

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

Департамент анализа данных и машинного обучения
Факультета информационных технологий и анализа больших данных

УТВЕРЖДАЮ

Проректор по учебной
и методической работе

_____ Е.А. Каменева

25.04.2023 г.

Маковейчук К. А.

Технологии параллельного программирования

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки
01.03.02 - Прикладная математика и информатика,
ОП «Прикладная математика и информатика»,
ОП «Прикладное машинное обучение»

*Рекомендовано Ученым советом
Факультета информационных технологий и анализа больших данных
(протокол №31 от 18.04.2023 г.)*

*Одобрено Советом учебно-научного
Департамента анализа данных и машинного обучения
(протокол №2 от 29.03.2023 г.)*

Москва 2023

СОДЕРЖАНИЕ

1. Наименование дисциплины	2
2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	2
3. Место дисциплины в структуре образовательных программ.....	5
4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	5
5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий.....	6
5.1. Содержание дисциплины.....	6
5.2. Учебно–тематический план.....	7
5.3. Содержание семинаров, практических занятий.....	8
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.....	9
6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	9
6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю	10
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.....	12
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	23
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.....	25
10. Методические указания для обучающихся по освоению дисциплины	26
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	29
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.....	30

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

Таблица 1

ОП «Прикладная математика и информатика»

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
ОПК-2	Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач	1. Системно подходит к выбору математических методов и систем программирования для решения прикладных задач	<u>Знать</u> математические методы и системы программирования для решения прикладных задач с учетом возможности использования параллельных вычислений, системный подход и принципы выбора алгоритмов решения прикладных задач на их основе с применением технологий параллельного программирования и распределенных вычислений; <u>Уметь</u> делать выбор на основе системного подхода математических методов и систем программирования для решения прикладных задач с учетом возможности использования параллельных вычислений, применять принципы выбора алгоритмов решения прикладных задач на их основе с использованием технологий параллельного программирования и распределенных вычислений
		2. Разрабатывает алгоритмы решения прикладных задач с использованием математических методов	<u>Знать</u> современные подходы для разработки алгоритмов решения прикладных задач с использованием математических методов и параллельного программирования <u>Уметь</u> применять современные подходы для разработки алгоритмов решения прикладных задач с использованием математических методов и параллельного программирования

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
		3. Реализует алгоритмы с использованием современных систем программирования	<u>Знать</u> теоретические основы применения технологий параллельного программирования при реализации алгоритмов с использованием современных систем программирования <u>Уметь</u> применять технологии параллельного программирования при реализации алгоритмов с использованием современных систем программирования
ПКП-4	Способность применять математический аппарат при разработке вычислительных алгоритмов для решения задач в области экономики и финансов	1. Владеет математическим аппаратом, необходимым для разработки вычислительных алгоритмов	<u>Знать</u> теоретические основы применения математического аппарата, необходимого для разработки вычислительных алгоритмов, с использованием параллельных вычислений для решения задач в области экономики и финансов <u>Уметь</u> применять математический аппарат, необходимый для разработки вычислительных алгоритмов, с использованием параллельных вычислений для решения задач в области экономики и финансов
		2. Разрабатывает вычислительные алгоритмы для решения задач в области экономики и финансов	<u>Знать</u> принципы разработки вычислительных алгоритмов для решения задач в области экономики и финансов с учетом параллельных вычислений для решения задач в области экономики и финансов <u>Уметь</u> применять принципы разработки вычислительных алгоритмов для решения задач в области экономики и финансов с учетом параллельных вычислений для решения задач в области экономики и финансов, для таких, как OpenMP для систем с общей памятью, MPI для систем с распределенной памятью, PyOpenCL, PyOpenCuda для распараллеливания процессов на GPU

ОП «Прикладное машинное обучение»

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
ПКП-6	Способность выполнять сборку модулей и компонент программной реализации моделей машинного обучения и развертывания реализации моделей машинного обучения	1. Демонстрирует знание архитектуры современных инструментальных средств реализации моделей машинного обучения и подходов к их интеграции	<u>Знать</u> архитектуру современных инструментальных средств реализации моделей машинного обучения с использованием приемов параллельного программирования, и подходов к их интеграции <u>Уметь</u> применять знание архитектуры современных инструментальных средств реализации моделей машинного обучения с использованием приемов параллельного программирования, и подходов к их интеграции
		2. Владеет практическим навыком интеграции компонент моделей машинного обучения.	<u>Знать</u> принципы практической интеграции компонент моделей машинного обучения с учетом параллельных вычислений <u>Уметь</u> применять принципы практической интеграции компонент моделей машинного обучения с учетом параллельных вычислений
		3. Владеет практическим навыком сборки и развертывания многокомпонентных моделей машинного обучения.	<u>Знать</u> принципы сборки и развертывания многокомпонентных моделей машинного обучения с учетом стандартов распараллеливания программ <u>Уметь</u> применять принципы сборки и развертывания многокомпонентных моделей машинного обучения с учетом стандартов распараллеливания программ, таких, как OpenMP для систем с общей памятью, MPI для систем с распределенной памятью, PyOpenCL, PyOpenCuda для распараллеливания процессов на GPU

3. Место дисциплины в структуре образовательных программ

Дисциплина «Технологии параллельного программирования» относится к Циклу профиля (элективный) по направлению подготовки 01.03.02 - Прикладная математика и информатика», ОП «Прикладная математика и информатика», ОП «Прикладное машинное обучение».

Изучение дисциплины «Технологии параллельного программирования» базируется на знаниях, полученных при изучении дисциплин «Прикладные задачи машинного обучения», «Численные методы», «Алгоритмы и структуры данных в языке Python», «Дискретная математика».

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

ОП «Прикладная математика и информатика» /

ОП «Прикладное машинное обучение»

Таблица 3

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 7 / 6 (в часах)
Общая трудоёмкость дисциплины	3/108	108
<i>Контактная работа - Аудиторные занятия</i>	<i>34</i>	<i>34</i>
<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	
Вид промежуточной аттестации	зачет	

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

5.1. Содержание дисциплины

Раздел 1. Введение в «Технологии параллельного программирования».

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

Раздел 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.

Раздел 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.

Раздел 4. Основы векторизации исполняемого кода в OpenMP.

Проблемы векторизации. Способы векторизации. Директивы для работы с параллелизмом по данным (векторизация) в OpenMP 4.0.

Раздел 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.

Раздел 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

Таблица 4

№ п/ п	Наименование тем (разделов) дисциплины	Трудоёмкость в часах					Формы текущего контроля успеваемости
		Всего	Контактная работа - Аудиторная работа			Самостоя тельная работа	
			Общая, в т.ч.:	Лекции	Семинары, практическ ие занятия		
1.	Введение в «Технологии параллельно- го программи- рования»	14	4	2	2	10	Самостоятельные ра- боты. Участие в реше- нии задач на практиче- ских занятиях. Обсуж- дение решенных задач.
2.	Основы OpenMP	20	6	2	4	14	Самостоятельные ра- боты. Участие в реше- нии задач на практиче- ских занятиях. Обсуж- дение решенных задач.
3.	Директивы распреде- ления и синхро- низации рабо- ты OpenMP	16	4	2	2	12	Самостоятельные ра- боты. Участие в реше- нии задач на практиче- ских занятиях. Обсуж- дение решенных задач.
4.	Основы век- торизации ис- полняемого кода в OpenMP	16	4	2	2	12	Самостоятельные ра- боты. Участие в реше- нии задач на практиче- ских занятиях. Обсуж- дение решенных задач.

5.	Технологии параллельного программирования для графических процессоров (GPU)	22	8	4	4	14	Самостоятельные работы. Участие в решении задач на практических занятиях. Обсуждение решенных задач.
6.	Основы технологии MPI	20	8	4	4	12	Самостоятельные работы. Участие в решении задач на практических занятиях. Обсуждение решенных задач.
	В целом по дисциплине	108	34	16	18	74	Согласно учебному плану: контрольная работа
	Итого в %		31	47	53	69	

5.3. Содержание семинаров, практических занятий

Таблица 5

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарских, практических занятиях, рекомендуемые источники из разделов 8,9 (указывается раздел и порядковый номер источника)	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. <i>Рекомендуемые источники: основная 8.(1-4), дополнительная 8.(5-7)</i>	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. <i>Рекомендуемые источники: основная 8.(1-4), дополнительная 8.(5-7)</i>	Практикум по решению задач по тематике занятия и коллективное обсуждение решений Выполнение тестовых заданий
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. <i>Рекомендуемые источники: основная 8.(1-4), дополнительная 8.(5-7)</i>	Практикум по решению задач по тематике занятия и коллективное обсуждение решений Выполнение тестовых заданий

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарских, практических занятиях, рекомендуемые источники из разделов 8,9 (указывается раздел и порядковый номер источника)	Формы проведения занятий
Основы векторизации исполняемого кода в OpenMP	Директивы для работы с параллелизмом по данным (векторизация) в OpenMP 4.0. <i>Рекомендуемые источники: основная 8.(1-4), дополнительная 8.(5-7)</i>	Практикум по решению задач по тематике занятия и коллективное обсуждение решений
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. <i>Рекомендуемые источники: основная 8.(1-4), дополнительная 8.(5-7)</i>	Практикум по решению задач по тематике занятия и коллективное обсуждение решений Выполнение тестовых заданий
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. <i>Рекомендуемые источники: основная 8.(1-4), дополнительная 8.(5-7)</i>	Практикум по решению задач по тематике занятия и коллективное обсуждение решений Выполнение тестовых и контрольных заданий

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

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

Таблица 6

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 OpenMP	Проблемы векторизации. Способы векторизации.	Работа с текстом лекции, разбор вопросов и заданий по теме занятия; изучение рекомендованных к занятию 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>

Пример реализации:

```
#pragma omp parallel num_threads(4)
{
    printf("Перед директивой single без nowait \n");
    #pragma omp single
    {
        printf("В директиве single \n");
    }
    printf("После директивы single без nowait. Это сообщение никогда не будет раньше предыдущих \n");
    #pragma omp barrier
    printf("Перед директивой single с nowait \n");
    #pragma omp single nowait
    {
        printf("В директиве single \n");
    }
    printf("После директивы single с no-wait. Это сообщение может быть раньше предыдущих \n");
}
```

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).

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

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

Перечень компетенций, формируемых в процессе освоения дисциплины, 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

ОП «Прикладная математика и информатика»

Таблица 7

Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
ОПК-2. Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач	1. Системно подходит к выбору математических методов и систем программирования для решения прикладных задач	<u>Знать</u> математические методы и системы программирования для решения прикладных задач с учетом возможности использования параллельных вычислений, системный подход и принципы выбора алгоритмов решения прикладных задач на их основе с применением технологий параллельного программирования и распределенных вычислений; <u>Уметь</u> делать выбор на основе системного подхода математических методов и систем программирования для решения прикладных задач с учетом возможности использования параллельных вычислений, применять принципы выбора алгоритмов решения прикладных задач на их основе с использованием технологий параллельного программирования и распределенных вычислений	1. Выполнить обследование задач финансового предприятия «Зета» в соответствии принципами организации информационных систем. Оценить задачи с точки зрения применимости алгоритмов параллельных вычислений.

Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
	2. Разрабатывает алгоритмы решения прикладных задач с использованием математических методов	Знать современные подходы для разработки алгоритмов решения прикладных задач с использованием математических методов и параллельного программирования Уметь применять современные подходы для разработки алгоритмов решения прикладных задач с использованием математических методов и параллельного программирования	2. Реализовать постановку задачи, алгоритм и программу решения задачи обработки условных финансовых данных с помощью итерационного метода решения системы линейных алгебраических уравнений (СЛАУ) как умножение матрицы на вектор с использованием технологий параллельного программирования и распределенных вычислений (возможные технологии для реализации программы: C++ & OpenMP Python + PyOpenCL Python + mpi4py)
	3. Реализует алгоритмы с использованием современных систем программирования	Знать теоретические основы применения технологий параллельного программирования при реализации алгоритмов с использованием современных систем программирования Уметь применять технологии параллельного программирования при реализации алгоритмов с использованием современных систем программирования	Определить инструментальные средства информационных технологий для реализации программы построения фрактала Ньютона для уравнения $x^3-1=0$ Обосновать применение технологий параллельного программирования при разработке программы. Реализовать алгоритм и программу для решения задачи. Применить в решении задачи технологии параллельного программирования.

Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
ПКП-4. Способность применять математический аппарат при разработке вычислительных алгоритмов для решения задач в области экономики и финансов	1. Владеет математическим аппаратом, необходимым для разработки вычислительных алгоритмов	<u>Знать</u> теоретические основы применения математического аппарата, необходимого для разработки вычислительных алгоритмов, с использованием параллельных вычислений для решения задач в области экономики и финансов <u>Уметь</u> применять математический аппарат, необходимый для разработки вычислительных алгоритмов, с использованием параллельных вычислений для решения задач в области экономики и финансов	Реализовать структуру данных и алгоритм умножения векторов в Python, с использованием GPU (библиотека PyOpenCL или PyOpenCuda), либо с использованием библиотеки mpi4py. Сравнить два метода параллельного программирования.
	2. Разрабатывает вычислительные алгоритмы для решения задач в области экономики и финансов	<u>Знать</u> принципы разработки вычислительных алгоритмов для решения задач в области экономики и финансов с учетом параллельных вычислений для решения задач в области экономики и финансов <u>Уметь</u> применять принципы разработки вычислительных алгоритмов для решения задач в области экономики и финансов с учетом параллельных вычислений для решения задач в области экономики и финансов, для таких, как OpenMP для систем с общей памятью, MPI для систем с распределенной памятью, PyOpenCL, PyOpenCuda для распараллеливания процессов на GPU	Разработать приложение для расчета прогноза погоды на основании 5 матриц исходных данных для точек атмосферы на 4 высотах, с применением технологий параллельного программирования и распределенных вычислений. Прогноз условно выполняется на основании решения уравнения $5x^5 + 3x^4 - 7x^3 + 2x^2 - 8x + 15 = y$, где x – данные матриц по каждой высоте. Выполнить разработку с применением параллельных вычислений и выбранной технологии, такой, как OpenMP, MPI, PyOpenCL или PyOpenCuda для распараллеливания процессов на GPU

ОП «Прикладное машинное обучение»

Таблица 8

Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
ПКП-6. Способность выполнять сборку модулей и компонент программной реализации моделей машинного обучения и развертывания реализации моделей машинного обучения	1. Демонстрирует знание архитектуры современных инструментальных средств реализации моделей машинного обучения и подходов к их интеграции	Знать архитектуру современных инструментальных средств реализации моделей машинного обучения с использованием приемов параллельного программирования, и подходов к их интеграции Уметь применять знание архитектуры современных инструментальных средств реализации моделей машинного обучения с использованием приемов параллельного программирования, и подходов к их интеграции	Реализовать структуру данных и алгоритм умножения векторов в Python, с использованием GPU (библиотека PyOpenCL или PyOpenCuda), либо с использованием библиотеки mpi4py. Сравнить два метода параллельного программирования.
	2. Владеет практическим навыком интеграции компонент моделей машинного обучения.	Знать принципы практической интеграции компонент моделей машинного обучения с учетом параллельных вычислений Уметь применять принципы практической интеграции компонент моделей машинного обучения с учетом параллельных вычислений	Определить спецификацию функциональных и нефункциональных требований к приложению для расчета прогноза погоды на основании 5 матриц исходных данных для точек атмосферы на 4 высотах, с применением технологий параллельного программирования и распределенных вычислений.
	3. Владеет практическим навыком сборки и развертывания многокомпонентных моделей машинного обучения.	Знать принципы сборки и развертывания многокомпонентных моделей машинного обучения с учетом стандартов распараллеливания программ Уметь применять принципы сборки и развертывания многокомпонентных моделей машинного обучения с учетом стандартов распараллеливания программ, таких, как OpenMP для систем с общей памятью, MPI для систем с распределенной памятью, PyOpenCL, PyOpenCuda для распараллеливания процессов на GPU	Разработать приложение для расчета прогноза погоды на основании 5 матриц исходных данных для точек атмосферы на 4 высотах, с применением технологий параллельного программирования и распределенных вычислений. Прогноз условно выполняется на основании решения уравнения $5x^5 + 3x^4 - 7x^3 + 2x^2 - 8x + 15 = y$, где x – данные матриц по каждой высоте. Выполнить разработку с применением параллельных вычислений и выбранной технологии, такой, как

Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
			OpenMP, MPI, PyOpenCL или PyOpenCuda для распараллеливания процессов на GPU

Примеры тестовых заданий

1. Какие подходы, применяемые в архитектурах компьютеров, позволяли повысить их производительность

Конвейерный способ обработки

Независимые функциональные устройства

Векторные операции верно

Уменьшение энергопотребления

2. Какой принцип организации памяти используется для обеспечения требуемой емкости и высокого быстродействия памяти компьютера за приемлемую цену

Интенсивный

Экстенсивный

Масштабируемости

Иерархический

3. Самый минимальный по объему тип памяти, но и самый быстрый это

Регистры центрального процессора

Кэш первого уровня

Оперативная память

Кэш нулевого уровня

4. Для эффективного использования иерархической структуры памяти при написании программ необходимо использовать

Принцип локальности по обращению

Принципы объектного-ориентированного программирования

Принцип случайного обращения

Принцип маленьких данных

5. Классификация Флинна основывается на

на понятии потока

на понятии декомпозиции данных

на особенностях процессоров

на особенностях организации памяти

6. SISD – это

одиночный поток команд и одиночный поток данных

множественный поток команд и одиночный поток данных

одиночный поток команд и множественный поток данных

множественный поток команд и множественный поток данных

7. Какие два крупных класса параллельных высокопроизводительных вычислительных систем можно выделить?

вычислительные системы с общей памятью

вычислительные системы с распределенной памятью

вычислительные системы с очень большим объемом оперативной памяти

вычислительные системы с водяным охлаждением

8. Какие из этих технологий позволяют создавать параллельные программы

OpenMP

MPI

Pascal

Parallel

9. Многопоточность это -

свойство платформы или приложения, состоящее в том, что процесс, порождённый в операционной системе, может состоять из нескольких потоков, выполняющихся «параллельно», то есть без предписанного порядка во времени.

свойство центрального процессора автоматически выполнять программы сразу в несколько потоков

свойство компилятора позволяющие создавать параллельные многопоточковые программы.

10. Технология параллельного программирования OpenMP включается в себя

директивы компилятора

библиотечные функции

переменные среды окружения

параллельный компилятор

11. Для создания параллельных программ в OpenMP применяется модель программирования

Разветвление-слияние (Fork–join)

Декомпозиция по данным

Главный подчиненный (Master-slave)

12. Выберите код в котором создается параллельная область с 5 дополнительными потоками выполнения

```
#pragma omp parallel shared(a,b,c) num_threads(4)
    #pragma omp for
    for (int i = 0; i < n; i++)
    {
        c[i] = a[i] + b[i];
    }

#pragma omp parallel shared(a,b,c) num_threads(6)
    #pragma omp for
    for (int i = 0; i < n; i++)
    {
        c[i] = a[i] + b[i];
    }
```

```

#pragma omp parallel shared(a,b,c) num_threads(5)
    #pragma omp for
    for (int i = 0; i < n; i++)
    {
        c[i] = a[i] + b[i];
    }

```

13. Следующий фрагмент кода запустили на компьютере с 8 ядерным процессором.

Сколько раз будет выведено слово Print?

```

#include "stdio.h"
#include "omp.h"

int main()
{
    printf("Print\n");
    #pragma omp parallel num_threads(10)
    {
        printf("Print\n");
    }
    #pragma omp parallel num_threads(2)
    {
        printf("Print\n");
    }
    printf("Print\n");
}

```

14. Какое значение переменной n будет выведено на экран в следующей программе при запуске на компьютере с 4-х ядерным процессором?

```

#include "stdio.h"
#include "omp.h"

int main()
{
    printf("Print\n");
    int n = 3;
    #pragma omp parallel num_threads(40)
    {
        n += 1;
    }
    printf("n=%d", n);
}

```

Значение не определено, от запуска к запуску может меняться

Программа не запустится, так как потоков больше чем ядер

Значение 43 будет выведено

Программа не скомпилируется из-за синтаксической ошибки

15. Выберите два основных класса переменных в OpenMP

общие

локальные

глобальные

независимые

16. Если несколько потоков одновременно записывают значение общей переменной без выполнения синхронизации, то

возникает ситуации «гонки данных» (data race)

возникает ситуации не разделенного доступа

в данном случае нет никаких особенных ситуаций

17. В результате выполнения данной программы на компьютере с восьмиядерным процессором слов «Print» будет выведено (ввести числом)

```
#include <stdio.h>
#include <omp.h>

int main()
{
    int n=1;
    omp_set_nested(1);
    omp_set_num_threads(2);
    #pragma omp parallel if (n>10)
    {
        printf("Print ");
    }
}
```

18. В результате выполнения следующей программы на компьютере с 4 ядерным процессором на экран будут выведены числа. Впишите каждое из чисел в отдельное окно ввода

```

#include <stdio.h>
#include <omp.h>

int main()
{
    int n=0;
    omp_set_nested(1);
    #pragma omp parallel num_threads(2) reduction(+:n)
    {
        n = 0;
        #pragma omp parallel num_threads(2) reduction(+:n)
        {
            n = 1;
        }
    }
    printf("%d ", n);
    n = 0;
    omp_set_nested(0);
    #pragma omp parallel num_threads(2) reduction(+:n)
    {
        n = 0;
        #pragma omp parallel num_threads(2) reduction(+:n)
        {
            n = 1;
        }
    }
    printf("%d ", n);
}

```

19. Какое максимальное ускорение (ускорение = время на 1 ядре / время на n ядрах) возможно в следующей программе при запуске на компьютере с 8 ядерным процессором

```

void sub_a(float **a, float **b, float **c, float **d, int n, int m)
{
    int i, j;
    #pragma omp parallel shared(a,b,c,d,n,m) private(i,j)
    {
        #pragma omp sections nowait
        {
            #pragma omp section
            for (i = 1; i < n; i++)
            {
                for (j = 0; j <= i; j++)
                    b[j][i] = (a[j][i] + a[j][i - 1]) / 2.;
            }
            #pragma omp section
            for (i = 1; i < m; i++)
            {
                for (j = 0; j <= i; j++)
                    d[j][i] = (c[j][i] + c[j][i - 1]) / 2.;
            }
        }
    }
}

```

Максимальное ускорение зависит от количества используемых ядер

Максимальное ускорение 8

Максимальное ускорение 2

20. Для выделения независимой задачи, которая может выполняться параллельно используется директива

`#pragma omp task`

`#pragma omp section`

`#pragma omp doit`

Примерные вопросы для подготовки к зачету

1. Классификация вычислительных систем Флинна. Примеры.
2. Закон Амдала. Пример расчета.
3. Технологии параллелизма. Классификация, примеры.
4. Анализ задачи умножения матриц.
5. Директивы параллельного программирования OpenMP.
6. Модели памяти и классы переменных в OpenMP.
7. Вложенный параллелизм в OpenMP.
8. Директивы распределения и синхронизации работы OpenMP
9. Распараллеливание выполнения циклов в OpenMP: директива и принцип организации.
10. Распределение работы на основе независимых задач в OpenMP: директива и принцип организации.
11. Синхронизация выполнения различных потоков.
12. Распределение операторов одного структурного блока между нитями.
13. Копирование значений приватных переменных.
14. Основы векторизации исполняемого кода в OpenMP
15. Директивы для работы с параллелизмом по данным (векторизация) в OpenMP 4.0.
16. Что такое процесс и поток в современных операционных системах.
17. Для чего необходима синхронизация в многопоточных программах.

18. В каких состояниях может находиться поток.
19. В результате каких действий поток может переходить из одного состояния в другое.
20. Технологии параллельного программирования для графических процессоров (GPU)
21. Архитектура графического процессора (GPU).
22. Возможные преимущества вычислений на графическом процессоре.
23. Средства разработки для графического процессора.
24. Python в работе с GPU: реализации библиотек, сравнение.
25. Основы неоднородных (гетерогенных) вычислений.
26. Уровни параллельности в архитектуре GPU.
27. Гетерогенное программирование на PyOpenCL и PyCUDA.
28. Построение приложений при помощи PyOpenCL и PyCUDA.
29. Основы технологии MPI
30. Основы MPI для Python.
31. Операции типа точка-точка в MPI.
32. Примитивы синхронизации между процессами в MPI.
33. Операции коллективного взаимодействия в MPI.
34. Операции сочетания параллельных результатов вычислений в MPI.
35. Параллельный алгоритм умножения матрицы на вектор: реализации на MPI.

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

а) основная:

1. Малявко, А. А. Параллельное программирование на основе технологий openmp, mpi, cuda : учебное пособие для академического бакалавриата / А. А. Малявко. — 2-е изд., испр. и доп. — Москва : Юрайт, 2023. — 129 с. — ЭБС Юрайт. — URL: <https://biblio-online.ru/bcode/514199> (дата обращения: 23.05.2023). — Текст : электронный.

2. Арыков, С. Б. Параллельное программирование над общей памятью. OpenMP : учебное пособие / С. Б. Арыков, М. А. Городничев, Г. А. Щукин. — Новосибирск : НГТУ, 2019. — 95 с. — ЭБС Лань. — URL: <https://e.lanbook.com/book/152252> (дата обращения: 23.05.2023). — Текст : электронный.

3. Филатов, А. С. Параллельное программирование : учебное пособие / А. С. Филатов. — Москва : РТУ МИРЭА, 2021. — 46 с. — ЭБС Лань. — URL: <https://e.lanbook.com/book/218429> (дата обращения: 23.05.2023). — Текст : электронный.

4. Жуков, Р. А. Язык программирования Python: практикум : учебное пособие / Р. А. Жуков. — Москва : ИНФРА-М, 2022. — 216 с. — (Высшее образование: Бакалавриат). — ЭБС ZNANIUM.com. — URL: <https://znanium.com/catalog/product/1689648> (дата обращения: 23.05.2023). — Текст : электронный.

б) дополнительная:

5. Карепова, Е. Д. Основы многопоточного и параллельного программирования: учебное пособие / Е. Д. Карепова. — Красноярск : Сиб. Федер. ун-т, 2016. — 356 с. — ЭБС ZNANIUM.com. — URL: <http://znanium.com/catalog/product/966962> (дата обращения: 23.05.2023). — Текст : электронный.

6. Основы высокопроизводительных вычислений. Т. 2. Технологии параллельного программирования: учебное пособие / К. Е. Афанасьев, С. В. Стуколов, В. В. Малышенко [и др.]. — Кемерово : Кемеровский государственный университет, 2012. — 412 с. — ЭБС Университетская библиотека online. — URL: <http://biblioclub.ru/index.php?page=book&id=232204> (дата обращения: 23.05.2023). — Текст : электронный.

7. Филатов, А. С. Параллельное программирование : учебное пособие / А. С. Филатов. — Москва : РТУ МИРЭА, 2021. — 46 с. — ЭБС Лань. — URL: <https://e.lanbook.com/book/218429> (дата обращения: 23.05.2023). — Текст : электронный.

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

1. <http://www.fa.ru> – официальный сайт Финансового университета при Правительстве Российской Федерации
2. <http://www.library.fa.ru/> - Библиотечно-информационный комплекс Финуниверситета
3. <https://org.fa.ru> - личный кабинет обучающегося
4. <https://org.fa.ru/app/ebs/list> - реестр ЭБС Финансового университета
5. <https://openedu.ru/course/msu/PARPROG/> - онлайн-курс «Суперкомпьютеры и параллельная обработка данных» на национальной платформе «Открытое образование» (курс МГУ им. М. В, Ломоносова)
6. <https://mooc.tsu.ru/mooc-openedu/mpi/> - онлайн-курс «Параллельное программирование с использованием OpenMP и MPI» на платформе MOOK Томского государственного университета
7. <https://stepik.org/course/115024/promo?search=2147452613> - онлайн-курс «Введение в параллельное программирование (OpenMP и MPI)» на платформе Stepik (Томский государственный университет, альтернативная платформа)
8. <https://teach-in.ru/course/parallel-computing-lukyanenko/lecture> - онлайн-курс «Параллельные вычисления» на платформе «Лекторий Teach-in» (лекции ученых МГУ им. М. В, Ломоносова)
9. <https://campus.fa.ru/course/view.php?id=15769> – базовый ЭУК по дисциплине «Технологии параллельного программирования» Финансового университета
10. <http://onreader.mdl.ru/PythonParallelProgrammingCookbook.2nd/content/index.html> - Законе Джанкарло. Книга рецептов параллельного программирования Python. 2е изд. Copyright © 2019 Packt Publishing. Документ предоставляется по лицензии Creative Commons Attribution 3.0 License.
11. https://registry.khronos.org/OpenCL/specs/3.0-unified/html/OpenCL_API.html - The OpenCL™ Specification Khronos® OpenCL Working Group Version V3.0.14, Mon, 17 Apr 2023

12. <https://github.com/inducer> - Andreas Klöckner's web page PyOpenCL
13. <https://documen.tician.de/pyopencl/> - PyOpenCL's documentation
14. <https://pypi.org/project/mpi4py/> - Python bindings for MPI
15. <https://mpi4py.readthedocs.io/en/stable/index.html> - MPI for Python

10. Методические указания для обучающихся по освоению дисциплины

Цель методических рекомендаций – обеспечить студенту бакалавриата (далее – студенту) оптимальную организацию процесса изучения дисциплины, а также выполнения различных форм самостоятельной работы.

Методические рекомендации по изучению 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 департамента 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 департамента.

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.

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

11.1. Комплект лицензионного программного обеспечения:

Пакет офисных программ (Astra Linux, Library Office и пр.);
Антивирус Kaspersky.

11.2. Современные профессиональные базы данных и информационные справочные системы:

Информационно-правовая система «Консультант Плюс»;

Информационно-правовая система «Гарант»;

Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>;

Система комплексного раскрытия информации «СКРИН»: <https://skrin.ru>.

11.3. Сертифицированные программные и аппаратные средства защиты информации - не предусмотрены.

11.4. Язык программирования Python 3.x в среде Windows, библиотеки PyOpenCl, mpi4py.

11.5. Платформа для научных исследований, основанная на языке программирования Python, Anaconda.

11.6. Интерфейс передачи сообщений MS MPI <https://www.microsoft.com/en-us/download/details.aspx?id=100593>.

11.7. Компилятор g++ <https://www.msys2.org> .

11.8. Открытый стандарт для распараллеливания программ на языках Си, Си++ и Фортран OpenMP, реализация для компилятора g++ https://packages.msys2.org/package/mingw-w64-x86_64-openmp?repo=mingw64 .

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

Материально-техническая база Финансового университета, необходимая для осуществления образовательного процесса по данной дисциплине, в соответствии с требованиями ФОС ВО включает в себя специальные помещения для проведения лекций, семинарских занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации; помещения укомплектованы специализированной мебелью и техническими средствами обучения, необходимыми для представления информации большой аудитории.

Помещения для самостоятельной работы студентов включают в себя библиотеку с читальным залом, укомплектованную в соответствии с существующими

нормами необходимой учебной и учебно-методической литературой и иными материалами; медиатеку с выходом в Интернет, компьютерные классы с возможностью работы в Интернет; аудитории для консультационной деятельности.