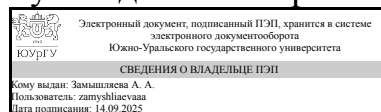


ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ:
Руководитель направления



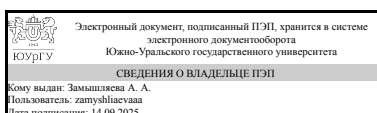
А. А. Замышляева

РАБОЧАЯ ПРОГРАММА

дисциплины 1.О.32 Высокопроизводительные параллельные вычисления
для направления 01.03.02 Прикладная математика и информатика
уровень Бакалавриат
форма обучения очная
кафедра-разработчик Центр ОП топ-уровня в сфере ИИ "ВиртУм"

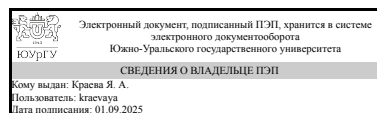
Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 01.03.02 Прикладная математика и информатика, утверждённым приказом Минобрнауки от 10.01.2018 № 9

Зав.кафедрой разработчика,
д.физ.-мат.н., проф.



А. А. Замышляева

Разработчик программы,
к.физ.-мат.н., доцент



Я. А. Краева

1. Цели и задачи дисциплины

Целью дисциплины является формирование у студентов базовых знаний, умений и навыков разработки параллельных программ для современных высокопроизводительных вычислительных архитектур (многоядерные процессоры, графические процессоры, вычислительные кластеры) с применением современных парадигм и инструментов программирования (OpenMP, MPI, CUDA) в области машинного и глубокого обучения.

Краткое содержание дисциплины

Понятие параллельных вычислений. Необходимость параллельных вычислений в искусственном интеллекте. Примеры вычислительно емких задач из различных предметных областей, иллюстрирующих интеграцию параллельных высокопроизводительных вычислений и искусственного интеллекта. Виды параллельной обработки данных. Обзор параллельных вычислительных систем. Оценка производительности многопроцессорных систем. Классификации параллельных вычислительных систем: классификация Флинна, классификация MIMD-систем. Модели программирования для различных архитектур. Понятие параллельного алгоритма. Цикл разработки параллельного алгоритма. Примеры параллельных алгоритмов. Оценка эффективности параллельного алгоритма (ускорение, эффективность, стоимость). Законы Амдала и Густавсона—Барсиса. Факторы, препятствующие масштабированию алгоритмов. Модель программирования в общей памяти. Модель параллелизма fork-join. Стандарт OpenMP. Принципы использования. Основные директивы и функции. Модель передачи сообщений в системах с распределенной памятью. Модели SPMD и MPMD запуска параллельных программ. Стандарт MPI. Принципы использования. Основные функции. Введение в программирование на графических процессорах. Архитектура и устройство графических процессоров. Основные различия CPU и GPU. Программно-аппаратная архитектура CUDA. Особенности выполнения GPU программ. Асинхронность в CUDA. Обработка ошибок. Иерархия памяти CUDA. Атомарные операции в CUDA. Разбор параллельной реализации матричного умножения как ключевой операции в обучении и инференсе нейронных сетей. CUDA потоки и события. Использование CUDA потоков для реализации асинхронных операций. Интегрированные среды разработки с поддержкой CUDA, инструменты для отладки и профилирования. Оптимизация GPU программ. Фреймворки с поддержкой CUDA (PyTorch, TensorFlow). Библиотеки для ускорения обучения и инференса моделей машинного и глубокого обучения (cuBLAS, cuSolver, cuFFT, cuDNN, TensorRT, PyCUDA и др.). Программирование систем с несколькими графическими процессорами. Распределенное обучение моделей машинного и глубокого обучения (Distributed Deep Learning, DDL). Синхронное и асинхронное DDL. Централизованное и децентрализованное DDL. Схемы распределенного обучения ML и DL моделей.

2. Компетенции обучающегося, формируемые в результате освоения дисциплины

Планируемые результаты освоения ОП ВО (компетенции)	Планируемые результаты обучения по дисциплине
--	--

ОПК-6 Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности	Знает: теоретические основы параллельных вычислений и распределённых систем; принципы построения и функционирования суперкомпьютеров и вычислительных кластеров Умеет: выбирать подходящую модель параллельного программирования и оптимальную архитектуру вычислительной системы для конкретного класса задач; применять на практике методы и средства разработки параллельных программ Имеет практический опыт: разработки параллельных программ на российских платформах с использованием различных средств: библиотек языков и систем программирования; стандарты OpenMP, MPI и CUDA
ПК-4 [PL-3] Способен применять языки программирования C/C++ для решения задач в области ИИ	Знает: - [И-3, ПУ] основы синтаксиса языка C/C++, основы построения систем ИИ, общие принципы параллельных вычислений, возможности и ограничения исполнения программ на GPU и FPGA Умеет: - [И-1, ПУ] решать проблемы одновременного доступа к данным из нескольких потоков, грамотно применяет атомарные операции и механизм блокировок [И-3, ПУ] использовать готовые фреймворки ИИ (PyTorch, TensorFlow и пр.) Имеет практический опыт: - [И-1, ПУ] оценки производительности, профилирования кода и устранения найденных узких мест
ПК-10 [BD-4] Способен применять различные модели и (или) технологии обработки данных	Умеет: - [И-1, ЭУ] организовать централизованное хранилище данных (Data Lake) Имеет практический опыт: - [И-1, ЭУ] организации распределенного хранения данных, параллельной обработки, а также обработки потоковых данных [И-1, ЭУ] Руководства разработкой решений с элементами ИИ с применением различных технологий обработки данных

3. Место дисциплины в структуре ОП ВО

Перечень предшествующих дисциплин, видов работ учебного плана	Перечень последующих дисциплин, видов работ
1.О.18 Современные языки программирования высокого уровня, 1.О.11 Программирование на C++, 1.О.12 Операционные системы	Не предусмотрены

Требования к «входным» знаниям, умениям, навыкам студента, необходимым при освоении данной дисциплины и приобретенным в результате освоения предшествующих дисциплин:

Дисциплина	Требования
1.О.11 Программирование на C++	Знает: синтаксис языка C++ и технологии разработки прикладного ПО на языке C++, -[И-1, ПУ] средства разработки языка C++ для создания прикладной системы ИИ; библиотеки OpenCV для C++, TensorFlow C++, основные концепции и синтаксис языка программирования C++ Умеет: разрабатывать прикладные программные решения на языке C++, -[И-1, ПУ] использовать средства разработки языка C++ для создания прикладной системы ИИ, стандартные библиотеки C++, реализовывать эффективные алгоритмы и решать практические задачи средствами C++ Имеет практический опыт: создания приложений на языке C++ с соблюдением принципов ООП и code style, -[И-2, ПУ] разработки и отладки прикладных решений на языке программирования C++ с учетом контроля памяти, многопоточности, профилирования кода, высокой производительности, написания программного кода на C++, его отладки и профилирования производительности
1.О.18 Современные языки программирования высокого уровня	Знает: базовые понятия и парадигмы современных языков программирования высокого уровня, функциональные возможности интегрированных сред разработки прикладного и системного программного обеспечения на языках высокого уровня, особенности работы компиляторов и компоновщиков под различные операционные системы, наборы инструкций для системных утилит автоматической сборки программ, средства мониторинга вычислительных ресурсов компьютерных программ, основные подходы к разработке прикладных алгоритмов в рамках парадигмы структурного программирования на языке высокого уровня, базовые синтаксические конструкции языка программирования высокого уровня: операторы, выражения, блоки, ветвления, циклы; методы оценки сложности алгоритмов, функциональные возможности стандартной библиотеки языка высокого уровня, общие сведения об аппаратных и системных возможностях вычислительной техники для оптимизации программного обеспечения Умеет: разрабатывать программы с применением различных языков программирования, выбирать подходящие инструменты для конкретной задачи, использовать возможности современных интегрированных сред разработки прикладного и системного программного обеспечения на языках высокого уровня для разработки алгоритмов и программ, использовать утилиты автоматической сборки и развертывания программ в операционных системах, проектировать

	архитектуру программного обеспечения, использовать современные языки программирования для разработки программного обеспечения, разрабатывать прикладное программное обеспечение в рамках парадигмы структурного программирования на языке программирования высокого уровня с применением основных синтаксических конструкций и функциональных возможностей стандартной библиотеки языка высокого уровня, устанавливать и настраивать среду разработки для выбранного языка программирования Имеет практический опыт: работы с современными интегрированными средами разработки прикладного и системного программного обеспечения на языках высокого уровня, разработки, отладки и развёртывания программного обеспечения в операционных системах семейства Unix и Windows, разработки, тестирования и отладки программ с использованием современных языков программирования, инструментов и технологий, разработки прикладного программного обеспечения, отладки, поиска и устранения ошибок программного кода, оценки сложности алгоритмов, использования возможностей стандартной библиотеки, сторонних библиотек программного кода, настройки и интеграции программных решений с аппаратным обеспечением и внешними устройствами
1.О.12 Операционные системы	Знает: основные концепции современных операционных систем, основные концепции, принципы, теории и факты, связанные с построением современных операционных систем, структуру современных операционных систем, принципы работы их основных компонентов: ядра, менеджера памяти, подсистемы ввода-вывода, файловой системы, основные средства, предоставляемые современными операционными системами прикладным программам для решения системных и пользовательских задач, -[И-3, ПУ] основные средства мониторинга и диагностики ОС Умеет: использовать стандартные инструменты современных ОС при решении задач профессиональной деятельности, использовать стандартные инструменты современных операционных систем при решении практических задач, использовать стандартные интерфейсы современных операционных систем для решения задач профессиональной деятельности, устанавливать и настраивать операционную систему, создавать прикладные программы в терминах API ОС, использовать интерфейсы прикладного программирования, предоставляемые современными операционными

	системами, -[И-1, ПУ] выбирать операционную систему и ее параметры с учетом требований к развертыванию и сопровождению моделей искусственного интеллекта в среде эксплуатации Имеет практический опыт: работы с основными видами интерфейсов ОС - командным и API, работы с основными компонентами современных операционных систем, создания командных файлов, использования API операционных систем при разработке прикладных программ для решения задач профессиональной деятельности, использования основных видов интерфейсов операционной системы Windows, создания прикладных программ с использованием API Windows, -[И-2, БУ] реализации скриптов и настройки операционной системы для автоматизации запуска, мониторинга и устойчивой работы сервисов искусственного интеллекта в среде эксплуатации [И-3, ПУ] использования средств мониторинга и диагностики ОС для анализа стабильности и производительности сервисов искусственного интеллекта в среде эксплуатации
--	---

4. Объём и виды учебной работы

Общая трудоемкость дисциплины составляет 3 з.е., 108 ч., 54,25 ч. контактной работы

Вид учебной работы	Всего часов	Распределение по семестрам в часах
		Номер семестра
		6
Общая трудоёмкость дисциплины	108	108
<i>Аудиторные занятия:</i>	48	48
Лекции (Л)	32	32
Практические занятия, семинары и (или) другие виды аудиторных занятий (ПЗ)	0	0
Лабораторные работы (ЛР)	16	16
<i>Самостоятельная работа (СРС)</i>	53,75	53,75
Подготовка к промежуточным тестам	18	18
Подготовка к зачету	10	10
Изучение дополнительных материалов в области параллельных высокопроизводительных вычислений в AI	25,75	25.75
Консультации и промежуточная аттестация	6,25	6,25
Вид контроля (зачет, диф.зачет, экзамен)	-	зачет

5. Содержание дисциплины

№ раздела	Наименование разделов дисциплины	Объем аудиторных занятий по видам в часах
-----------	----------------------------------	---

		Всего	Л	ПЗ	ЛР
1	Параллельные вычисления и искусственный интеллект	2	2	0	0
2	Параллельные вычислительные системы	4	4	0	0
3	Понятие параллельного алгоритма	2	2	0	0
4	Параллельное программирование на общей памяти. Стандарт OpenMP	6	2	0	4
5	Параллельное программирование на распределенной памяти. Стандарт MPI	8	4	0	4
6	Параллельное программирование на графических процессорах. Стандарт CUDA	26	18	0	8

5.1. Лекции

№ лекции	№ раздела	Наименование или краткое содержание лекционного занятия	Кол-во часов
1	1	Понятие параллельных вычислений. Необходимость параллельных вычислений в искусственном интеллекте. Примеры вычислительно емких задач из различных предметных областей, иллюстрирующих интеграцию параллельных высокопроизводительных вычислений и искусственного интеллекта. Виды параллельной обработки данных.	2
2	2	Обзор параллельных вычислительных систем. Оценка производительности многопроцессорных систем.	2
3	2	Классификации параллельных вычислительных систем: классификация Флинна, классификация MIMD-систем. Модели программирования для различных архитектур.	2
4	3	Понятие параллельного алгоритма. Цикл разработки параллельного алгоритма. Примеры параллельных алгоритмов. Оценка эффективности параллельного алгоритма (ускорение, эффективность, стоимость). Законы Амдала и Густавсона—Барсиса. Факторы, препятствующие масштабированию алгоритмов.	2
5	4	Модель программирования в общей памяти. Модель параллелизма fork-join. Стандарт OpenMP. Принципы использования. Основные директивы и функции.	2
6-7	5	Модель передачи сообщений в системах с распределенной памятью. Модели SPMD и MPMD запуска параллельных программ. Стандарт MPI. Принципы использования. Основные функции.	4
8	6	Введение в программирование на графических процессорах. Архитектура и устройство графических процессоров. Основные различия CPU и GPU.	2
9	6	Программно-аппаратная архитектура CUDA. Особенности выполнения GPU программ. Асинхронность в CUDA. Обработка ошибок.	2
10-11	6	Иерархия памяти CUDA. Атомарные операции в CUDA. Разбор параллельной реализации матричного умножения как ключевой операции в обучении и инференсе нейронных сетей.	4
12	6	CUDA потоки и события. Использование CUDA потоков для реализации асинхронных операций.	2
13	6	Интегрированные среды разработки с поддержкой CUDA, инструменты для отладки и профилирования. Оптимизация GPU программ.	2
14-15	6	Фреймворки с поддержкой CUDA (PyTorch, TensorFlow). Библиотеки для ускорения обучения и инференса моделей машинного и глубокого обучения (cuBLAS, cuSolver, cuFFT, cuDNN, TensorRT, PyCUDA и др.).	4
16	6	Программирование систем с несколькими графическими процессорами. Распределенное обучение моделей машинного и глубокого обучения	2

		(Distributed Deep Learning, DDL). Синхронное и асинхронное DDL. Централизованное и децентрализованное DDL. Схемы распределенного обучения ML и DL моделей.	
--	--	--	--

5.2. Практические занятия, семинары

Не предусмотрены

5.3. Лабораторные работы

№ занятия	№ раздела	Наименование или краткое содержание лабораторной работы	Кол-во часов
1-2	4	Разработка параллельных программ с использованием OpenMP	4
3-4	5	Разработка параллельных программ с использованием MPI	4
5-6	6	Разработка параллельных программ с использованием CUDA	4
7-8	6	Оптимизация ML и DL моделей с использованием фреймворков и библиотек с поддержкой CUDA	4

5.4. Самостоятельная работа студента

Выполнение CPC			
Подвид CPC	Список литературы (с указанием разделов, глав, страниц) / ссылка на ресурс	Семестр	Кол-во часов
Подготовка к промежуточным тестам	Основная и дополнительная литература. Презентации к лекциям.	6	18
Подготовка к зачету	Основная и дополнительная литература. Презентации к лекциям.	6	10
Изучение дополнительных материалов в области параллельных высокопроизводительных вычислений в AI	1. Lewis N., Bez J.L., Byna S. I/O in Machine Learning Applications on HPC Systems: A 360-degree Survey // ACM Comput. Surv., 2025. Vol. 57, no. 10. Article 256. https://doi.org/10.1145/3722215 . 2. Moldoveanu M., Zaidi A. In-Network Learning: Distributed Training and Inference in Networks // Entropy, 2023. Vol. 25, no. 6. Article 920. URL: https://doi.org/10.3390/e25060920 . 3. Ben-Nun T., Hoeffler T. Demystifying Parallel and Distributed Deep Learning: An In-depth Concurrency Analysis // ACM Comput. Surv., 2019. Vol. 52, no. 4. Article 65. URL: https://doi.org/10.1145/3320060 . 4. Sze V., Chen Y.-H., Yang T.-J., Emer J.S. Efficient Processing of Deep Neural Networks // Morgan & Claypool Publishers, 2020. Synthesis Lectures on Computer Architecture. ISBN 978-3-031-00638-8. URL: https://doi.org/10.2200/S01004ED1V01Y202004CAC050 . 5. Kurni M., Krishnamaneni R., Murthy A.N., Sen S. High-Performance Computing and Artificial Intelligence: Synergy for the Future. United Kingdom: Cambridge Scholars Publishing, 2025. 225 p. ISBN 978-1-0364-4774-8. 6. Raghuwanshi M., Borkar P., Jhaveri R.H., Raut R. Parallel and High-Performance Computing in Artificial Intelligence (Advances in Computational Collective Intelligence). New York: Auerbach Publications. 340 p. ISBN 978-1040332702. URL:	6	25,75

	https://doi.org/10.1201/9781003425458 . 7. Terzo O., Martinovič J. HPC, Big Data, and AI Convergence Towards Exascale. 2022. 322 p. ISBN URL: https://doi.org/10.1201/9781003176664 .		
--	---	--	--

6. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации

Контроль качества освоения образовательной программы осуществляется в соответствии с Положением о балльно-рейтинговой системе оценивания результатов учебной деятельности обучающихся.

6.1. Контрольные мероприятия (КМ)

№ КМ	Се-местр	Вид контроля	Название контрольного мероприятия	Вес	Макс. балл	Порядок начисления баллов	Учитывается в ПА
1	6	Текущий контроль	Тест 1. Параллельные вычисления и искусственный интеллект	0,2	5	Тест проводится по окончании изучения темы лекции. Прохождение компьютерного теста оценивается от 0 до 5 баллов. Тест состоит из 5 равнозначных вопросов, правильный ответ на один вопрос дает один балл. Ограничение по времени - 20 минут.	зачет
2	6	Текущий контроль	Тест 2. Обзор параллельных вычислительных систем	0,2	5	Тест проводится по окончании изучения темы лекции. Прохождение компьютерного теста оценивается от 0 до 5 баллов. Тест состоит из 5 равнозначных вопросов, правильный ответ на один вопрос дает один балл. Ограничение по времени - 20 минут.	зачет
3	6	Текущий контроль	Тест 3. Классификация параллельных вычислительных систем	0,2	5	Тест проводится по окончании изучения темы лекции. Прохождение компьютерного теста оценивается от 0 до 5 баллов. Тест состоит из 5 равнозначных вопросов, правильный ответ на один вопрос дает один балл. Ограничение по времени - 20 минут.	зачет
4	6	Текущий контроль	Тест 4. Параллельный алгоритм и оценка его эффективности	0,2	5	Тест проводится по окончании изучения темы лекции. Прохождение компьютерного теста оценивается от 0 до 5 баллов. Тест состоит из 5 равнозначных вопросов, правильный ответ на один вопрос дает один балл. Ограничение по времени - 20 минут.	зачет
5	6	Текущий контроль	Тест 5. Параллельное программирование в стандарте OpenMP	0,4	5	Тест проводится по окончании изучения темы лекции. Прохождение компьютерного теста оценивается от 0 до 5 баллов. Тест состоит из 5 равнозначных вопросов, правильный ответ на один вопрос дает один балл. Ограничение по времени - 20 минут.	зачет

6	6	Текущий контроль	Тест 6. Параллельное программирование в стандарте MPI	0,4	5	Тест проводится по окончании изучения темы лекции. Прохождение компьютерного теста оценивается от 0 до 5 баллов. Тест состоит из 5 равнозначных вопросов, правильный ответ на один вопрос дает один балл. Ограничение по времени - 20 минут.	зачет
7	6	Текущий контроль	Тест 7. Параллельное программирование на GPU на стандарте CUDA	0,4	5	Тест проводится по окончании изучения темы лекции. Прохождение компьютерного теста оценивается от 0 до 5 баллов. Тест состоит из 5 равнозначных вопросов, правильный ответ на один вопрос дает один балл. Ограничение по времени - 20 минут.	зачет
8	6	Текущий контроль	Тест 8. Иерархия памяти и атомарные операции CUDA	0,4	5	Тест проводится по окончании изучения темы лекции. Прохождение компьютерного теста оценивается от 0 до 5 баллов. Тест состоит из 5 равнозначных вопросов, правильный ответ на один вопрос дает один балл. Ограничение по времени - 20 минут.	зачет
9	6	Текущий контроль	Тест 9. CUDA потоки и события	0,4	5	Тест проводится по окончании изучения темы лекции. Прохождение компьютерного теста оценивается от 0 до 5 баллов. Тест состоит из 5 равнозначных вопросов, правильный ответ на один вопрос дает один балл. Ограничение по времени - 20 минут.	зачет
10	6	Текущий контроль	Тест 10. Средства отладки и профилировки CUDA-программ	0,4	5	Тест проводится по окончании изучения темы лекции. Прохождение компьютерного теста оценивается от 0 до 5 баллов. Тест состоит из 5 равнозначных вопросов, правильный ответ на один вопрос дает один балл. Ограничение по времени - 20 минут.	зачет
11	6	Текущий контроль	Тест 11. Фреймворки и библиотеки ML и DL с поддержкой CUDA	0,4	5	Тест проводится по окончании изучения темы лекции. Прохождение компьютерного теста оценивается от 0 до 5 баллов. Тест состоит из 5 равнозначных вопросов, правильный ответ на один вопрос дает один балл. Ограничение по времени - 20 минут.	зачет
12	6	Текущий контроль	Тест 12. Multi-GPU программирование	0,4	5	Тест проводится по окончании изучения темы лекции. Прохождение компьютерного теста оценивается от 0 до 5 баллов. Тест состоит из 5 равнозначных вопросов, правильный ответ на один вопрос дает один балл. Ограничение по времени - 20 минут.	зачет
13	6	Текущий контроль	Лабораторная работа 1. Разработка параллельных программ с использованием OpenMP	2	10	Защита лабораторной работы осуществляется индивидуально студентом. Выполнение лабораторной работы оценивается от 0 до 10 баллов. Лабораторная работа состоит из пяти заданий, каждое из которых	зачет

					оценивается определенным количеством баллов в зависимости от уровня его сложности. Максимальная оценка выставляется в случае, если студент предоставил корректное решение всех заданий лабораторной работы, за которые начисляются следующие баллы: – правильно реализовано задание 1 (установлены ПО и библиотеки для запуска программ, использующих CUDA): 0.5 баллов; – правильно реализовано задание 2 (реализована OpenMP-программа для поиска максимального значения среди минимальных элементов строк матрицы, проведены вычислительные эксперименты, выполнена визуализация результатов): 1 балл; – правильно реализовано задание 3 (реализована OpenMP-программа для выполнения операции редукции, проведены вычислительные эксперименты, выполнена визуализация результатов): 1.5 балла; – правильно реализовано задание 4 (реализован параллельный метод Гаусса для решения уравнения линейной регрессии с помощью OpenMP, проведены вычислительные эксперименты, выполнена визуализация результатов): 3 балла; – правильно реализовано задание 5 (реализован параллельный алгоритм кластеризации K-means с помощью OpenMP, проведены вычислительные эксперименты, выполнена визуализация результатов): 4 балла.		
14	6	Текущий контроль	Лабораторная работа 2. Разработка параллельных программ с использованием MPI	2	10	Защита лабораторной работы осуществляется индивидуально студентом. Выполнение лабораторной работы оценивается от 0 до 10 баллов. Лабораторная работа состоит из пяти заданий, каждое из которых оценивается определенным количеством баллов в зависимости от уровня его сложности. Максимальная оценка выставляется в случае, если студент предоставил корректное решение всех заданий лабораторной работы, за которые начисляются следующие баллы: – правильно реализовано задание 1 (создан проект в среде MS Visual Studio с поддержкой MPI, выполнена проверка подключения MPI): 0.5	зачет

					баллов; – правильно реализовано задание 2 (реализована MPI-программа для вычисления скалярного произведения векторов, проведены вычислительные эксперименты, выполнена визуализация результатов): 1.5 балла; – правильно реализовано задание 3 (реализована MPI-программа для транспонирования квадратной матрицы, проведены вычислительные эксперименты, выполнена визуализация результатов): 2 балла; – правильно реализовано задание 4 (реализована MPI-программа для выполнения операции глобального суммирования элементов вектора с использованием пересылок данных типа точка-точка, проведены вычислительные эксперименты, выполнена визуализация результатов): 3 балла; – правильно реализовано задание 5 (реализована MPI-программа для параллельной предварительной обработки большого набора данных изображений, проведены вычислительные эксперименты, выполнена визуализация результатов): 3 балла.	
15	6	Текущий контроль	Лабораторная работа 3. Разработка параллельных программ с использованием CUDA	2	10	Зачет

						выполнения операции свертки изображения с заданным ядром, выполнены оптимизации для вычислений, проведены вычислительные эксперименты, выполнена визуализация результатов): 1.5 балла; – правильно реализовано задание 4 (реализован параллельный алгоритм классификации KNN с использованием CUDA, выполнено профилирование CUDA-программы с помощью NVIDIA Nsight Compute, проведены вычислительные эксперименты, выполнена визуализация результатов): 3 балла; – правильно реализовано задание 5 (реализована нейронная сеть с одним полносвязным слоем для классификации данных, проведены вычислительные эксперименты, выполнена визуализация результатов): 4 балла.	
16	6	Текущий контроль	Лабораторная работа 4. Оптимизация ML и DL моделей с использованием фреймворков и библиотек с поддержкой CUDA	2	10	Защита лабораторной работы осуществляется индивидуально студентом. Выполнение лабораторной работы оценивается от 0 до 10 баллов. Лабораторная работа состоит из пяти заданий, каждое из которых оценивается определенным количеством баллов в зависимости от уровня его сложности. Максимальная оценка выставляется в случае, если студент предоставил корректное решение всех заданий лабораторной работы, за которые начисляются следующие баллы: – правильно реализовано задание 1 (реализованы базовые операции над тензорами с использованием PyTorch/TensorFlow, проведены вычислительные эксперименты, выполнена визуализация результатов): 0.5 баллов; – правильно реализовано задание 2 (реализована CUDA-ускоренное вычисление функции активации ReLU для каждого элемента входной матрицы с помощью библиотеки NVIDIA cuDNN, проведены вычислительные эксперименты, выполнена визуализация результатов): 1 балл; – правильно реализовано задание 3 (реализована свертка изображения с заданным ядром с использованием	зачет

					библиотеки cuFFT, проведены вычислительные эксперименты, выполнена визуализация результатов): 1.5 балла; – правильно реализовано задание 4 (реализована пользовательская функция активации и ее производная на CUDA и интегрирована в нейронную сеть, построенную с использованием PyTorch/TensorFlow, проведены вычислительные эксперименты, выполнена визуализация результатов): 3 балла; – правильно реализовано задание 5 (выполнена интеграция NVIDIA TensorRT в существующие пайплайны инференса, разработанные на основе PyTorch/TensorFlow, для ускорения процесса предсказания, проведены вычислительные эксперименты, выполнена визуализация результатов): 4 балла.		
17	6	Промежуточная аттестация	Итоговое тестирование	-	20	Промежуточная аттестация проводится во время зачета в виде компьютерного теста. Тест состоит из 20 равноценных вопросов, позволяющих оценить сформированность компетенций по курсу в целом, правильный ответ на один вопрос дает один балл. Время на прохождение теста – не менее 45 мин.	зачет

6.2. Процедура проведения, критерии оценивания

Вид промежуточной аттестации	Процедура проведения	Критерии оценивания
зачет	При оценивании результатов учебной деятельности обучающегося по дисциплине используется балльно-рейтинговая система оценивания результатов учебной деятельности обучающихся (Положение о БРС утверждено приказом ректора от 24.05.2019 г. № 179, в редакции приказа ректора от 10.03.2022 г. № 25-13/09). Процедура прохождения промежуточной аттестации осуществляется согласно Положению о текущем контроле успеваемости и промежуточной аттестации (приказ ректора от 27.02.2024 № 33-13/09). Оценка за дисциплину формируется на основе полученных оценок за контрольно-рейтинговые мероприятия текущего контроля следующим образом: • Зачтено: Величина рейтинга обучающегося по дисциплине 60...100 %. • Не зачтено: Величина рейтинга обучающегося по дисциплине 0...59 %. Если студент согласен с оценкой, полученной по результатам текущего контроля, то он может в день, предшествующий промежуточной аттестации, дать свое согласие на автомат в личном кабинете. В случае явки студента на промежуточную аттестацию, давшего свое согласие на	В соответствии с пп. 2.5, 2.6 Положения

3	Основная литература	ЭБС издательства Лань	Тоуманнен Б. Программирование GPU при помощи Python и CUDA. Москва: ДМК Пресс, 2020. 252 с. ISBN 978-5-97060-821-0. URL: https://e.lanbook.com/book/179469 .
4	Основная литература	ЭБС издательства Лань	Гергель В.П. Теория и практика параллельных вычислений: учебное пособие. 2-е изд. Москва: ИНТУИТ, 2016. 500 с. ISBN 978-5-94774-645-7. URL: https://e.lanbook.com/book/100527 .
5	Дополнительная литература	ЭБС издательства Лань	Шарден Б., Массарон Л., Боскетти А. Крупномасштабное машинное обучение вместе с Python. Москва: ДМК Пресс, 2018. 358 с. ISBN 978-5-97060-506-6. URL: https://e.lanbook.com/book/105836 .
6	Дополнительная литература	ЭБС издательства Лань	Антонов А.С. Параллельное программирование с использованием технологии MPI: учебное пособие. 2-е изд. Москва: ИНТУИТ, 2016. 83 с. URL: https://e.lanbook.com/book/100359 .
7	Дополнительная литература	ЭБС издательства Лань	Левин М.П. Параллельное программирование с использованием OpenMP: учебное пособие. 2-е изд. Москва: ИНТУИТ, 2016. 133 с. ISBN 978-5-94774-857-4. URL: https://e.lanbook.com/book/100358 .

Перечень используемого программного обеспечения:

1. ФГАОУ ВО "ЮУрГУ (НИУ)"-Портал "Электронный ЮУрГУ" (<https://edu.susu.ru>)(бессрочно)
2. -Visual Studio 2017 Community(бессрочно)
3. -Python(бессрочно)

Перечень используемых профессиональных баз данных и информационных справочных систем:

Нет

8. Материально-техническое обеспечение дисциплины

Вид занятий	№ ауд.	Основное оборудование, стенды, макеты, компьютерная техника, предустановленное программное обеспечение, используемое для различных видов занятий
Зачет		Персональный компьютер, доступ к сети интернет
Лекции		Проектор, персональный компьютер, доступ к сети интернет
Лабораторные занятия		Персональный компьютер, доступ к сети интернет, установленный VS Code