



Федеральное государственное образовательное бюджетное учреждение высшего образования
«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)

Кафедра информационных технологий
Факультет информационных технологий и анализа больших данных

Документ подписан усиленной неквалифицированной электронной подписью.

Организация: «ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ
РОССИЙСКОЙ ФЕДЕРАЦИИ»

Сертификат: tz2XIYLS+o1oM3erdYrijMTgNbFCMdHh

Утверждено: Проректор по учебной и методической работе, Е.А. Каменева

Дата: 26.06.2026 г.

К.А. Маковейчук

Технологии параллельного программирования

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки

09.03.03 - Прикладная информатика,

Образовательная программа «Прикладные информационные системы в экономике и
финансах»

Профиль:

«Прикладные информационные системы в экономике и финансах»

Рекомендовано

*Факультет информационных технологий и анализа больших данных
(протокол № 09 от 16.06.2026 г.)*

Одобрено

*Кафедра информационных технологий
(протокол № 04 от 25.05.2026 г.)*

© Москва 2026



Содержание

1. Наименование дисциплины	3
2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	3
3. Место дисциплины в структуре образовательной программы	6
4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	6
5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий	7
5.1. Содержание дисциплины	7
5.2. Учебно – тематический план	9
5.3. Содержание семинаров, практических занятий	10
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	12
6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	12
6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю	14
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	15
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	6
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	9
10. Методические указания для обучающихся по освоению дисциплины	29
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)	32
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	32



1. Наименование дисциплины

Технологии параллельного программирования

2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
ПКП-1	Способен разрабатывать и отлаживать программный код	Выполняет формализацию и алгоритмизацию поставленных задач для разработки программного кода	Знать: основные модели и технологии параллельного программирования; принципы декомпозиции вычислительных задач и распараллеливания алгоритмов; особенности архитектур систем с общей и распределённой памятью; подходы к анализу эффективности параллельных алгоритмов; базовые технологии параллельного программирования (OpenMP, MPI, GPU-вычисления). Уметь: выполнять формализацию вычислительных задач с учётом возможности параллельного выполнения; разрабатывать алгоритмы с использованием технологий параллельного программирования; определять потенциальный уровень параллелизма задачи; выбирать подходящие средства распараллеливания для различных вычислительных архитектур; анализировать эффективность и масштабируемость параллельных решений.
		Оформляет программный код в соответствии с установленными требованиями	Знать: требования к структуре и оформлению программного кода; принципы документирования программных проектов; методы тестирования и отладки параллельных



Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
			программ; особенности сопровождения многопоточных и распределённых приложений; основные инструменты разработки и контроля версий. Уметь: оформлять программный код в соответствии с требованиями читаемости и сопровождаемости; использовать средства тестирования и отладки параллельных программ; документировать программные решения и результаты вычислительных экспериментов; применять системы контроля версий при разработке программных проектов; выявлять и устранять ошибки синхронизации и конкурентного доступа.
ПКП-6	Способен участвовать в промышленной разработке программного обеспечения	Работает в соответствии с промышленными методологиями разработки	Знать: основные подходы и методологии промышленной разработки программного обеспечения; особенности разработки высокопроизводительных и параллельных приложений; этапы жизненного цикла программного обеспечения; принципы коллективной разработки и организации вычислительных экспериментов; методы оценки производительности и масштабируемости программных решений. Уметь: применять элементы современных методологий разработки при создании параллельных программ; организовывать процесс разработки и тестирования вычислительных приложений; проводить вычислительные эксперименты и анализировать результаты; учитывать требования производительности и масштабируемости при разработке программного обеспечения; взаимодействовать с



Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
			участниками проекта при разработке параллельных решений.
		Применяет инструменты поддержки методологий разработки и использует навыки ведения документации в соответствии с требованиями методологии	Знать: инструменты поддержки коллективной разработки программного обеспечения; принципы ведения технической документации; средства контроля версий и сопровождения программных проектов; подходы к фиксации параметров вычислительных экспериментов и результатов тестирования; инструменты автоматизации сборки и запуска программ. Уметь: использовать системы контроля версий при разработке параллельных программ; документировать структуру проекта и результаты вычислительных экспериментов; оформлять отчёты по тестированию и производительности программ; организовывать хранение и воспроизводимость результатов вычислений; использовать инструменты сопровождения разработки.



3. Место дисциплины в структуре образовательной программы

Дисциплина «Технологии параллельного программирования» относится к «Модулю "разработке распределенных приложений"».

4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся

очно-заочная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 7 (в часах)
Общая трудоемкость дисциплины	3/108	108
Контактная работа – Аудиторные занятия	34	34
<i>Лекции</i>	<i>16</i>	<i>16</i>
<i>Семинары, практические занятия</i>	<i>18</i>	<i>18</i>
<i>Самостоятельная работа</i>	<i>74</i>	<i>74</i>
Вид текущего контроля	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	Зачет	Зачет



5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий

5.1. Содержание дисциплины

Тема 1. Введение в «Технологии параллельного программирования»

История развития параллелизма в архитектуре ЭВМ. Высокопроизводительные вычисления. Конвейерная обработка данных и команд. Многофункциональная обработка периферийными процессорами. Векторная обработка. Суперкомпьютеры. Классификация вычислительных систем Флинна. Вычислительные системы с общей и с распределенной памятью. Законы Амдала и Густафсона. Обзор задач, требующих суперкомпьютеров. Технологии параллелизма. Иерархическая организация памяти в компьютере. Локальность по обращению. Анализ задачи умножения матриц.

Тема 2. Основы OpenMP

Особенности программирования для систем с общей памятью. Понятие процесса, потока и многопоточности. Технология OpenMP, ее особенности и компоненты. Основные сведения о спецификациях набора стандарта OpenMP. Директивы параллельного программирования. Модели памяти и классы переменных в OpenMP. Режимы выполнения многопоточных программ. Вложенный параллелизм.

Тема 3. Директивы распределения и синхронизации работы OpenMP

Распараллеливание выполнения циклов. Распределение работы на основе независимых задач. Синхронизация выполнения различных потоков. Распределение операторов одного структурного блока между нитями. Копирование значений приватных переменных.

Тема 4. Основы векторизации исполняемого кода в OpenMP

Проблемы векторизации. Способы векторизации. Директивы для работы с параллелизмом по данным (векторизация) в OpenMP.

Тема 5. Технологии параллельного программирования для графических процессоров (GPU)

Возможные преимущества вычислений на графическом процессоре. Задача компьютерной визуализации трехмерных сцен. Архитектура графического процессора (GPU). Средства разработки для графического процессора. Python в работе с GPU.



Основы неоднородных (гетерогенных) вычислений. Уровни параллельности в архитектуре GPU. Гетерогенное программирование на PyOpenCL. Построение приложений при помощи PyOpenCL и PyCUDA.

Тема 6. Основы технологии MPI

Особенности программирования для систем с распределенной памятью. Технология MPI. Основы MPI на Python. Синхронные и асинхронные примитивы взаимодействия. Операции типа точка-точка. Операции коллективного взаимодействия. Параллельный алгоритм умножения матрицы на вектор. Операции сочетания параллельных результатов вычислений. Примитивы синхронизации между процессами. Закон Амдала. Оценки предельно возможного ускорения параллельных алгоритмов.



5.2. Учебно – тематический план

Очно-заочная форма обучения

п/ п	Наименование тем (разделов) дисциплины	Трудоемкость в часах				
		Всего	Контактная работа - Аудиторная работа			Самостоятельная работа
			Общая, в т.ч.:	Лекции	Семинары, практические занятия	
1	Введение в «Технологии параллельного программирования»	14	4	2	2	10
2	Основы OpenMP	14	4	2	2	10
3	Директивы распределения и синхронизации работы OpenMP	16	4	2	2	12
4	Основы векторизации исполняемого кода в OpenMP	14	4	2	2	10
5	Технологии параллельного программирования для графических процессоров (GPU)	26	10	4	6	16
6	Основы технологии MPI	24	8	4	4	16
В целом по дисциплине		108	34	16	18	74



5.3. Содержание семинаров, практических занятий

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Введение в «Технологии параллельного программирования»	Суперкомпьютеры. Классификация вычислительных систем Флинна. Технологии параллелизма. Иерархическая организация памяти в компьютере. Локальность по обращению. Анализ задачи умножения матриц.	Практикум по решению задач по тематике занятия и коллективное обсуждение решений. Дискуссия «Суперкомпьютеры – ТОП 100: анализ»
Основы OpenMP	Основные сведения о спецификациях набора стандарта OpenMP. Директивы параллельного программирования. Модели памяти и классы переменных в OpenMP. Режимы выполнения многопоточных программ. Вложенный параллелизм.	Практикум по решению задач по тематике занятия и коллективное обсуждение решений. Выполнение индивидуальных заданий, тестирование.
Директивы распределения и синхронизации работы OpenMP	Распараллеливание выполнения циклов. Распределение работы на основе независимых задач. Синхронизация выполнения различных потоков. Распределение операторов одного структурного блока между нитями. Копирование значений приватных переменных.	Практикум по решению задач по тематике занятия и коллективное обсуждение решений. Выполнение индивидуальных заданий.
Основы векторизации исполняемого кода в OpenMP	Директивы для работы с параллелизмом по данным (векторизация) в OpenMP.	Практикум по решению задач по тематике занятия и коллективное обсуждение решений. Выполнение индивидуальных заданий, тестирование.



Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Технологии параллельного программирования для графических процессоров (GPU)	Python в работе с GPU. Основы неоднородных (гетерогенных) вычислений. Уровни параллельности в архитектуре GPU. Гетерогенное программирование на PyOpenCL. Построение приложений при помощи PyOpenCL и PyCUDA.	Практикум по решению задач по тематике занятия и коллективное обсуждение решений. Выполнение индивидуальных заданий, тестирование.
Основы технологии MPI	Основы MPI на Python. Синхронные и асинхронные примитивы взаимодействия. Операции типа точка-точка. Операции коллективного взаимодействия. Параллельный алгоритм умножения матрицы на вектор. Операции сочетания параллельных результатов вычислений. Примитивы синхронизации между процессами.	Практикум по решению задач по тематике занятия и коллективное обсуждение решений. Выполнение индивидуальных заданий, тестирование.



6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Введение в «Технологии параллельного программирования»	История развития параллелизма в архитектуре ЭВМ. Конвейерная обработка данных и команд. Многофункциональная обработка периферийными процессорами. Векторная обработка. Вычислительные системы с общей и с распределенной памятью. Обзор задач, требующих суперкомпьютеров.	Работа с текстом лекции, разбор вопросов и заданий по теме занятия; изучение рекомендованных к занятию литературных источников.
Основы OpenMP	Особенности программирования для систем с общей памятью. Понятие процесса, потока и многопоточности. Технология OpenMP, ее особенности и компоненты.	Работа с текстом лекции, разбор вопросов и заданий по теме занятия; изучение рекомендованных к занятию литературных источников.
Директивы распределения и синхронизации работы OpenMP	Примеры задач, допускающих распараллеливание выполнения циклов. Сравнение эффективности распараллеливания циклов (директива For) и распределения работы на основе независимых задач (директива Task) на примере различных задач.	Работа с текстом лекции, разбор вопросов и заданий по теме занятия; изучение рекомендованных к занятию литературных источников.
Основы векторизации исполняемого кода в	Проблемы векторизации. Способы векторизации.	Работа с текстом лекции, разбор вопросов и заданий по теме занятия; изучение рекомендованных к занятию



Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
OpenMP		литературных источников.
Технологии параллельного программирования для графических процессоров (GPU)	Возможные преимущества вычислений на графическом процессоре. Задача компьютерной визуализации трехмерных сцен. Архитектура графического процессора (GPU). Средства разработки для графического процессора.	Работа с текстом лекции, разбор вопросов и заданий по теме занятия; изучение рекомендованных к занятию литературных источников.
Основы технологии MPI	Особенности программирования для систем с распределенной памятью. Технология MPI. Закон Амдала. Оценки предельно возможного ускорения параллельных алгоритмов.	Работа с текстом лекции, разбор вопросов и заданий по теме занятия; изучение рекомендованных к занятию литературных источников.



6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

Примерные задания контрольной работы

Задание 1. После выполнения выделенного параллельного участка в программе происходит неявная барьерная синхронизация параллельно работающих потоков. Их дальнейшее выполнение происходит только тогда, когда все они достигнут данной точки. Если в подобной задержке нет необходимости, опция `nowait` позволяет потокам, уже дошедшим до конца участка, продолжить выполнение без синхронизации с остальными. Реализуйте программу, иллюстрирующую описанную ситуацию.

Реализуйте программу на C++ с подключением технологии OpenMP, и проверьте получаемый результат, обоснуйте результат в выводах.

Можно использовать компилятор g++ совместно с установленной для него реализацией технологии OpenMP, либо IDE MS Visual Studio Community с настройками, разрешающими OpenMP, в свойствах проекта, либо использовать любой другой доступный компилятор C++ с OpenMP, в том числе облачный, например, <https://www.codingame.com/playgrounds/54443/openmp/playground>

Задание 2. Выполнить умножение матриц, используя любые изученные технологии распараллеливания и ускорения вычислений (OpenMP, Mpi4py, PyOpenCL). Сравнить производительность программы для матриц разного размера и при разном количестве потоков (для OpenMP); рабочих процессов (для MPI), для CPU и GPU (для PyOpenCL).

Дополнительная информация

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях Кафедры информационных технологий Факультета информационных технологий и анализа больших данных.



7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы содержится в разделе 2. «Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине».

Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний

Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
ПКП-1. Способен разрабатывать и отлаживать программный код	Выполняет формализацию и алгоритмизацию поставленных задач для разработки программного кода	Знать: основные модели и технологии параллельного программирования; принципы декомпозиции вычислительных задач и распараллеливания алгоритмов; особенности архитектур систем с общей и распределённой памятью; подходы к анализу	1. Разработать параллельный алгоритм обработки данных с использованием OpenMP для системы с общей памятью. 2. Выполнить анализ возможности распараллеливания вычислительной задачи и обосновать выбор технологии реализации (OpenMP, MPI, GPU). 3. Реализовать параллельную обработку матриц или векторов и провести сравнение последовательной и параллельной версий программы. 4. Оценить влияние числа потоков/процессов на производительность программы.



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		эффективности параллельных алгоритмов; базовые технологии параллельного программирования (OpenMP, MPI, GPU-вычисления). Уметь: выполнять формализацию вычислительных задач с учётом возможности параллельного выполнения; разрабатывать алгоритмы с использованием технологий параллельного программирования; определять потенциальный уровень параллелизма задачи; выбирать подходя-	



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		щие средства распараллеливания для различных вычислительных архитектур; анализировать эффективность и масштабируемость параллельных решений.	
	Оформляет программный код в соответствии с установленными требованиями	Знать: требования к структуре и оформлению программного кода; принципы документирования программных проектов; методы тестирования и отладки параллельных программ; особенности сопровождения многопоточных и распределённых приложений;	1. Выполнить документирование разработанного параллельного приложения с описанием алгоритма, параметров запуска и результатов тестирования. 2. Провести тестирование и отладку многопоточной программы, выявить и устранить ошибки синхронизации. 3. Оформить проект параллельной программы с использованием системы контроля версий и структурированного представления исходного кода. 4. Подготовить отчёт по результатам сравнения производительности последовательной и параллельной реализации алгоритма.



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		основные инструменты разработки и контроля версий. Уметь: оформлять программный код в соответствии с требованиями читаемости и сопровождаемости; использовать средства тестирования и отладки параллельных программ; документировать программные решения и результаты вычислительных экспериментов; применять системы контроля версий при разработке программных проектов; выявлять и	



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		устранять ошибки синхронизации и конкурентного доступа.	
ПКП-6. Способен участвовать в промышленной разработке программного обеспечения	Работает в соответствии с промышленными методологиями разработки	Знать: основные подходы и методологии промышленной разработки программного обеспечения; особенности разработки высокопроизводительных и параллельных приложений; этапы жизненного цикла программного обеспечения; принципы коллективной разработки и организации вычислительных экспериментов; методы	1. Выполнить разработку параллельного приложения по этапам: постановка задачи, проектирование, реализация, тестирование и анализ производительности. 2. Провести сравнительный анализ различных технологий распараллеливания для заданной вычислительной задачи. 3. Подготовить описание pipeline вычислительного эксперимента и результатов оценки производительности программы. 4. Выполнить анализ масштабируемости приложения при изменении числа потоков или процессов.



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		оценки производительности и масштабируемости программных решений. Уметь: применять элементы современных методологий разработки при создании параллельных программ; организовывать процесс разработки и тестирования вычислительных приложений; проводить вычислительные эксперименты и анализировать результаты; учитывать требования производительности и масштабируемости при разра-	



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		ботке программного обеспечения; взаимодействовать с участниками проекта при разработке параллельных решений.	
	Применяет инструменты поддержки методологий разработки и использует навыки ведения документации в соответствии с требованиями методологии	Знать: инструменты поддержки коллективной разработки программного обеспечения; принципы ведения технической документации; средства контроля версий и сопровождения программных проектов; подходы к фиксации параметров вычислительных экспериментов и результатов тестирования;	1. Оформить репозиторий проекта параллельной программы с документацией и примерами запуска. 2. Подготовить отчёт по вычислительному эксперименту с описанием параметров среды выполнения и полученных результатов. 3. Выполнить фиксацию изменений проекта с использованием системы контроля версий. 4. Подготовить документацию по разработанному параллельному приложению.



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		инструменты автоматизации сборки и запуска программ. Уметь: использовать системы контроля версий при разработке параллельных программ; документировать структуру проекта и результаты вычислительных экспериментов; оформлять отчёты по тестированию и производительности программ; организовывать хранение и воспроизводимость результатов вычислений; использовать	



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		инструменты сопровождения разработки.	



Примеры практико-ориентированных заданий

1. Напишите программу, которая находит строку с максимальной суммой элементов в матрице размером 8x8 с использованием mpi4py. Матрица случайных целых чисел от 1 до 100 генерируется в процессе с рангом 0 и распределяется по строкам между процессами с использованием метода Scatter. Каждый процесс вычисляет сумму элементов своей строки и передаёт её в процесс 0 с использованием метода Gather. Процесс 0 определяет строку с максимальной суммой, её индекс и саму строку, которые выводятся на экран.
2. Напишите программу, которая выполняет поэлементное умножение двух векторов (размерность 1 миллион элементов). Измерьте время выполнения этой операции как для последовательной, так и для параллельной версии с использованием OpenMP. Сравните время выполнения программы с использованием OpenMP и без него. Для замера времени можно использовать функции `omp_get_wtime()` или `clock()`, в зависимости от выбранного подхода.
3. Напишите программу, которая сравнивает время вычисления некой математической операции, а именно сложения двух векторов из элементов с плавающей точкой. Для выполнения такого сопоставления одна и та же операция должна быть выполнена в двух отдельных функциях. Первая функция выполняется только в имеющемся ЦПУ, в то время как вторая функция (с определением внутри нее функция ядра) написана с применением PyOpenCL для использования установленной карты GPU. Данный тест выполняется для вектора с размером в 10 000 элементов. Для замера времени можно использовать библиотеку `time` или `clock()`, в зависимости от выбранного подхода.

Примеры вопросов для подготовки к промежуточной аттестации

Примерные вопросы для подготовки к зачету

1. История появления параллельной и высокопроизводительной обработки информации. Суперкомпьютеры.
 2. Классификация вычислительных систем Флинна. Примеры.
 3. Технологии параллелизма. Классификация, примеры.
 4. Принцип иерархической организации памяти в компьютере. Примеры.
 5. Локальность по обращению. Принципы локальности памяти.
 6. Анализ задачи умножения матриц.
 7. Основные сведения о спецификациях набора стандарта OpenMP.
 8. Директивы параллельного программирования OpenMP.
 9. Модели памяти и классы переменных в OpenMP.



10. Режимы выполнения многопоточных программ.
11. Вложенный параллелизм в OpenMP.
12. Директивы распределения и синхронизации работы OpenMP
13. Распараллеливание выполнения циклов в OpenMP: директива и принцип организации.
14. Распределение работы на основе независимых задач в OpenMP: директива и принцип организации.
15. Синхронизация выполнения различных потоков.
16. Распределение операторов одного структурного блока между нитями.
17. Копирование значений приватных переменных.
18. Основы векторизации исполняемого кода в OpenMP
19. Директивы для работы с параллелизмом по данным (векторизация) в OpenMP 4.0.
20. Технологии параллельного программирования для графических процессоров (GPU)
21. Python в работе с GPU: реализации библиотек, сравнение.
22. Основы неоднородных (гетерогенных) вычислений.
23. Архитектура GPU.
24. Гетерогенное программирование на PyOpenCL и PyCUDA.
25. Построение приложений при помощи PyOpenCL и PyCUDA.
26. Основы технологии MPI
27. Основы MPI для Python.
28. Синхронные и асинхронные примитивы взаимодействия MPI.
29. Операции типа точка-точка в MPI.
30. Операции коллективного взаимодействия в MPI.
31. Параллельный алгоритм умножения матрицы на вектор: реализации на MPI.
32. Операции сочетания параллельных результатов вычислений в MPI.
33. Примитивы синхронизации между процессами в MPI.



8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. Земляная, Е. В. Введение в параллельное программирование на основе технологий MPI и OpenMP [Электронный ресурс] : учебное пособие / Земляная Е. В., Башахин М. В. Дубна : Государственный университет «Дубна», 2023 101 с. Книга из коллекции Государственный университет «Дубна» - Информатика <https://e.lanbook.com/book/369416> ISBN 978-5-89847-696-0. [БИК ID: RU-LAN-BOOK-369416]
2. Юрина, Т. А. Параллельное программирование [Электронный ресурс] / Юрина Т. А. Омск : СибАДИ, 2023 106 с. Книга из коллекции СибАДИ - Информатика <https://e.lanbook.com/book/407159>. [БИК ID: RU-LAN-BOOK-407159]
3. Малявко, Александр Антонович Параллельное программирование на основе технологий openmp, cuda, opencl, mpi : учебник для вузов / А. А. Малявко. 3-е изд., испр. и доп Электрон. дан. Москва : Юрайт, 2026 135 с (Высшее образование) URL: <https://urait.ru/bcode/585284> (дата обращения: 27.02.2026). Режим доступа: Электронно-библиотечная система Юрайт, для авториз. пользователей <https://urait.ru/bcode/585284> ISBN 978-5-534-14116-0 : 589.00. [БИК ID: RU2fURAIT2f585284]

Дополнительная литература:

1. Золкин, А.Л. Развертывание центров обработки данных с поддержкой GPU-кластеров : Учебное пособие / А.Л. Золкин, Р.А. Вербицкий Электрон. дан. Москва : Русайнс, 2025 181 с. Режим доступа: book.ru Internet access <https://book.ru/book/959499> ISBN 978-5-466-09908-9. [БИК ID: RU\bookru\bibl\959499]
2. Гафаров, Фаиль Мубаракович Нейронные сети в PyTorch : Учебное пособие / Казанский (Приволжский) федеральный университет Казань : Казанский (Приволжский) федеральный университет, 2024 106 с. ВО - Бакалавриат <https://znanium.ru/catalog/document?id=453352>. [БИК ID: RU\infra-m\znanium\bibl\2173433]
3. Минязев, Р. Ш. Параллельное программирование (MPI, OpenMP, CUDA) [Электронный ресурс] : учебно-методическое пособие / Минязев Р. Ш. Казань : КНИТУ-КАИ, 2021 68 с. Рекомендовано к изданию Учебно-методическим управлением КНИТУ-КАИ Книга из коллекции КНИТУ-КАИ



- Информатика <https://e.lanbook.com/book/264890> ISBN 978-5-7579-2567-7. [БИК ID: RU-LAN-BOOK-264890]
4. Ерохин, В. В. Совершенствование параллельного программирования на CUDA C для GPU [Электронный ресурс] : монография / Ерохин В. В., Горин Д. С., Бунина Л. В. Москва : РТУ МИРЭА, 2025 377 с. Книга из коллекции РТУ МИРЭА - Информатика <https://e.lanbook.com/book/504826> ISBN 978-5-7339-2565-3. [БИК ID: RU-LAN-BOOK-504826]
 5. Елесина, С. И. Технология параллельного программирования OpenMP [Электронный ресурс] : учебное пособие / Елесина С. И. Рязань : РГРТУ, 2021 48 с. Книга из коллекции РГРТУ - Информатика <https://e.lanbook.com/book/220409>. [БИК ID: RU-LAN-BOOK-220409]
 6. Горожанина, Е. И. Высокопроизводительные вычисления и анализ больших данных [Электронный ресурс] : учебное пособие / Горожанина Е. И. Самара : ПГУТИ, 2022 132 с. Книга из коллекции ПГУТИ - Информатика <https://e.lanbook.com/book/411386>. [БИК ID: RU-LAN-BOOK-411386]
 7. Ильин, Д. Ю. Высокопроизводительные вычисления в информационных системах [Электронный ресурс] / Ильин Д. Ю. Москва : РТУ МИРЭА, 2023 66 с. Книга из коллекции РТУ МИРЭА - Информатика <https://e.lanbook.com/book/382733> ISBN 978-5-7339-1946-1. [БИК ID: RU-LAN-BOOK-382733]

Нормативные правовые акты:

1. Федеральный закон от 27.07.2006 № 149-ФЗ «Об информации, информационных технологиях и о защите информации».
2. Федеральный закон от 27.07.2006 № 152-ФЗ «О персональных данных».
3. Указ Президента Российской Федерации от 10.10.2019 № 490 «О развитии искусственного интеллекта в Российской Федерации».
4. Кодекс этики в сфере искусственного интеллекта, разработанный Альянсом в сфере искусственного интеллекта.

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/> (<http://library.fa.ru/files/elibfa.pdf>)
2. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
3. Электронно-библиотечная система "Университетская библиотека ОНЛАЙН" <http://biblioclub.ru/>
4. Электронно-библиотечная система Znanium <http://www.znanium.com>



5. Электронно-библиотечная система издательства "Лань" <https://e.lanbook.com/>
6. Образовательная платформа "ЮРАЙТ" <https://urait.ru/>
7. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>
8. Информационно-образовательный портал Финуниверситета: <https://org.fa.ru>
9. Платформа «Открытое образование» <https://openedu.ru/>



10. Методические указания для обучающихся по освоению дисциплины

Студентам необходимо ознакомиться содержанием рабочей программы дисциплины (далее – РПД), с целями и задачами дисциплины, ее связями с другими дисциплинами образовательной программы, списком рекомендованной литературы и перечнем ресурсов сети «Интернет», необходимыми для освоения дисциплины. Часть разделов дисциплины студенты могут освоить с помощью бесплатных онлайн-курсов (с получением сертификата или иного документа об окончании), указанных в перечне ресурсов. Лекции, задания для семинарских и самостоятельных занятий, презентации и другие материалы по дисциплине, представленные в базовом ЭУК по дисциплине на образовательном портале Финансового университета, студенты изучают в процессе освоения курса в течение семестра. На сайте в разделе кафедры студенты могут ознакомиться в течение семестра с графиком консультаций преподавателей.

Рекомендации по подготовке к лекционным занятиям:

(теоретический курс)

Изучение дисциплины требует систематического и последовательного накопления знаний, следовательно, пропуски отдельных тем не позволяют глубоко освоить предмет. Именно поэтому контроль над систематической работой студентов всегда находится в центре внимания кафедры.

Лекционные занятия проводятся в соответствии с тематическим планом, при изложении материала рекомендуется использовать презентации в среде PowerPoint и фрагменты печатных материалов по теме лекции.

Студентам рекомендуется:

- перед каждой лекцией просматривать рабочую программу дисциплины, что позволит сэкономить время на записывание темы лекции, ее основных вопросов, рекомендуемой литературы;
- на отдельные лекции приносить соответствующий материал на бумажных или электронных носителях, представленный лектором на портале. Данный материал будет охарактеризован, прокомментирован, дополнен непосредственно на лекции;
- перед очередной лекцией необходимо просмотреть по конспекту материал предыдущей лекции. При затруднениях в восприятии материала следует обратиться к основным литературным источникам, если разобраться в материале опять не удалось, то обратитесь к лектору (по графику его консультаций) или к преподавателю на практических занятиях. Не оставляйте «белых пятен» в освоении материала.

Рекомендации по подготовке к практическим (семинарским) занятиям:

Проведение практических занятий осуществляется в компьютерных классах.

Студентам следует:



- приносить с собой на бумажных или электронных носителях рекомендованную преподавателем литературу к конкретному занятию;
- до очередного практического занятия по рекомендованным литературным источникам проработать теоретический материал, соответствующей темы занятия;
- при подготовке к практическим занятиям следует обязательно использовать не только лекции, но и другую учебную литературу;
- в начале занятий задать преподавателю вопросы по материалу, вызвавшему затруднения в его понимании и освоении, при решении задач, заданных для самостоятельного решения;
- в ходе семинара давать конкретные, четкие ответы по существу вопросов;
- на занятии доводить каждую задачу до окончательного решения, демонстрировать понимание проведенных расчетов (анализов, ситуаций), в случае затруднений обращаться к преподавателю.

Методические рекомендации по выполнению различных форм самостоятельных домашних заданий:

Самостоятельная работа студентов включает в себя выполнение различного рода заданий, которые ориентированы на более глубокое усвоение материала изучаемой дисциплины. К выполнению заданий для самостоятельной работы предъявляются следующие требования: задания должны выполняться самостоятельно и представляться в установленный срок, а также соответствовать установленным требованиям по оформлению.

Студентам следует:

- руководствоваться графиком самостоятельной работы, определенным РПД;
- выполнять все плановые задания, выдаваемые преподавателем для самостоятельного выполнения, и разбирать на семинарах и консультациях неясные вопросы;
- использовать при подготовке нормативные документы Финансового университета;
- при подготовке к экзамену параллельно прорабатывать соответствующие теоретические и практические разделы дисциплины, фиксируя неясные моменты для их обсуждения на плановой консультации.

При подготовке контрольной работы следует использовать нормативные документы Финансового университета, Методические рекомендации по планированию и организации внеаудиторной самостоятельной работы студентов по образовательным программам бакалавриата и магистратуры в Финансовом университете, утвержденные приказом Финуниверситета от 11.05.2021 г. № 1040/о (см. сайт Финансового Университета: на главной странице раздел "Наш



университет" → "Единая правовая база Финуниверситета" → "Организация учебного процесса" → "Нормативные документы по самостоятельной работе").

Методические рекомендации по работе с литературой:

Любая форма самостоятельной работы студента (подготовка к семинарскому занятию, выполнение расчетно-аналитической работы, начинается с изучения соответствующей литературы как в библиотеке, так и дома.

К каждой теме учебной дисциплины подобрана основная и дополнительная литература.

При работе с литературой рекомендуется делать записи. Записи в той или иной форме не только способствуют пониманию и усвоению изучаемого материала, но и помогают вырабатывать навыки явного изложения в письменной форме тех или иных теоретических вопросов.

Методические указания по проведению практических занятий:

По структуре практические занятия следует разделить на учебные и контрольные.

Учебные практические занятия структурно состоят из следующих компонент:

- проверка наличия выполненного задания самостоятельной работы каждого студента;
- выборочная проверка корректности выполнения домашнего задания;
- разбор типичных ошибок, возникших в самостоятельной работе;
- рассмотрение теоретических вопросов, связанных с текущим практическим занятием;
- разбор методов выполнения практических заданий и решения задач;
- корректировка заданий для самостоятельной работы студентов.

Контрольные практические занятия структурно состоят из следующих компонент:

- проведение аудиторных самостоятельных работ;
- подведение итогов и разбор типичных ошибок, возникших при выполнении самостоятельных работ.

Студенты должны обратить внимание на перечень основных контрольных мероприятий, которые проводятся в соответствии с рабочей программой дисциплины.

Конкретные сроки проведения этих мероприятий своевременно доводятся до сведения студентов.



11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем

Комплект лицензионного программного обеспечения:

1. Python 3.8
2. Jupyter Notebook (Windows 10)
3. PyTorch
4. Python 3.x Язык программирования для скриптов автоматизации, работы с MkDocs
5. Текстовый редактор
6. Операционная система
7. Антивирус Kaspersky

Современные профессиональные базы данных и информационные справочные системы

1. Информационно-правовая система «Гарант»
2. Информационно-правовая система «Консультант Плюс»
3. Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>
4. Язык программирования Python 3. <https://pythonworld.ru/>
5. Размещение ЭУК: <https://www.campus.fa.ru/>
6. Создание тестов, система Moodle, осуществление прокторинга (асинхронный, синхронный): <https://edu.fa.ru/>

Сертифицированные программные и аппаратные средства защиты информации

1. не предусмотрены

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

1. Учебная аудитория для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенная оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), набор демонстрационного оборудования (проектор, экран)
2. Помещение для самостоятельной работы обучающихся оснащенное компьютерной техникой с возможностью подключения к сети "Интернет" и



обеспечением доступа к электронной информационно-образовательной среде Университета.

3. **Компьютерный класс** для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенный оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), персональные компьютеры, набор демонстрационного оборудования (проектор, экран)
4. Библиотека, читальный зал