apache Drill, Cloudera Impala, Presto, Hive.		4		4
5	Технологии обработки данных в распределенной оперативной памяти. Apache Spark		6		4
6	Обработка данных в реальном времени. Kafka, Spark Streaming		4		4
7	BigData NoSQL, Key-value базы данных		6		4
Итого часов			30		30
Подготовка к экзамену		30 час.			

Общая трудоёмкость	90 час., 2 зач.ед.
--------------------	--------------------

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

Семестр: 2 (Весенний)

1. Распределённые файловые системы (GFS, HDFS)

Распределённые файловые системы (GFS, HDFS). Её составляющие. Их достоинства, недостатки и сфера применения. Чтение и запись в HDFS. HDFS APIs: Web, shell, Java.

2. Парадигма MapReduce

Парадигма MapReduce. Основная идея, формальное описание. Обзор реализаций. Виды и классификация многопроцессорных вычислительных систем. Hadoop. Схема его работы, роли серверов в Hadoop-кластере. API для работы с Hadoop (Native Java API vs. Streaming), примеры.

MapReduce, продолжение. Типы Join'ов и их реализации в парадигме MR. Паттерны проектирования MR (pairs, stripes, составные ключи).

3. Управление ресурсами Hadoop-кластера. YARN

Hadoop MRv1 vs. YARN. Нововведения в последних версиях Hadoop. Планировщик задач в YARN. Apache Slider.

4. SQL over BigData: Apache Drill, Cloudera Impala, Presto, Hive.

SQL over BigData: Apache Drill, Cloudera Impala, Presto, Hive. Повторение SQL. HiveQL vs. SQL. Виды таблиц в Hive, типы данных, трансляция Hive-запросов в MapReduce-задачи.

Аналитические функции в Hive. Расширения Hive: Streaming, User defined functions. Оптимизация запросов в Hive.

5. Технологии обработки данных в распределенной оперативной памяти. Apache Spark

Spark RDD vs Spark Dataframes

Spark SQL

Spark GraphFrames

6. Обработка данных в реальном времени. Kafka, Spark Streaming

Обработка данных в реальном времени. Spark Streaming.

Распределённая очередь Apache Kafka. Kafka streams.

7. BigData NoSQL, Key-value базы данных

HBase. NoSQL подходы к реализации распределенных баз данных, key-value хранилища. Основные компоненты BigTable-подобных систем и их назначение, отличие от реляционных БД. Чтение, запись и хранение данных в HBase. Minor- и major-компактификация. Надёжность и отказоустойчивость в HBase.

Cassandra. Основные особенности. Чтение и запись данных. Отказоустойчивость. Примеры применения HBase и Cassandra.

Отличие архитектуры HBase от Cassandra.

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

Для практических занятий:

Компьютерный класс. Каждый компьютер должен иметь выход в интернет и ПО для подключения к удалённым серверам.

6. Перечень рекомендуемой литературы

Основная литература

1. Персональный компьютер для всех [Текст] : в 4-х кн. : Кн. 1. Хранение и обработка информации / А. Я. Савельев, Б. А. Сазонов, С. Э. Лукьянов / под ред. А. Я. Савельева .— М. : Высшая школа, 1991 .— 191 с.

Григорьев А. А., Исаев Е. А., Тарасов П. А. Передача, хранение и обработка больших объемов научных данных. - ИНФРА-М, 2021

Дополнительная литература

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

<https://www.coursera.org/specializations/big-data-engineering> - специализация из 5 курсов, посвящённая тематике обработки больших данных.

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

Для практических занятий:

Компьютерный класс. Каждый компьютер должен иметь выход в интернет и ПО для подключения к удалённым серверам.

Удалённый кластер с такими характеристиками:

Характеристики одной машины

Кол-во машин	Объём оперативной памяти	Кол-во ядер CPU	Объём дисковой памяти
Операционная система			
1	8	2	200
Linux Ubuntu 16.04			
9	32	8	600
Linux Ubuntu 16.04			

На кластере должен быть развернута последняя версия Cloudera Manager, в который нужно встроить такие сервисы: HDFS, YARN, Hive, Spark2 on YARN, HBase, Zookeeper, Kafka.

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

Успешное освоение курса требует большой самостоятельной работы студента. В программе курса приведено минимально необходимое время для работы студента над темой. Самостоятельная работа включает в себя:

- проработку учебного материала (по учебной и научной литературе);
- подготовку ответов на вопросы, предназначенных для самостоятельного изучения;
- доказательство отдельных утверждений, свойств;
- подготовку к практическим занятиям,

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

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Прикладная математика и информатика
профиль подготовки:	Прикладная математика и информатика Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
курс:	<u>1</u>
квалификация:	магистр

Семестр, формы промежуточной аттестации: 2 (весенний) - Экзамен

Разработчики:

О.Н. Ивченко, заведующий кафедрой
П.В. Мезенцев, ассистент
А.А. Горохов, канд. физ.-мат. наук, доцент
П.И. Ахтямов, ассистент
А.Н. Штохов, ассистент
А.И. Выборнов, ассистент

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-2 Способен управлять проектом на всех этапах его жизненного цикла	УК-2.1 Формулирует в рамках обозначенной проблемы цель, задачи, актуальность, значимость (научную, практическую, методическую и иную в зависимости от типа проекта), ожидаемые результаты и возможные сферы их применения
	УК-2.2 Способен прогнозировать результат деятельности и планировать последовательность шагов для достижения данного результата. Формирует план-график реализации проекта в целом и план контроля его выполнения
	УК-2.3 Способен организовать и координировать работу участников проекта, обеспечивать работу команды необходимыми ресурсами
	УК-2.4 Представляет публично результаты проекта (или отдельных его этапов) в форме отчетов, статей, выступлений на научно-практических конференциях, семинарах и т.п.
ОПК-6 Способен выбирать и (или) разрабатывать подходы к решению типовых и новых задач в области информатики и вычислительной техники, учитывая особенности и ограничения различных методов решения	ОПК-6.1 Способен анализировать задачу, планировать пути решения, предлагать и комбинировать способы решения
	ОПК-6.2 Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем
	ОПК-6.3 Способен использовать исследовательские методы при решении новых задач, применяя знания из различных областей науки (техники)
	ОПК-6.4 Владеет аналитическими и вычислительными методами решения, понимает и учитывает на практике границы применимости получаемых решений
	ОПК-6.5 Способен самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте

2. Показатели оценивания компетенций

В результате изучения дисциплины «Хранение и обработка больших объёмов данных» обучающийся должен:

знать:

- типы хранилищ больших объёмов данных;
- подходы к потоковой и пакетной обработке данных;
- принципы трансляции высокоуровневых языков программирования (SQL-подобных и функциональных) в последовательность задач на Hadoop кластере.

уметь:

- пользоваться распределенной файловой системой;
- запускать задачи на Hadoop кластере;
- писать задачи для запуска на Hadoop кластере с помощью нативного Java-интерфейса;
- писать задачи для запуска на Hadoop кластере с помощью любого другого языка программирования (с помощью инструментария Hadoop streaming);
- пользоваться высокоуровневыми языками программирования для BigData для обработки большого объема данных на вычислительном кластере;
- решать задачи статистики, задачи поиска и индексации, задачи машинного обучения на Hadoop кластере.

владеть:

- навыками работы с большими объемами данных и кругозором в выборе архитектурного решения поставленной задачи.

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

Контрольные задания:

1. Какие семантики доставки сообщений вы знаете? Хотя бы для двух из них приведите пример реальных систем.
2. Что такое Compaction в HBase? Какие они бывают и чем отличаются?
3. Назовите основное отличие архитектуры HBase от архитектуры Cassandra. Какие плюсы и минусы имеет архитектура Cassandra по сравнению с HBase?
4. Что такое Big Data (большие данные) и какие основные характеристики этого типа информации?
5. Какие технологии используются для хранения больших объемов данных? Расскажите о реляционных и нереляционных базах данных.
6. Какие методы обработки больших данных существуют? Объясните различия между batch-обработкой и стримингом данных.
7. Что такое распределенные вычисления и как они применяются в обработке больших объемов информации?
8. Какие алгоритмы используются для анализа больших данных? Объясните, что такое машинное обучение и его роль в обработке Big Data.
9. Какие проблемы могут возникнуть при работе с большими данными? Какие методы существуют для обеспечения безопасности и конфиденциальности данных?
10. Какие преимущества и недостатки связаны с использованием облачных сервисов для хранения и обработки больших объемов информации?

Методические рекомендации:

1. Выбор подходящей технологии хранения данных: Определите, какие технологии хранения данных (базы данных, облачные хранилища и т. д.) наилучшим образом подходят для ваших конкретных потребностей.
2. Структурирование данных: Разработайте четкую структуру данных, определите форматы и типы данных, чтобы обеспечить удобство и эффективность доступа к информации.
3. Резервное копирование данных: Регулярно создавайте резервные копии данных, чтобы предотвратить потерю информации в случае сбоев системы или других проблем.
4. Обеспечение безопасности данных: Применяйте меры защиты данных, такие как шифрование, аутентификация и контроль доступа, чтобы предотвратить несанкционированный доступ к информации.
5. Масштабируемость: При проектировании системы хранения и обработки данных учитывайте возможность масштабирования, чтобы обеспечить эффективную работу с растущими объемами информации.
6. Оптимизация запросов и обработки данных: Используйте оптимизированные алгоритмы и инструменты для эффективной обработки запросов к данным и анализа больших объемов информации.
7. Мониторинг и управление данными: Внедрите систему мониторинга данных, которая позволит отслеживать состояние системы хранения, производительность и доступность данных.

8. Соблюдение правил и нормативов: Убедитесь, что ваша система хранения и обработки данных соответствует требованиям законодательства о защите данных (например, GDPR) и другим регулятивным стандартам.

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

1. HDFS, Hadoop

Устройство HDFS, основные идеи.

Схема чтения из HDFS.

Схема записи в HDFS.

Недостатки HDFS.

Хранение файлов: блоки и сплиты.

Консольный и программный интерфейсы HDFS.

2. BigData, MapReduce

Идея MapReduce, примеры применения.

Проблемы распределенных вычислений.

MPI vs. MapReduce

Реализации MapReduce

Hadoop, возможности для программирования.

Пример на Hadoop Java API, запуск задания.

Mapper и reducer.

Компоненты кластера Hadoop.

Веб-интерфейсы Namenode и Jobtracker.

3. Hadoop

Combiner.

Comparator.

Partitioner.

Типы данных и форматы файлов.

Все вместе: схема работы Hadoop.

Настройки задачи.

Процесс запуска задачи на кластере.

Управление задачей (старт, стоп)

Термины (job, task, attempt)

Счетчики.

Полные интерфейсы Mapper и Reducer.

In-mapper combiner.

Стратегии stripes и pairs.

Последовательности Hadoop задач.

Топологическая сортировка графа.

Reduce-side join.

Secondary sort.

Map-side join.

Distributed cache.

Bucket-side join.

Streaming для Hadoop, основные идеи.

Недостатки и достоинства streaming.

Пример запуска задач с помощью streaming.

4. YARN

YARN: основные идеи и термины.

Веб-интерфейс resource manager.

Distributed shell.

Запуск MR-задач на YARN.

MapReduce uber job.

Планировщики, управление ресурсами.

YARN High Availability.

Особенности Hadoop версий 3.x.

5. Hive

Примеры задач и применимость SQL для решения.

Возможности Hive.

Архитектура Hive, термины, примеры запросов.

Hive: типы данных, форматы хранения таблиц.

Создание таблиц.

Partitioning.

Bucketing.

Язык запросов: импорт и экспорт данных.

User defined functions.

Streaming в Hive.

6. Spark

Apache Spark: основные идеи.

Представление вычислений в виде графа.

Структура данных на worker'ах.

RDD для HDFS, интерфейс.

Пример задачи на Spark, экосистема проекта.

Схема выполнения задачи на Spark, термины.

Spark на YARN.

Разработка приложений на Spark, примеры.

SparkSQL, взаимодействие с Hive.

7. Realtime обработка

Гарантии обработки.

Функции верхнего уровня.

Lambda архитектура.

Spark streaming. Dstream.

At least once и exactly once в spark.

Apache Kafka.

8. HBase

Key-value хранилища и HDFS.

Архитектура HBase, термины.

Распределение данных по машинам.

Схема записи данных.

Удаление.

Компактификация.

Чтение.

Роль мастер-сервера.

Обеспечение отказоустойчивости.

Отличия от реляционных СУБД.

Операции put, get, scan.

Структура записи в HBase.

Применение для хранения web-страниц.

Применение для хранения графов.

9. Cassandra

Предпосылки создания.
Партиционирование ключей.

10.Реплицирование
Обеспечение отказоустойчивости.

Пример экзаменационного билета:

1. Какие из преобразований HIVE позволяют изменять количество строк в таблице? * UDF * UDAF * UDTF * PTF (оконные ф-ции).
2. В таблице HBase в качестве ключа таблицы используется доменное имя. Для каких целей удобно хранить домен в обратном порядке (market.yandex.ru —> ru.yandex.market)?
3. Подходы к обеспечению обновления кода Spark Streaming с сохранением семантики доставки. Семантика и плюсы/минусы для каждого подхода. Зачем нужны эти подходы к обновлению кода с сохранением семантики, если в Spark Streaming есть checkpoint?
4. Есть стандартный wordcount: mapper разбивает на слова, reducer суммирует. Какими способами в Hadoop можно его ускорить?

Критерии оценивания

"отлично"

10 - всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;

9 - систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений;

8 - глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, правильное обоснование принятых решений;

"хорошо"

7 - твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

6 - знает материал, грамотно излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

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

"удовлетворительно"

4 - фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

3 - характер знаний достаточен для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

"неудовлетворительно"

2 - не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет правильно использовать полученные знания при решении типовых практических задач.

1 - не знает формулировок 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

При проведении экзамена обучающемуся предоставляется 60 минут на подготовку. Опрос обучающегося по билету на экзамене не должен превышать двух астрономических часов.

Во время проведения экзамена обучающиеся могут пользоваться программой дисциплины и своими конспектами.

Экзаменационный билет состоит из 4 вопросов, каждый из которых оценивается в 0,5 балла.

2 вопроса являются теоретическими, другие 2 содержат задачи. Для решения задач не требуется написания кода.

Итоговая оценка по курсу складывается из оценки за выполненные в ходе семестра практические задания и оценки за ответы на теоретические вопросы на экзамене.