

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных технологий и прикладной математики

УТВЕРЖДАЮ:

Проректор по учебной работе,
качеству образования – первый
проректор

_____ Хагуров Т.А.

подпись
« 29 » августа 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1. О.38 Высоконагруженные приложения

Направление подготовки 02.03.01 Математика и компьютерные науки

Профиль Искусственный интеллект и аналитика данных

Форма обучения очная

Квалификация бакалавр

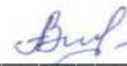
Краснодар 2025

Рабочая программа дисциплины «Высоконагруженные приложения» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 02.03.01 Математика и компьютерные науки

Программу составил(и):

Харченко Анна Владимировна, доцент, канд. пед наук

И.О. Фамилия, должность, ученая степень, ученое звание



подпись

Рабочая программа дисциплины «Высоконагруженные приложения» утверждена на заседании кафедры информационных технологий протокол № 1 от «26» августа 2025г.

Заведующий кафедрой Подколзин В.В.



Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики протокол № 01 «28» августа 2025 г.

Председатель УМК факультета Коваленко А.В.



подпись

Рецензенты:

Мостовой Евгений Викторович, генеральный директор ООО «Портал-Юг»,
e-mail: mostovoy@portal-yug.ru

Луценко Евгений Вениаминович, доктор экономических наук, кандидат технических наук, профессор кафедры компьютерных технологий и систем Федерального государственного бюджетное образовательное учреждение высшего образования «Кубанский государственный аграрный университет имени И.Т. Трубилина», e-mail: prof.lutsenko@gmail.com

1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

Основная цель дисциплины "Высоконагруженные приложения" заключается в формировании у студентов комплексных знаний и практических навыков проектирования, разработки и эксплуатации масштабируемых, отказоустойчивых и высокопроизводительных систем обработки данных и сервисов искусственного интеллекта, способных эффективно функционировать в условиях экстремальных нагрузок и больших объемов информации.

1.2 Задачи дисциплины:

- изучение современных архитектурных подходов и шаблонов проектирования для построения масштабируемых распределенных систем в области ИИ и аналитики данных;
- освоение эффективных стратегий хранения, обработки и анализа больших объемов структурированных и неструктурированных данных;
- формирование практических навыков обеспечения отказоустойчивости и надежности ML-моделей и аналитических пайплайнов;
- развитие компетенций по оптимизации производительности высоконагруженных ИИ-сервисов и систем обработки данных;
- ознакомление с современными практиками управления, мониторинга и эксплуатации высоконагруженных приложений в production-среде;
- формирование навыков проектирования, разработки и тестирования распределенных систем;
- изучение методов обеспечения безопасности и конфиденциальности в высоконагруженных системах обработки данных;
- развитие системного мышления и способности принимать обоснованные архитектурные решения при работе с big data и ИИ-сервисами.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Высоконагруженные приложения» относится к «Обязательная часть» Блока 1 «Дисциплины (модули)» учебного плана.

1.4 Профессиональные роли в структуре образовательной программы

Роль 1: Data Analyst (Аналитик данных)

Извлечение знаний из данных, построение аналитических моделей, использующих МО и ИИ.

Задачи:

- 1. Статистический анализ, визуализация данных, предварительная обработка.*
- 2. Создание прогнозных моделей*
- 3. Построение аналитических моделей для поддержки бизнес-решений.*

Роль 2: MLOps (Специалист по эксплуатации ИИ)

Автоматизация и операционное управление жизненным циклом МО-моделей

Задачи:

- 1. DevOps для ML.*
- 2. Автоматизация, мониторинг ML-систем.*
- 3. Операционное управление жизненным циклом ML-моделей.*

Роль 3: AI PM (Менеджер проектов ИИ)

Управление процессами создания ИИ-решений, включая координацию команды разработки

Задачи:

- 1. Управление ИИ-проектами от идеи до внедрения*
- 2. Анализ бизнес-требований и постановка задач*
- 3. Оценка эффективности и ROI ИИ-решений*

1.5 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

ОПК-4 **Способен находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем**

ОПК-4.1 **Разрабатывает и оптимизирует алгоритмы с учетом вычислительной сложности и аппаратных ограничений**

Знать *Математические основы алгоритмов для высоконагруженных систем: теория очередей, теория графов, методы оптимизации
Методы оценки и анализа вычислительной сложности алгоритмов в условиях больших данных
Принципы работы с распределенными вычислениями (MapReduce, Spark) и параллельными алгоритмами*

Уметь *Разрабатывать эффективные алгоритмы для обработки потоковых данных в реальном времени
Оптимизировать алгоритмы с учетом архитектурных особенностей современных процессоров и систем хранения
Анализировать и выбирать математические методы для решения задач масштабирования и балансировки нагрузки*

Владеть *Навыками реализации математических алгоритмов для высокопроизводительных вычислений
Методами профилирования и оптимизации алгоритмической сложности в high-load системах
Техниками распараллеливания вычислений для многопроцессорных и распределенных систем*

ОПК-4.2 **Тестирует и внедряет алгоритмы в реальные задачи, оценивая их точность и производительность**

Знать *Методы нагрузочного тестирования алгоритмов в условиях высоких параллельных нагрузок
Метрики оценки производительности алгоритмов в распределенных системах (throughput, latency, error rate)
Принципы A/B-тестирования и канареечных развертываний алгоритмических компонентов
Методы мониторинга дрейфа производительности и точности алгоритмов в production-среде*

Уметь *Проводить стресс-тестирование алгоритмов на синтетических и реальных данных высокой нагрузки
Сравнивать производительность различных алгоритмических реализаций в условиях ограничений hardware
Настраивать системы мониторинга для отслеживания производительности алгоритмов в реальном времени
Оценивать влияние алгоритмических изменений на общую производительность high-load системы*

Владеть	<p>Навыками использования инструментов нагрузочного тестирования (JMeter, k6, Yandex.Tank)</p> <p>Методами профилирования и оптимизации алгоритмов под конкретные hardware-конфигурации</p> <p>Практиками постепенного внедрения алгоритмических изменений в высоконагруженные системы</p> <p>Инструментами анализа и визуализации метрик производительности алгоритмов (Grafana, Prometheus)</p>
ОПК-5	Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности
ОПК-5.1	Применяет современные языки программирования и технологии для решения математических и вычислительных задач
Знать	<p>Современные языки программирования для high-load систем: Go, Rust, C++, Python с асинхронными фреймворками</p> <p>Технологии для распределенных вычислений: gRPC, Protocol Buffers, Apache Kafka, Redis Cluster</p> <p>Принципы построения отказоустойчивых и масштабируемых архитектур</p>
Уметь	<p>Разрабатывать высокопроизводительные микросервисы на современных языках программирования</p> <p>Применять технологии кэширования и балансировки нагрузки для оптимизации производительности</p> <p>Использовать инструменты мониторинга и диагностики производительности распределенных систем</p>
Владеть	<p>Навыками работы с фреймворками для построения high-load API (FastAPI, Gin, Actix)</p> <p>Методами оптимизации сетевого взаимодействия в распределенных системах</p> <p>Практиками обеспечения низкой задержки и высокой пропускной способности систем</p>
ОПК-6	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения
ОПК-6.2	Применяет методы структурного и объектно-ориентированного программирования, создает модульные и масштабируемые программы
Знать	<p>Паттерны проектирования для высоконагруженных систем: Circuit Breaker, Bulkhead, CQRS, Event Sourcing</p> <p>Принципы модульной архитектуры и микросервисного подхода к разработке</p> <p>Методы обеспечения горизонтального масштабирования и отказоустойчивости</p>
Уметь	<p>Проектировать модульную архитектуру high-load приложений с четким разделением ответственности</p> <p>Реализовывать масштабируемые сервисы с использованием принципов объектно-ориентированного программирования</p> <p>Применять практики рефакторинга для улучшения производительности и поддерживаемости кода</p>

- Владеть** *Навыками создания масштабируемых и отказоустойчивых программных компонентов*
Методами проектирования API для high-load систем с учетом требований производительности
Техниками обеспечения согласованности данных в распределенных системах
- PL-2** **Способен применять JVM-совместимые языки программирования для решения задач в области ИИ**
- PL-2.1** *Разрабатывает и отлаживает прикладные решения разного уровня сложности и для широкого круга конечных пользователей с использованием JVM-совместимых языков программирования, тестирует, испытывает и оценивает качество таких решений*
- Знать** *Принципы объектно-ориентированного и функционального программирования на JVM-совместимых языках (Kotlin, Scala).*
Архитектуру и основные компоненты JVM (JVM Memory Model, Garbage Collection).
Фреймворки для построения высоконагруженных приложений (Spring Boot, Micronaut, Vert.x).
Библиотеки для машинного обучения и обработки данных на JVM (Tribuo, Apache Spark MLlib, DL4J).
- Уметь** *Разрабатывать высокопроизводительные backend-сервисы и микросервисы на JVM-совместимых языках.*
Профилировать и оптимизировать JVM-приложения для работы с большими объемами данных и высокой параллельной нагрузкой.
Интегрировать ML-модели, реализованные на Python, в JVM-приложения (например, с использованием JRूपe).
Писать модульные и интеграционные тесты для обеспечения качества и надежности кода.
- Владеть** *Навыками разработки на одном из JVM-совместимых языков (Kotlin/Scala).*
Инструментами для профилирования и мониторинга JVM (JVisualVM, JMH, GC logs analysis).
Навыками работы с системами сборки и управления зависимостями (Gradle, Maven).
Практиками отладки и решения проблем производительности в JVM-среде.
- AI S-1** **Способен управлять рисками в разработке систем ИИ, выстраивать управление безопасностью ИИ в компании с учетом этики ИИ**
- AI S-1.1** *Выявляет и моделирует угрозы на всём жизненном цикле ИИ-систем, оценивает и приоритизирует риски*
- Знать** *Уязвимости высоконагруженных систем (DDoS, инъекции)*
Методы обеспечения кибербезопасности в ИИ
Принципы работы систем мониторинга аномалий
- Уметь** *Проводить стресс-тесты и анализ уязвимостей*
Разрабатывать стратегии защиты данных
Настраивать системы обнаружения вторжений (IDS/IPS)
- Владеть** *Инструментами (OWASP ZAP, Burp Suite)*
Навыками аудита безопасности
ML-методами для детектирования аномалий

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 2 зач. ед. (72 часов), их распределение по видам работ представлено в таблице

Вид учебной работы	Всего часов	Семестры (часы)					
		8					
Контактная работа, в том числе:	30,2	30,2					
Аудиторные занятия (всего):	28	28					
Занятия лекционного типа							
Лабораторные занятия	28	28					
Занятия семинарского типа (семинары, практические занятия)							
Иная контактная работа:	0,2	0,2					
Контроль самостоятельной работы (КСР)							
Промежуточная аттестация (ИКР)	0,2	0,2					
Самостоятельная работа, в том числе:	43,8	43,8					
<i>Выполнение индивидуальных заданий (подготовка сообщений, презентаций)</i>	30	30					
Подготовка к текущему контролю	13,8	13,8					
Контроль:							
Подготовка к экзамену							
Общая трудоемкость	час.	72	22				
	в том числе контактная работа	30,2	30,2				
	зач. ед	2	2				

2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы (темы) дисциплины, изучаемые в 7 семестре

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа СРС
			Л	ПЗ	ЛР	
1	2	3	4	5	6	7
1	Архитектура высоконагруженных систем	10			4	6
2	Горизонтальное и вертикальное масштабирование	10			4	6
3	Эффективные хранилища данных	10			4	6
4	Обработка больших объемов данных:	10			4	6
5	Отказоустойчивость и надежность	10			4	6
6	Производительность алгоритмов и аппаратная оптимизация	10			4	6
7	Мониторинг и управление высоконагруженными ИИ-сервисами	11,8			4	7,8
ИТОГО по разделам дисциплины		71,8			28	43,8
Контроль самостоятельной работы (КСР)		2				

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	СРС
1	2	3	4	5	6	7
	Промежуточная аттестация (ИКР)	0,2				
	Подготовка к текущему контролю					
	Общая трудоемкость по дисциплине	72				

Примечание: Л – лекции, ПЗ – практические занятия/семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

2.3 Содержание разделов (тем) дисциплины

2.3.1 Занятия лекционного типа

Не предусмотрены

Примечание: ЛР – отчет/защита лабораторной работы, КП – выполнение курсового проекта, КР – курсовой работы, РГЗ – расчетно-графического задания, Р – написание реферата, Э – эссе, К – коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.2 Занятия семинарского типа

Не предусмотрены

Примечание: ЛР – отчет/защита лабораторной работы, КП – выполнение курсового проекта, КР – курсовой работы, РГЗ – расчетно-графического задания, Р – написание реферата, Э – эссе, К – коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.3 Лабораторные занятия

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
1.	Архитектура высоконагруженных систем	Разработка микросервисной архитектуры для ML-сервиса – Создание простого сервиса рекомендаций на Flask/FastAPI, разбиение на микросервисы (модель, API, кэш). – Сравнение latency и throughput в monolithic vs microservices под нагрузкой (Locust).	РЗ
2.	Архитектура высоконагруженных систем	Анализ CAP-теоремы на примере распределённого чат-бота – Развёртывание чат-бота с обработкой запросов в разных DC (AWS/GCP), тестирование согласованности (Consistency) при разрывах сети.	РЗ
3.	Горизонтальное и вертикальное масштабирование	Автомасштабирование веб-сервера для обработки запросов к ИИ-модели – Настройка Auto Scaling Group (AWS/GCP) для сервиса классификации изображений, тесты при пиковой нагрузке.	РЗ

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
4.	Горизонтальное и вертикальное масштабирование	Сравнение вертикального и горизонтального scaling для базы данных – Тестирование PostgreSQL на одном мощном сервере vs шардированного MongoDB под нагрузкой (YCSB Benchmark).	P3
5.	Эффективные хранилища данных	Оптимизация запросов к векторной БД для поиска похожих изображений – Развёртывание Milvus/Pinecone, сравнение скорости поиска при разных индексах (IVF, HNSW).	P3
6.	Эффективные хранилища данных	Колоночное vs row-based хранилище для аналитики – Загрузка датасета в ClickHouse (колоночная) и PostgreSQL, сравнение скорости агрегаций (SQL-запросы с GROUP BY).	P3
7.	Обработка больших объемов данных	Распределённая обработка логов с помощью Apache Spark – Анализ логов веб-сервера (например, NGINX) для выявления аномалий через Spark SQL.	P3
8.	Обработка больших объемов данных	Стриминг данных в реальном времени для ML-модели – Настройка Kafka + Python-консьюмера, предсказание спама в потоке сообщений (модель на Scikit-learn).	P3
9.	Отказоустойчивость и надежность	Тестирование failover механизмов в Kubernetes – Развёртывание отказоустойчивого ML-сервиса (например, sentiment analysis), симуляция падения нод.	P3
10.	Отказоустойчивость и надежность	Репликация данных в распределённой СУБД – Настройка реплик MongoDB, тест на устойчивость к потере primary-ноды.	P3
11.	Производительность алгоритмов и аппаратная оптимизация	Ускорение inference модели на GPU vs CPU – Развёртывание PyTorch-модели (например, ResNet), замеры скорости на CPU и CUDA.	P3
12.	Производительность алгоритмов и аппаратная оптимизация	Оптимизация кода на C++ для обработки данных – Реализация алгоритма предобработки текста (токенизация) на Python и C++ (с PyBind11), сравнение производительности.	P3
13.	Мониторинг и управление высоконагруженными ИИ-сервисами	Детектирование дрейфа данных в ML-модели – Настройка Evidently AI для мониторинга сдвигов в распределении входных данных.	P3
14.	Мониторинг и управление высоконагруженными ИИ-сервисами	A/B-тестирование двух версий рекомендательной системы – Развёртывание двух моделей (например, collaborative filtering vs content-based), сбор метрик через Prometheus.	P3

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.4 Примерная тематика курсовых работ (проектов)

Не предусмотрены

.....

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Изучение теоретического материала	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой информационных технологий, протокол №1 от 30.08.2019
2	Решение задач	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой информационных технологий, протокол №1 от 30.08.2019

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,
- в форме аудиофайла,
- в печатной форме на языке Брайля.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа,
- в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии

В соответствии с требованиями ФГОС в программа дисциплины предусматривает использование в учебном процессе следующих образовательные технологии: чтение лекций с использованием мультимедийных технологий; метод малых групп, разбор практических задач и кейсов.

При обучении используются следующие образовательные технологии:

– Технология коммуникативного обучения – направлена на формирование коммуникативной компетентности студентов, которая является базовой, необходимой для адаптации к современным условиям межкультурной коммуникации.

– Технология разноуровневого (дифференцированного) обучения – предполагает осуществление познавательной деятельности студентов с учётом их индивидуальных способностей, возможностей и интересов, поощряя их реализовывать свой творческий

потенциал. Создание и использование диагностических тестов является неотъемлемой частью данной технологии.

- Технология модульного обучения – предусматривает деление содержания дисциплины на достаточно автономные разделы (модули), интегрированные в общий курс.

- Информационно-коммуникационные технологии (ИКТ) - расширяют рамки образовательного процесса, повышая его практическую направленность, способствуют интенсификации самостоятельной работы учащихся и повышению познавательной активности. В рамках ИКТ выделяются 2 вида технологий:

- Технология использования компьютерных программ – позволяет эффективно дополнить процесс обучения языку на всех уровнях.

- Интернет-технологии – предоставляют широкие возможности для поиска информации, разработки научных проектов, ведения научных исследований.

- Технология индивидуализации обучения – помогает реализовывать личностно-ориентированный подход, учитывая индивидуальные особенности и потребности учащихся.

- Проектная технология – ориентирована на моделирование социального взаимодействия учащихся с целью решения задачи, которая определяется в рамках профессиональной подготовки, выделяя ту или иную предметную область.

- Технология обучения в сотрудничестве – реализует идею взаимного обучения, осуществляя как индивидуальную, так и коллективную ответственность за решение учебных задач.

- Игровая технология – позволяет развивать навыки рассмотрения ряда возможных способов решения проблем, активизируя мышление студентов и раскрывая личностный потенциал каждого учащегося.

- Технология развития критического мышления – способствует формированию разносторонней личности, способной критически относиться к информации, умению отбирать информацию для решения поставленной задачи.

Комплексное использование в учебном процессе всех вышеназванных технологий стимулируют личностную, интеллектуальную активность, развивают познавательные процессы, способствуют формированию компетенций, которыми должен обладать будущий специалист.

Основные виды интерактивных образовательных технологий включают в себя:

- работа в малых группах (команде) - совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности;

- проектная технология - индивидуальная или коллективная деятельность по отбору, распределению и систематизации материала по определенной теме, в результате которой составляется проект;

- анализ конкретных ситуаций - анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений;

- развитие критического мышления – образовательная деятельность, направленная на развитие у студентов разумного, рефлексивного мышления, способного выдвинуть новые идеи и увидеть новые возможности.

Подход разбора конкретных задач и ситуаций широко используется как преподавателем, так и студентами во время лекций, лабораторных занятий и анализа результатов самостоятельной работы. Это обусловлено тем, что при исследовании и решении каждой конкретной задачи имеется, как правило, несколько методов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

Семестр	Вид занятия	Используемые интерактивные образовательные технологии	количество интерактивных часов
8	ЛР	Практические занятия в режимах взаимодействия «преподаватель – студент» и «студент – студент»	4
Итого			4

Примечание: Л – лекции, ПЗ – практические занятия/семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

Темы, задания и вопросы для самостоятельной работы призваны сформировать навыки поиска информации, умения самостоятельно расширять и углублять знания, полученные в ходе лекционных и практических занятий.

Подход разбора конкретных ситуаций широко используется как преподавателем, так и студентами при проведении анализа результатов самостоятельной работы.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

1. Оценочные и методические материалы

4.1 Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «Высоконагруженные приложения».

Оценочные средства включает контрольные материалы для проведения **текущего контроля** в форме тестовых заданий, разноуровневых заданий и **промежуточной аттестации** в форме вопросов и заданий к зачету.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

Структура оценочных средств для текущей и промежуточной аттестации

№ п/п	Контролируемые разделы (темы) дисциплины*	Код контролируемой компетенции (или ее части)	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1	Архитектура высоконагруженных систем	ОПК-4.1 ОПК-4.2 ОПК-5.1 ОПК-6.2 PL-2.1 AI S-1.1	<i>Задания для самостоятельного решения 1-2</i>	<i>Теоретический вопрос 1-4</i>
2	Горизонтальное и вертикальное масштабирование	ОПК-4.1 ОПК-4.2 ОПК-5.1 ОПК-6.2 PL-2.1 AI S-1.1	<i>Задания для самостоятельного решения 3-4</i>	<i>Теоретический вопрос 5-8</i>
3	Эффективные хранилища данных	ОПК-4.1 ОПК-4.2 ОПК-5.1 ОПК-6.2 PL-2.1 AI S-1.1	<i>Задания для самостоятельного решения 5-6</i>	<i>Теоретический вопрос 9-12</i>
4	Обработка больших объемов данных:	ОПК-4.1 ОПК-4.2 ОПК-5.1 ОПК-6.2 PL-2.1 AI S-1.1	<i>Задания для самостоятельного решения 7-8</i>	<i>Теоретический вопрос 13-16</i>
5	Отказоустойчивость и надежность	ОПК-4.1 ОПК-4.2 ОПК-5.1 ОПК-6.2 PL-2.1 AI S-1.1	<i>Задания для самостоятельного решения 9-10</i>	<i>Теоретический вопрос 17-20</i>

6	Производительность алгоритмов и аппаратная оптимизация	ОПК-4.1 ОПК-4.2 ОПК-5.1 ОПК-6.2 PL-2.1 AI S-1.1	Задания для самостоятельного решения 11-12	Теоретический вопрос 21-24
7	Мониторинг и управление высоконагруженными ИИ-сервисами	ОПК-4.1 ОПК-4.2 ОПК-5.1 ОПК-6.2 PL-2.1 AI S-1.1	Задания для самостоятельного решения 13-14	Теоретический вопрос 25-28

Показатели, критерии и шкала оценки сформированных компетенций

Соответствие **пороговому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **удовлетворительно /зачтено**):

ОПК-4 Способен находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем

ОПК-4.1 Разрабатывает и оптимизирует алгоритмы с учетом вычислительной сложности и аппаратных ограничений

Знать Базовые алгоритмы и структуры данных для обработки больших объемов информации

Основные метрики производительности алгоритмов (временная сложность, потребление памяти)

Принципы кэширования и предвыборки данных для оптимизации доступа

Уметь Анализировать сложность алгоритмов и выбирать оптимальные для high-load сценариев

Реализовывать базовые алгоритмы обработки потоковых данных

Применять простые методы оптимизации производительности вычислений

Владеть Навыками оценки производительности алгоритмов на тестовых наборах данных

Основными методами профилирования и выявления "узких мест" в вычислениях

Техниками базовой оптимизации алгоритмической сложности

ОПК-4.2 Тестирует и внедряет алгоритмы в реальные задачи, оценивая их точность и производительность

Знать Базовые методы тестирования производительности алгоритмов (юнит-тесты, нагрузочные тесты)

Основные метрики для оценки эффективности алгоритмов (время выполнения, потребление памяти)

Принципы мониторинга работы алгоритмов в production-среде

Уметь	<p>Проводить базовое тестирование алгоритмов на различных объемах данных</p> <p>Сравнивать производительность разных алгоритмических подходов для решения одной задачи</p> <p>Настраивать простой мониторинг для отслеживания производительности алгоритмов</p>
Владеть	<p>Навыками написания тестов для проверки корректности работы алгоритмов</p> <p>Основными методами измерения производительности алгоритмических решений</p> <p>Техниками сбора и анализа метрик работы алгоритмов в реальных условиях</p>
ОПК-5	Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности
ОПК-5.1	Применяет современные языки программирования и технологии для решения математических и вычислительных задач
Знать	<p>Основные конструкции языков программирования для high-load разработки (Python, Go)</p> <p>Принципы работы веб-серверов и протоколов HTTP/gRPC</p> <p>Базовые технологии кэширования и асинхронной обработки запросов</p>
Уметь	<p>Разрабатывать простые высоконагруженные сервисы на современных языках программирования</p> <p>Настраивать базовое кэширование и асинхронную обработку запросов</p> <p>Использовать инструменты мониторинга для отслеживания производительности приложений</p>
Владеть	<p>Навыками создания REST API с использованием современных фреймворков</p> <p>Методами настройки и использования систем кэширования (Redis)</p> <p>Практиками написания эффективного кода для high-load окружений</p>
ОПК-6	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения
ОПК-6.2	Применяет методы структурного и объектно-ориентированного программирования, создает модульные и масштабируемые программы
Знать	<p>Основные принципы объектно-ориентированного программирования и проектирования</p> <p>Паттерны проектирования для создания модульных систем</p> <p>Подходы к обеспечению масштабируемости программных решений</p>
Уметь	<p>Проектировать модульную архитектуру приложений с разделением ответственности</p> <p>Реализовывать масштабируемые компоненты с использованием ООП</p> <p>Применять принципы SOLID при разработке high-load систем</p>
Владеть	<p>Навыками создания чистого и поддерживаемого кода для высоконагруженных систем</p> <p>Методами рефакторинга для улучшения архитектуры существующих решений</p> <p>Техниками модульного тестирования компонентов high-load приложений</p>

PL-2	Способен применять JVM-совместимые языки программирования для решения задач в области ИИ
PL-2.1	<i>Разрабатывает и отлаживает прикладные решения разного уровня сложности и для широкого круга конечных пользователей с использованием JVM-совместимых языков программирования, тестирует, испытывает и оценивает качество таких решений</i>
Знать	<i>Принципы объектно-ориентированного и функционального программирования на JVM-совместимых языках (Kotlin, Scala). Архитектуру и основные компоненты JVM (JVM Memory Model, Garbage Collection). Фреймворки для построения высоконагруженных приложений (Spring Boot, Micronaut, Vert.x). Библиотеки для машинного обучения и обработки данных на JVM (Tribuo, Apache Spark MLlib, DL4J).</i>
Уметь	<i>Разрабатывать высокопроизводительные backend-сервисы и микросервисы на JVM-совместимых языках. Профилировать и оптимизировать JVM-приложения для работы с большими объемами данных и высокой параллельной нагрузкой. Интегрировать ML-модели, реализованные на Python, в JVM-приложения (например, с использованием JPure). Писать модульные и интеграционные тесты для обеспечения качества и надежности кода.</i>
Владеть	<i>Навыками разработки на одном из JVM-совместимых языков (Kotlin/Scala). Инструментами для профилирования и мониторинга JVM (JVisualVM, JMH, GC logs analysis). Навыками работы с системами сборки и управления зависимостями (Gradle, Maven). Практиками отладки и решения проблем производительности в JVM-среде.</i>
AI S-1	Способен управлять рисками в разработке систем ИИ, выстраивать управление безопасностью ИИ в компании с учетом этики ИИ
AI S-1.1	<i>Выявляет и моделирует угрозы на всём жизненном цикле ИИ-систем, оценивает и приоритизирует риски</i>
Знать	<i>Уязвимости высоконагруженных систем (DDoS, инъекции) Методы обеспечения кибербезопасности в ИИ Принципы работы систем мониторинга аномалий</i>
Уметь	<i>Проводить стресс-тесты и анализ уязвимостей Разрабатывать стратегии защиты данных Настраивать системы обнаружения вторжений (IDS/IPS)</i>
Владеть	<i>Инструментами (OWASP ZAP, Burp Suite) Навыками аудита безопасности ML-методами для детектирования аномалий</i>

Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Типовые задачи для самостоятельного решения:

1. Разработка микросервиса для рекомендаций ИИ. Создать сервис на FastAPI/Python, генерирующий персонализированные рекомендации (например, для интернет-магазина).. Реализовать кэширование предсказаний в Redis и балансировку нагрузки через Nginx.
2. Построение event-driven архитектуры для обработки данных. Настроить взаимодействие микросервисов через Kafka (например, сбор пользовательских действий для аналитики).
3. Автомасштабирование ML-сервиса в Kubernetes. Развернуть модель классификации текста в K8s, настроить HPA на основе CPU/RAM-метрик.
4. Шардирование MongoDB для хранения логов. Разделить коллекцию логов API по шардам, протестировать скорость запросов под нагрузкой (YCSB).
5. Оптимизация запросов к векторной БД. Развернуть Milvus/Pinecone для поиска похожих товаров, сравнить производительность при разных индексах (HNSW, IVF).
6. Колоночное хранилище для аналитики. Загрузить датасет в ClickHouse, оптимизировать запросы с агрегациями (GROUP BY, оконные функции).
7. Стриминг данных с Kafka + Flink. Настроить потоковую обработку кликов пользователей в реальном времени (подсчет CTR).
8. Распределённая обработка данных в Spark. Проанализировать логи сервера (например, подсчет 500-х ошибок по часам).
9. Тестирование отказоустойчивости ML-пайплайна. Сымитировать падение ноды в Kubernetes, проверить восстановление сервиса (Readiness/Liveness пробы).
10. Репликация PostgreSQL для сервиса аутентификации. Настроить master-slave репликацию, проверить работу при отказе master-ноды.
11. Ускорение inference модели на GPU. Развернуть PyTorch-модель с поддержкой CUDA, замерить скорость обработки запросов vs CPU.
12. Оптимизация C++-кода для предобработки данных. Реализовать алгоритм токенизации текста на C++ с PyBind11, сравнить производительность с Python.
13. Мониторинг дрейфа данных в ML-модели. Настроить Evidently для детектирования сдвигов в распределении входных признаков.
14. A/B-тестирование рекомендательных алгоритмов. Развернуть две версии модели (content-based vs collaborative filtering), сравнивать метрики через Grafana.

Зачетно-экзаменационные материалы для промежуточной аттестации (экзамен/зачет)

Вопросы для подготовки к зачету

1. Основные принципы проектирования высоконагруженных систем для ИИ-сервисов
2. Сравнение микросервисной и event-driven архитектур для обработки потоковых данных
3. Паттерны отказоустойчивости в распределенных системах (Circuit Breaker, Retry, Saga)
4. Оптимизация API-интерфейсов для ML-моделей (gRPC vs REST, бинарные протоколы)
5. Стратегии горизонтального масштабирования для ML-инференса
6. Автомасштабирование в Kubernetes: HPA, VPA и их настройка для ИИ-сервисов
7. Балансировка нагрузки для GPU-кластеров: особенности и решения
8. Шардирование векторных баз данных для поиска похожих объектов

9. Оптимизация колоночных хранилищ (ClickHouse, Cassandra) для аналитических запросов
10. Кэширование предсказаний ML-моделей: стратегии и инструменты (Redis, RedisAI)
11. Организация feature store для высоконагруженных ML-систем
12. Репликация и согласованность данных в распределенных БД (CAP-теорема)
13. Архитектура потоковой обработки данных для ИИ-приложений (Kafka + Flink/Spark)
14. Оптимизация ETL-пайплайнов для обучения моделей на больших данных
15. Распределенные вычисления для обработки временных рядов (архитектура, инструменты)
16. Организация Data Lake для хранения и обработки разнородных данных
17. Стратегии обеспечения отказоустойчивости ML-пайплайнов
18. Репликация и восстановление векторных баз данных
19. Тестирование на отказоустойчивость (chaos engineering для ИИ-систем)
20. Организация multi-region развертывания для глобальных сервисов
21. Ускорение инференса моделей: квантование, оптимизация графов (TensorRT, ONNX)
22. Профилирование и оптимизация Python-кода для высоконагруженных ИИ-сервисов
23. Использование GPU/TPU для ускорения обработки данных
24. Оптимизация сетевых взаимодействий в распределенных системах
25. Мониторинг дрейфа данных и концептуального сдвига в ML-моделях
26. Инструменты для логирования и анализа работы распределенных систем (ELK, OpenTelemetry)
27. A/B-тестирование и канареечные релизы для ML-моделей
28. Управление конфигурацией в высоконагруженных системах (ConfigMaps, Vault)

4.2 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Методические рекомендации, определяющие процедуры оценивания выполнения контрольных заданий:

Задание считается выполненным при выполнении следующих условий:

- предоставлен исходный код
- продемонстрирована работоспособность программы
- студент понимает исходный код и отвечает на вопросы по его организации.

Методические рекомендации к сдаче зачета

Процедура промежуточной аттестации проходит в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся ФГБОУ ВО «КубГУ».

Текущий контроль успеваемости студентов проводится в целях совершенствования и непрерывного контроля качества образовательного процесса, проверки усвоения учебного материала, активизации самостоятельной работы студентов, стимулирования их учебной работы, обеспечения эффективности образовательного процесса, предупреждения рисков отчисления студентов.

Текущий контроль знаний студентов осуществляется постоянно в течение всего семестра.

Виды текущего контроля: устный (письменный) опрос на занятиях; проверка выполнения домашних заданий; проведение контрольных работ; оценка активности студента на занятии.

Студенты очной формы обучения обязаны сдать зачет до начала экзаменационной сессии.

По результатам сдачи зачета выставляется оценка «зачтено»/ «не зачтено».

Выставление зачетов для студентов очной формы обучения проводится в период до экзаменационной сессии.

При отсутствии зачетной книжки у студента экзаменатор не имеет права принимать у него зачет/экзамен. Такой студент считается не явившимся на зачет/экзамен. В исключительных случаях, на основании распоряжения декана преподаватель может допустить студента к зачету/экзамену при наличии документа, удостоверяющего личность.

В целях объективного оценивания знаний во время проведения зачетов и экзаменов не допускается наличие у студентов посторонних предметов и технических устройств.

Студенту, использующему в ходе зачета неразрешенные источники и средства получения информации, выставляется неудовлетворительная оценка, и он удаляется из аудитории.

Во время зачета студенты могут пользоваться утвержденной рабочей программой учебной дисциплины, которая должна быть в наличии на экзамене, а также с разрешения экзаменатора справочной литературой и другими пособиями.

Студенты, нарушающие правила поведения при проведении зачетов и экзаменов, могут быть незамедлительно удалены из аудитории, к ним могут быть применены меры дисциплинарного воздействия.

На зачете/экзамене могут присутствовать ректор, проректор по учебной работе, декан факультета, заведующий кафедрой, которая обеспечивает учебный процесс по данной дисциплине. Присутствие на экзаменах и зачетах посторонних лиц без разрешения ректора или проректора по учебной работе не допускается.

После прослушивания лекции рекомендуется выполнить упражнения, приводимые в лекции для самостоятельной работы, а также выполнить на компьютере с использованием среды Python задачи, приводимые в лекции в качестве примеров.

При самостоятельной работе студентов необходимо изучить литературу, приведенную в перечнях, для осмысления вводимых понятий, анализа предложенных подходов и методов разработки параллельных программ. Разрабатывая решение новой задачи, студент должен уметь выбрать методы решения задачи с учетом целевой аппаратной платформы, проводить отладку и профилирование программы.

Критерии оценивания

Оценка	
Не зачтено	Зачтено
<ul style="list-style-type: none"> • если студент правильно решил менее 70 % задач и/или не имеет представление как решать остальные задачи • не знает значительной части теоретического материала, допускает существенные ошибки. 	<ul style="list-style-type: none"> • если студент правильно решил 70 % задач, имеет представление как решать остальные задачи • на теоретические вопросы дан развернутый ответ. Материал изложен в целом последовательно. Имеются логичные и аргументированные выводы

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4.3. Методические указания по организации вычислительной инфраструктуры

Условия применения:

- Курс рассчитан на студентов, имеющих базовые знания в области системного администрирования, сетей и программирования.
- Для выполнения работ требуется доступ к облачным платформам (AWS, GCP, Yandex Cloud) или мощному локальному кластеру с поддержкой виртуализации.
- Необходимо предварительное знакомство с Docker, Kubernetes и системами мониторинга.

Цели, задачи и ожидаемые результаты:

Цели:

- Создать отказоустойчивую и масштабируемую инфраструктуру, способную выдерживать экстремальные нагрузки.
- Обеспечить студентов практическими навыками работы с инструментами оркестрации, мониторинга и управления высоконагруженными системами.

Задачи преподавателя:

- Подготовить и предоставить студентам шаблоны Terraform/Ansible для автоматизированного развертывания стенда.
- Настроить централизованные системы сбора метрик (Prometheus) и логов (ELK Stack).
- Разработать сценарии для генерации нагрузки (с использованием Locust, Yandex Tank).

Ожидаемые результаты студентов:

- Умение самостоятельно разворачивать и настраивать кластер Kubernetes для высоконагруженных задач.
- Навыки конфигурирования автоматического масштабирования (HPA, Cluster Autoscaler).
- Способность настраивать и использовать комплексные системы мониторинга и алертинга.

Порядок реализации:

1. **Базовый стенд:** Развертывание минимального кластера Kubernetes с помощью kubectl или облачного провайдера.
2. **Инфраструктура мониторинга:** Установка и настройка стека Prometheus + Grafana + Alertmanager для сбора метрик с нод, подов и приложений.
3. **Логирование:** Развертывание ELK-стека (Elasticsearch, Logstash, Kibana) или его аналога (Loki) для агрегации и анализа логов.
4. **Генерация нагрузки:** Настройка и запуск скриптов на Locust для создания реалистичной нагрузки на тестовые приложения.
5. **Настройка автоматического масштабирования:** Конфигурирование Horizontal Pod Autoscaler (HPA) и Vertical Pod Autoscaler (VPA) для тестового веб-приложения.

Порядок проверки корректности:

Чек-лист для проверки инфраструктуры:

- Кластер Kubernetes находится в состоянии Ready.
- Все системные пода (CoreDNS, Metrics-Server) работают.
- Prometheus собирает метрики со всех нод и целевых приложений.
- В Grafana доступны дашборды для отслеживания нагрузки CPU, RAM, сети.
- При нагрузочном тестировании происходит автоматическое масштабирование подов приложения.

4.4. Методические указания по организации лабораторных работ

Условия применения:

- Лабораторные работы выполняются после освоения теоретического материала по соответствующей теме.
- Для выполнения необходим доступ к заранее развернутой вычислительной инфраструктуре (п. 4.3).
- Работы выполняются индивидуально или в малых группах (2-3 человека).

Цели, задачи и ожидаемые результаты:

Цели:

- Закрепить на практике принципы построения и оптимизации высоконагруженных систем.
- Сформировать навыки работы с современным инструментарием для обеспечения производительности и отказоустойчивости.

Задачи преподавателя:

- Разработать детальные задания для каждой лабораторной работы (в соответствии с РПД, п. 2.3.3).
- Подготовить тестовые сценарии и скрипты для проверки результатов.
- Обеспечить консультационную поддержку в процессе выполнения работ.

Ожидаемые результаты студентов:

- Умение проводить сравнительный анализ архитектур и технологий под нагрузкой.
- Навыки тонкой настройки СУБД, кэшей и веб-серверов для достижения максимальной производительности.
- Способность проводить стресс-тестирование и анализировать его результаты.

Порядок реализации:**Примеры лабораторных работ (согласно РПД):**

ЛР 1-2: Сравнение монолитной и микросервисной архитектуры под нагрузкой. Анализ CAP-теоремы.

ЛР 3-4: Настройка автомасштабирования и сравнение стратегий масштабирования БД.

ЛР 5-6: Оптимизация производительности векторных и колоночных хранилищ данных.

ЛР 7-8: Построение pipeline для обработки больших данных (Spark) и стриминга (Kafka).

ЛР 9-10: Тестирование отказоустойчивости в Kubernetes и репликации данных.

ЛР 11-12: Аппаратная оптимизация (CPU/GPU) и низкоуровневая оптимизация кода (C++).

ЛР 13-14: Мониторинг ML-моделей в продакшене и проведение A/B-тестов.

Порядок проверки корректности:**Чек-лист для оценки работы:**

- Предоставлен исходный код и конфигурационные файлы.
- Продемонстрирована работа системы в соответствии с заданием (например, показаны графики нагрузки и метрики масштабирования).
- Студент может пояснить принятые архитектурные и конфигурационные решения.
- Результаты нагрузочного тестирования соответствуют критериям, указанным в задании.

4.5. Методические указания по организации проектной деятельности студентов**Условия применения:**

- Проектная деятельность является завершающим этапом освоения дисциплины.
- Проекты выполняются малыми группами (3-4 человека) в течение семестра.
- Для реализации необходима вся инфраструктура, развернутая в рамках п. 4.3.

Цели, задачи и ожидаемые результаты:

- Интегрировать полученные знания и навыки для решения комплексной задачи проектирования и развертывания высоконагруженной системы.

- Развить навыки командной работы, проектного планирования и презентации результатов.

Задачи преподавателя:

- Сформировать портфель проектных заданий, имитирующих реальные бизнес-кейсы.
- Выступать в роли ментора, проводя регулярные code review и архитектурные сессии.
- Разработать четкие и измеримые критерии оценки итогового решения.

Ожидаемые результаты студентов:

- Готовая к работе высоконагруженная система, развернутая в кластере Kubernetes.
- Полный комплект документации, включая описание архитектуры, инструкции по развертыванию и отчет по нагрузочному тестированию.
- Умение командой спроектировать, реализовать и оттестировать систему, отвечающую требованиям производительности и надежности.

Порядок реализации:

Примеры проектных заданий:

1. **Проектирование высоконагруженного сервиса рекомендаций.** Разработка системы, включающей микросервисы для сбора данных, обучения модели и инференса, с кэшированием и балансировкой нагрузки.
2. **Создание платформы для онлайн-трансляций с чатом.** Реализация системы, способной обрабатывать тысячи одновременных подключений, с использованием WebSocket, Redis для pub/sub и горизонтальным масштабированием.
3. **Разработка высокопроизводительного API-шлюза.** Построение шлюза с кэшированием, rate limiting, Circuit Breaker и детализированным мониторингом всех входящих запросов.
4. **Построение ETL-пайплайна для обработки потоковых данных IoT-устройств.** Настройка приема, обработки и сохранения большого потока данных с использованием Kafka, Spark Streaming и высокопроизводительной БД.

Критерии оценки:

- **Архитектура и реализация (50%):** Соответствие принципам проектирования ВН-систем, грамотное использование паттернов, качество кода.
- **Производительность и масштабируемость (30%):** Результаты нагрузочного тестирования, эффективность настроек масштабирования, отказоустойчивость.
- **Документация и презентация (20%):** Полнота и ясность документации, качество выступления и ответов на вопросы.

Порядок проверки корректности:

- Проводится демо-сессия, на которой команда разворачивает систему с нуля и проводит нагрузочный тест.
- Преподаватель проверяет выполнение всех функциональных и нефункциональных требований, указанных в ТЗ.
- Оценивается способность команды аргументировать выбор технологий и архитектурных решений.

5. Перечень учебной литературы, информационных ресурсов и технологий

5.1 Учебная литература:

1. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2023. — 432 с. — (Высшее образование). — ISBN 978-5-534-07604-

2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/513067> (дата обращения: 08.06.2025).

2. Чернышев, С. А. Принципы, паттерны и методологии разработки программного обеспечения : учебное пособие для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2023. — 176 с. — (Высшее образование). — ISBN 978-5-534-14383-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/520097> (дата обращения: 08.06.2025).

3. «Архитектурные решения информационных систем : учебник для вузов / А. И. Водяхо, Л. С. Выговский, В. А. Дубенецкий, В. В. Цехановский. — 3-е изд., стер. — Санкт-Петербург : Лань, 2022. — 356 с. — ISBN 978-5-507-44710-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/254624> (дата обращения: 08.06.2025). — Режим доступа: для авториз. пользователей.»

4. . Кафка Д., Мурти Д. Нигам С. Микросервисная архитектура. Реализация и проектирование: Пер. с англ. — М.: ДМК-Пресс, 2017. — 432 с.

5. Фаулер М., Палмер Р., Льюис Джеймс. Микросервисы. Паттерны разработки и лучшие практики: Пер. с англ. — М.: ДМК-Пресс, 2018. — 510 с.

6. МакКроски Р. Микросервисы. Создание расширяемых и отказоустойчивых приложений: Пер. с англ. — М.: ДМК-Пресс, 2018. — 460 с.

7. Саммервиль Дж., Жерарди Р. Микросервисы в действии: Пер. с англ. — СПб.: Питер, 2018. — 304 с.

8. Ганри Ш. Микросервисы. Надежные и масштабируемые системы: Пер. с англ. — М.: ООО «ДМК Пресс», 2019. — 340 с.

9. Баджуска Д., Москетто Р. Микросервисы изнутри. Основы, инструменты и практические рекомендации: Пер. с англ. — М.: ДМК-Пресс, 2020. — 480 с.2.

5.2. Периодические издания:

1. Базы данных компании «Ист Вью» <http://dlib.eastview.com>
2. Электронная библиотека GREBENNIKON.RU <https://grebennikon.ru/>

5.3. Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

Электронно-библиотечные системы (ЭБС):

1. ЭБС «ЮРАЙТ» <https://urait.ru/>
2. ЭБС «УНИВЕРСИТЕТСКАЯ БИБЛИОТЕКА ОНЛАЙН» <http://www.biblioclub.ru/>
3. ЭБС «BOOK.ru» <https://www.book.ru>
4. ЭБС «ZNANIUM.COM» www.znanium.com
5. ЭБС «ЛАНЬ» <https://e.lanbook.com>

Профессиональные базы данных

1. Scopus <http://www.scopus.com/>
2. ScienceDirect <https://www.sciencedirect.com/>
3. Журналы издательства Wiley <https://onlinelibrary.wiley.com/>
4. Научная электронная библиотека (НЭБ) <http://www.elibrary.ru/>
5. Полнотекстовые архивы ведущих западных научных журналов на Российской платформе научных журналов НЭИКОН <http://archive.neicon.ru>
6. Национальная электронная библиотека (доступ к Электронной библиотеке диссертаций Российской государственной библиотеки (РГБ) <https://rusneb.ru/>
7. Президентская библиотека им. Б.Н. Ельцина <https://www.prilib.ru/>
8. База данных CSD Кембриджского центра кристаллографических данных (CCDC) <https://www.ccdc.cam.ac.uk/structures/>
9. Springer Journals: <https://link.springer.com/>

10. Springer Journals Archive: <https://link.springer.com/>
11. Nature Journals: <https://www.nature.com/>
12. Springer Nature **Protocols and Methods**:
<https://experiments.springernature.com/sources/springer-protocols>
13. Springer Materials: <http://materials.springer.com/>
14. Nano Database: <https://nano.nature.com/>
15. Springer eBooks (i.e. 2020 eBook collections): <https://link.springer.com/>
16. "Лекториум ТВ" <http://www.lektorium.tv/>
17. Университетская информационная система РОССИЯ <http://uisrussia.msu.ru>

Информационные справочные системы

1. Консультант Плюс - справочная правовая система (доступ по локальной сети с компьютеров библиотеки)

Ресурсы свободного доступа

1. КиберЛенинка <http://cyberleninka.ru/>;
2. Американская патентная база данных <http://www.uspto.gov/patft/>
3. Министерство науки и высшего образования Российской Федерации
<https://www.minobrnauki.gov.ru/>;
4. Федеральный портал "Российское образование" <http://www.edu.ru/>;
5. Информационная система "Единое окно доступа к образовательным ресурсам"
<http://window.edu.ru/>;
6. Единая коллекция цифровых образовательных ресурсов <http://school-collection.edu.ru/> .
7. Проект Государственного института русского языка имени А.С. Пушкина
"Образование на русском" <https://pushkininstitute.ru/>;
8. Справочно-информационный портал "Русский язык" <http://gramota.ru/>;
9. Служба тематических толковых словарей <http://www.glossary.ru/>;
10. Словари и энциклопедии <http://dic.academic.ru/>;
11. Образовательный портал "Учеба" <http://www.ucheba.com/>;
12. [Законопроект "Об образовании в Российской Федерации". Вопросы и ответы http://xn--273--84d1f.xn--p1ai/voprosy_i_otvety](http://xn--273--84d1f.xn--p1ai/voprosy_i_otvety)

Собственные электронные образовательные и информационные ресурсы КубГУ

1. Электронный каталог Научной библиотеки КубГУ <http://megapro.kubsu.ru/MegaPro/Web>
2. Электронная библиотека трудов ученых КубГУ
<http://megapro.kubsu.ru/MegaPro/UserEntry?Action=ToDb&idb=6>
3. Среда модульного динамического обучения <http://moodle.kubsu.ru>
4. База учебных планов, учебно-методических комплексов, публикаций и конференций
<http://infoneeds.kubsu.ru/>
5. Библиотека информационных ресурсов кафедры информационных образовательных технологий <http://mschool.kubsu.ru/>;
6. Электронный архив документов КубГУ <http://docspace.kubsu.ru/>
7. Электронные образовательные ресурсы кафедры информационных систем и технологий в образовании КубГУ и научно-методического журнала "ШКОЛЬНЫЕ ГОДЫ"
<http://icdau.kubsu.ru/>

5.4 Перечень информационно-коммуникационных технологий

– Проверка домашних заданий и консультирование посредством электронной почты.

- Использование электронных презентаций при проведении лекционных занятий
- Система MOODLE
- Проверка домашних заданий и консультирование посредством ЭОИС КубГУ

5.5 Перечень лицензионного и свободно распространяемого программного обеспечения

OpenOffice
 Компилятор C++
 Oracle VirtualBox 6
 VMware Workstation 16
 Putty 0.76 или Kitty 0.76
 FileZilla 3.57.0
 WinSCP 5.19
 Advanced port scanner 2.5
 Python 3 (3.7 И 3.9)
 numpy 1.22.0
 opencv 4.5.5
 Keras 2.7.0
 Tensor flow 2.7.0
 matplotlib 3.5.1
 PyCharm 2021
 Cuda Toolkit 11.6
 Фреймворк Django
 Firefox, любая версия
 Putty, любая версия
 Visual Studio Code, версия 1.52+
 Eclipse PHP Development Tools, версия 2020-06+
 Плагин Remote System Explorer (RSE) для Eclipse PDT
[JetBrains PHP Storm](#)
 GIT
 Java Version 8 Update 311
 Clojure 1.10.3.1029.ps1
 SWI Prolog 8.4
 IntelliJ Idea IDE 2021
 Mozilla Firefox 96
 Google Chrome 97
 GitHub Desktop 2.9
 PHP Storm 2021
 FileZilla 3.57.0
 Putty 0.76

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

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

Кейсы ПАО «Сбербанк»

Кейс 1: Высоконагруженная платформа для генеративного ИИ инвестиционных обзоров

Описание:

Разработка масштабируемой системы для автоматического составления инвестиционных обзоров на основе LLM, способной обрабатывать сотни запросов одновременно в режиме реального времени. Система должна интегрироваться с источниками биржевых данных и обеспечивать генерацию аналитических отчетов объемом 500-1000 слов.

Цель:

Создать отказоустойчивую платформу, обеспечивающую низкую задержку инференса ML-моделей при пиковых нагрузках, с эффективным управлением GPU-ресурсами и кэшированием предсказаний.

Ожидаемый результат:

Архитектура микросервисной системы с горизонтальным масштабированием, показатели latency < 2 сек при 1000 RPS, реализованное кэширование предсказаний в Redis, конфигурация мониторинга производительности в Grafana.

Кейс 2: Распределенная система анализа жалоб клиентов в реальном времени

Описание:

Построение системы семантического анализа обращений клиентов, обрабатывающей тысячи запросов в минуту из чатов, мобильного приложения и жалобных форм. Система должна выделять темы жалоб, определять тональность и критичность инцидентов.

Цель:

Реализовать event-driven архитектуру с горизонтальным масштабированием NLP-микросервисов, обеспечить обработку потоковых данных через Apache Kafka и балансировку нагрузки между инстансами моделей.

Ожидаемый результат:

Отказоустойчивая система с throughput > 5000 сообщений/мин, метриками доступности 99.9%, дашбордом мониторинга в реальном времени и механизмами автоматического восстановления при сбоях.

Кейс 3: Высокопроизводительная платформа для генерации синтетических банковских данных

Описание:

Создание pipeline генерации синтетических банковских данных для обучения ML-моделей, способного производить миллионы транзакций и клиентских профилей с сохранением статистических характеристик реальных данных.

Цель:

Разработать распределенную систему генерации и валидации данных с использованием Apache Spark, обеспечить хранение больших объемов данных в колоночных хранилищах и параллельную обработку с сохранением консистентности.

Ожидаемый результат:

Pipeline с производительностью > 1 млн записей/час, интеграция с ClickHouse для аналитических запросов, система мониторинга качества синтетических данных и метрики соответствия реальным распределениям.

Кейсы от «АВА ЛАБ»

Кейс 1: Высоконагруженная ВІ-система с RAG на естественном языке

Описание:

Разработка интерфейса на естественном языке для ВІ-системы Fastboard, обеспечивающего мгновенные ответы на аналитические запросы к большим массивам данных в ClickHouse с использованием RAG-архитектуры.

Цель:

Оптимизировать производительность векторного поиска при высокой частоте запросов, реализовать кэширование результатов и эмбедингов, обеспечить масштабируемость сервисов LLM и векторных БД.

Ожидаемый результат:

Система с временем отклика < 1 сек при 100+ concurrent запросах, реализованное многоуровневое кэширование, дашборд мониторинга производительности и качества поиска.

Кейс 2: Мультимодальный агент для анализа строительных площадок в реальном времени

Описание:

Создание системы мониторинга строительных объектов, обрабатывающей видеопотоки с камер, голосовые и текстовые запросы от специалистов с применением компьютерного зрения и NLP.

Цель:

Обеспечить обработку потокового видео в реальном времени, распределенную обработку мультимодальных данных и низкую задержку при взаимодействии с агентом через оптимизацию инференса ML-моделей на GPU-кластерах.

Ожидаемый результат:

Система с поддержкой 50+ одновременных видеопотоков, временем обработки запросов < 3 сек, масштабируемой архитектурой для компьютерного зрения и NLP-сервисов.

Кейс 3: Высоконагруженная платформа прогнозирования сроков сдачи объектов

Описание:

Разработка системы прогнозирования сроков сдачи строительных объектов на основе анализа текстовых отчетов и визуальных данных, обрабатывающей информацию с сотен строительных площадок.

Цель:

Реализовать параллельную обработку больших объемов изображений и документов, создать масштабируемую архитектуру для анализа временных рядов и обеспечить высокую доступность системы для непрерывного мониторинга.

Ожидаемый результат:

Платформа с производительностью обработки > 1000 документов/час, временем формирования прогноза < 10 сек, системой мониторинга производительности и метриками точности прогнозов > 85%.

Общие рекомендации по реализации кейсов:

- Использовать микросервисную архитектуру с четким разделением ответственности
- Применять горизонтальное масштабирование и автоматическое масштабирование в Kubernetes (HPA)
- Внедрить кэширование на всех уровнях (Redis, Memcached) для снижения нагрузки на базы данных
- Использовать асинхронную обработку запросов через message brokers (Kafka, RabbitMQ)
- Настроить комплексный мониторинг производительности (Prometheus, Grafana) и распределенную трассировку
- Обеспечить отказоустойчивость через репликацию данных и механизмы Circuit Breaker
- Оптимизировать работу с базами данных: шардирование, индексы, репликация
- Применять стратегии балансировки нагрузки (round-robin, least connections, IP hash)

7. Материально-техническое обеспечение по дисциплине (модулю)

№	Вид работ	Наименование учебной аудитории, ее оснащенность оборудованием и техническими средствами обучения
1.	Лекционные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
2.	Лабораторные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, проектором, программным обеспечением
3.	Групповые (индивидуальные) консультации	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
4.	Текущий контроль, промежуточная аттестация	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
5.	Самостоятельная работа	Кабинет для самостоятельной работы, оснащенный компьютерной техникой с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета.

Примечание: Конкретизация аудиторий и их оснащение определяется ОПОП.