

МИНПРОСВЕЩЕНИЯ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
"Тульский государственный педагогический университет им. Л.Н. Толстого"
(ФГБОУ ВО "ТГПУ им. Л.Н. Толстого")

Параллельное программирование

рабочая программа дисциплины (модуля)

Закреплена за кафедрой	институт передовых информационных технологий
ОПОП	Направление 09.03.01 Информатика и вычислительная техника направленность (профиль) Информационные системы и управление бизнес-процессами
Квалификация	Бакалавр
Год начала подготовки	2022
Форма обучения	очная
Общая трудоемкость	4 з.е.

Виды контроля по семестрам:
экзамен 7

Семестр(Курс.Номер семестра на курсе)	7(4.1)		Итого	
	УП	РПД	УП	РПД
Лекции	16	16	16	16
Лабораторные	30	30	30	30
Итого ауд.	46	46	46	46
КСР	2	2	2	2
Контактная работа	48	48	48	48
Сам. работа	60	60	60	60
Часы на контроль	36	36	36	36
Практическая подготовка	0	0	0	0
Семинары	0	0	0	0
Консультации	0	0	0	0
Итого трудоемкость в часах	144	144	144	144

Программу составил(и):

нет, ст. преподаватель, Гладких Илья Юрьевич

Рабочая программа дисциплины

Параллельное программирование

разработана в соответствии с ФГОС:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 929)

составлена на основании учебного плана:

Направление 09.03.01 Информатика и вычислительная техника
направленность (профиль) Информационные системы и управление бизнес-процессами
утвержденного Учёным советом вуза от 28.02.2022 протокол № 3.

РПД утверждена Учёным советом университета
протокол от 28.2.2022 г. № 3

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель преподавания дисциплины - изучение и практическое освоение общих принципов и современных методов параллельного программирования, получение навыков использования конкретных инструментов параллельного программирования.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Цикл (раздел) ООП:	Б1.В
2.1	Требования к предварительной подготовке обучающегося:
1.	Функциональный подход к программированию. Программирование на языке C++ и C#: базовый синтаксис, работа с массивами, пользовательские функции, указатели и адресное пространство.
2.	Практикум по веб-программированию
3.	Веб-программирование
4.	Интеллектуальный анализ данных и методы поддержки принятия решений
5.	Практикум по программированию мобильных приложений
6.	Тестирование программного обеспечения информационных систем
7.	Объектно-ориентированное программирование
8.	технологическая (проектно-технологическая) практика
9.	Алгоритмы и структуры данных
10.	Операционные системы
2.2	Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:
1.	Архитектура клиент-серверных приложений
2.	Проектирование клиент-серверных приложений
3.	Машинное обучение
4.	Искусственный интеллект и логическое программирование
5.	преддипломная практика

3. СООТНЕСЕНИЕ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ) С ИНДИКАТОРАМИ ДОСТИЖЕНИЯ КОМПЕТЕНЦИЙ

3.1 Компетенции обучающегося и индикаторы их достижения:

ОПК-8: Способен разрабатывать алгоритмы и программы, пригодные для практического применения

ОПК-8.1	Знает основные понятия алгоритмизации, языки программирования, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий
	критерии оценки эффективности параллельных программ и их ограничения
ОПК-8.2	Умеет разрабатывать алгоритмы и программы, пригодные для практического применения и автоматизации бизнес-процессов
	конструирования высокопроизводительных вычислительных систем
ОПК-8.3	Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач
	принципы и методы получения распараллеленного решения
ПК-3: Способность проводить тестирование компонентов программного обеспечения информационных систем	
ПК-3.1	Обладает базовыми знаниями в области тестирования информационных систем и управление процессом тестирования программных средств
	критерии оценки эффективности параллельных программ и их ограничения
ПК-3.2	Умеет применять в практической деятельности методы тестирования
	конструирования высокопроизводительных вычислительных систем
ПК-3.3	Имеет практический навык проведения тестирования компонентов программного обеспечения информационных систем
	принципы и методы получения распараллеленного решения

3.2 Результаты обучения по дисциплине:

В результате освоения дисциплины обучающийся должен:

	Знать:
3.1	критерии оценки эффективности параллельных программ и их ограничения
3.2	параллельные вычислительные методы
3.3	архитектурные принципы реализации параллельной обработки в вычислительных машинах

3.4	методы и языковые механизмы конструирования параллельных программ
3.5	принципы и методы получения распараллеленного решения
	Уметь:
У.1	разрабатывать параллельные программы с использованием библиотек MPI, OpenMP и технологии потоков
У.2	осуществлять перенос реализованных программных средств на параллельные платформы
У.3	проводить сравнительный анализ последовательных и параллельных программных средств в различных реализациях и оценивать их эффективность
	Владеть:
В.1	конструирования высокопроизводительных вычислительных систем
В.2	настройки высокопроизводительных вычислительных систем
В.3	оценки качества настройки высокопроизводительных вычислительных систем

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Литература	Содержание
	Введение в предмет				
1.1	Модели параллельного программирования. Схемы управления и эквивалентность вычислительных процессов /Лек/	7	2	Л1.1 Л1.2 Л1.3Л2.3 Л2.7 Л2.11	Формальные модели параллельного программирования. Параллельные схемы программ. Информационный базис и схема управления. Асинхронные вычислительные процессы над информационным базисом. Неуправляемые вычислительные процессы. Неуправляемые вычислительные процессы. Схема управления параллельной схемы программы. Управляющий автомат и особенности автоматных функций. Эквивалентность вычислительных процессов по произвольному отношению. Детерминизм и отношение "большей параллельности" схем управления.
1.2	Формальные модели параллельного программирования /Ср/	7	2	Л1.1 Л1.2 Л1.3Л2.3 Л2.7 Л2.11	Формальные модели параллельного программирования. Параллельные схемы программ. Информационный базис и схема управления. Асинхронные вычислительные процессы над информационным базисом. Неуправляемые вычислительные процессы.
1.3	Управление параллельными схемами программы /Ср/	7	2	Л1.1 Л1.2 Л1.3Л2.2 Л2.3 Л2.4 Л2.7 Л2.8 Л2.9 Л2.11	Неуправляемые вычислительные процессы. Схема управления параллельной схемы программы. Управляющий автомат и особенности автоматных функций. Эквивалентность вычислительных процессов по произвольному отношению. Детерминизм и отношение "большей параллельности" схем управления.
1.4	Создание параллельных приложений. Сравнение работы последовательных и параллельных решений. /Лаб/	7	2	Л1.1 Л1.2 Л1.3Л2.2 Л2.3 Л2.4 Л2.7 Л2.8 Л2.9 Л2.11	Параллельные схемы программ. Асинхронные вычислительные процессы над информационным базисом. Неуправляемые вычислительные процессы. Схема управления параллельной схемы программы. Управляющий автомат и особенности автоматных функций.
1.5	Создание параллельных приложений. Сравнение работы последовательных и параллельных решений. /Ср/	7	2	Л1.1 Л1.2 Л1.3Л2.2 Л2.3 Л2.4 Л2.7 Л2.8 Л2.9 Л2.11	Параллельные схемы программ. Асинхронные вычислительные процессы над информационным базисом. Неуправляемые вычислительные процессы. Схема управления параллельной схемы программы. Управляющий автомат и особенности автоматных функций.
	Вычислительные системы с массовым параллелизмом				

2.1	Моделирующие вычислительные системы и системы реального времени. Классификация вычислительных систем /Лек/	7	2	Л1.1 Л1.2 Л1.3Л2.2 Л2.3 Л2.5 Л2.7 Л2.9 Л2.10 Л2.11	Определение и особенности применения: моделирующие вычислительные системы и системы реального времени. Примеры. Характерные особенности таких систем. Классификация ВС по Флинну. Реконфигурируемые системы MIMD архитектуры. Статическая и динамическая реконфигурация. Оценка производительности массивно параллельных ВС. Причины снижения производительности.
2.2	Характеристики и особенности моделирующих вычислительных систем и систем реального времени /Ср/	7	2	Л1.1 Л1.2 Л1.3Л2.2 Л2.3 Л2.5 Л2.7 Л2.9 Л2.10 Л2.11	Определение и примеры, особенности применения: моделирующие вычислительные системы и системы реального времени. Характерные особенности таких систем.
2.3	Характеристики и особенности моделирующих вычислительных систем и систем реального времени /Лаб/	7	2	Л1.1 Л1.2 Л1.3Л2.2 Л2.3 Л2.5 Л2.7 Л2.9 Л2.10 Л2.11	Определение и примеры, особенности применения: моделирующие вычислительные системы и системы реального времени. Характерные особенности таких систем
2.4	Классификация ВС и оценка производительности массивно параллельных ВС /Лаб/	7	2	Л1.1 Л1.2 Л1.3Л2.3 Л2.7 Л2.11	Классификация ВС по Флинну. Реконфигурируемые системы MIMD архитектуры. Статическая и динамическая реконфигурация. Оценка производительности массивно параллельных ВС. Причины снижения производительности.
	Параллельное программирование C++ и C#				
3.1	Локальные и распределенные вычисления /Лек/	7	2	Л1.1 Л1.2 Л1.3Л2.3 Л2.7 Л2.11	Требования к распределенным системам. Взаимодействие в распределенных системах. Управление вычислительными заданиями. Модель распределенного вычисления.
3.2	Локальные и распределенные вычисления /Ср/	7	8	Л1.1 Л1.2 Л1.3Л2.3 Л2.7 Л2.11	Требования к распределенным системам. Взаимодействие в распределенных системах. Управление вычислительными заданиями. Модель распределенного вычисления.
3.3	Локальные и распределенные вычисления /Лаб/	7	4	Л1.1 Л1.2 Л1.3Л2.3 Л2.7 Л2.11	Требования к распределенным системам. Взаимодействие в распределенных системах. Управление вычислительными заданиями. Модель распределенного вычисления.
3.4	Структура библиотеки MPI /Лек/	7	2	Л1.1 Л1.2 Л1.3Л2.3 Л2.7 Л2.11	Описание библиотеки MPI: различные версии и модификации для разных языков программирования и платформ выполнения программ. Структура библиотеки MPI. Интеграция библиотеки MPI в код на языке C++ и в проекты. Общие процедуры MPI.
3.5	Структура библиотеки MPI /Ср/	7	8	Л1.1 Л1.2 Л1.3Л2.3 Л2.7 Л2.11	Описание библиотеки MPI: различные версии и модификации для разных языков программирования и платформ выполнения программ. Структура библиотеки MPI. Интеграция библиотеки MPI в код на языке C++ и в проекты. Общие процедуры MPI.
3.6	Структура библиотеки MPI /Ср/	7	2	Л1.1 Л1.2 Л1.3Л2.3 Л2.7 Л2.11	Описание библиотеки MPI: различные версии и модификации для разных языков программирования и платформ выполнения программ. Структура библиотеки MPI. Интеграция библиотеки MPI в код на языке C++ и в проекты. Общие процедуры MPI.
3.7	Структура библиотеки OpenMP /Лек/	7	2	Л1.1 Л1.2 Л1.3Л2.3 Л2.7 Л2.11	Описание библиотеки OpenMP. Структура библиотеки OpenMP. Интеграция библиотеки OpenMP в код на языке C++ и в проекты. Общие процедуры OpenMP.

3.8	Структура библиотеки OpenMP /Ср/	7	4	Л1.1 Л1.2 Л1.3Л2.3 Л2.7 Л2.11	Описание библиотеки OpenMP. Структура библиотеки OpenMP. Интеграция библиотеки OpenMP в код на языке C++ и в проекты. Общие процедуры OpenMP.
3.9	Структура библиотеки OpenMP /Лаб/	7	4	Л1.1 Л1.2 Л1.3Л2.3 Л2.7 Л2.11	Описание библиотеки OpenMP. Структура библиотеки OpenMP. Интеграция библиотеки OpenMP в код на языке C++ и в проекты. Общие процедуры OpenMP.
3.10	Организации доступа к данным и обмена сообщениями /Лек/	7	1	Л1.1 Л1.2 Л1.3Л2.3 Л2.7 Л2.11	Модели вычислений. Организация обмена сообщениями. Коллективные функции.
3.11	Организации доступа к данным и обмена сообщениями /Лаб/	7	4	Л1.1 Л1.2 Л1.3Л2.3 Л2.7 Л2.11	Модели вычислений. Организация обмена сообщениями. Коллективные функции.
3.12	Организации доступа к данным и обмена сообщениями /Ср/	7	4	Л1.1 Л1.2 Л1.3Л2.3 Л2.7 Л2.11	Модели вычислений. Организация обмена сообщениями. Коллективные функции.
3.13	Класс Thread /Лек/	7	1	Л1.4 Л1.1 Л1.2 Л1.3Л2.3 Л2.5 Л2.6 Л2.7 Л2.11	Описание класса Thread. Структура класса Thread. Интеграция класса Thread в код на языке C# и в проекты. Общие методы Thread.
3.14	Класс Thread /Лаб/	7	4	Л1.4 Л1.1 Л1.2 Л1.3Л2.3 Л2.5 Л2.6 Л2.7 Л2.11	Описание класса Thread. Структура класса Thread. Интеграция класса Thread в код на языке C# и в проекты. Общие методы Thread.
3.15	Класс Thread /Ср/	7	2	Л1.1 Л1.2 Л1.3Л2.3 Л2.5 Л2.6 Л2.7 Л2.11	Описание класса Thread. Структура класса Thread. Интеграция класса Thread в код на языке C# и в проекты. Общие методы Thread.
3.16	Работа над индивидуальным проектом /Ср/	7	8	Л1.1 Л1.2 Л1.3Л1.4 Л2.2 Л2.3 Л2.4 Л2.5 Л2.6 Л2.7 Л2.9 Л2.11	
	Реализация параллельных алгоритмов				
4.1	Параллельные алгоритмы. Этапы разработки параллельного алгоритма и рекурсии в параллельном программировании /Лек/	7	2	Л1.1 Л1.2 Л1.3Л1.4 Л2.2 Л2.3 Л2.4 Л2.7 Л2.9 Л2.11	Параллельные алгоритмы. Общие способы распараллеливания алгоритмов. Организация параллельного исполнения рекурсивных вычислений. Параллельные численные алгоритмы линейной алгебры. Декомпозиция, проектирование коммуникаций, укрупнение, планирование вычислений. Распараллеливание рекурсии: представление в виде простой и сложной рекурсии или в виде цикла
4.2	Этапы разработки параллельного алгоритма и рекурсии в параллельном программировании /Ср/	7	2	Л1.1 Л1.2 Л1.3Л1.4 Л2.2 Л2.3 Л2.4 Л2.7 Л2.9 Л2.11	Декомпозиция, проектирование коммуникаций, укрупнение, планирование вычислений. Распараллеливание рекурсии: представление в виде простой и сложной рекурсии или в виде цикла.
4.3	Вычисления над матричными данными в параллельном программировании. Параллелизация решения уравнений /Лек/	7	2	Л1.1 Л1.2 Л1.3Л1.4 Л2.2 Л2.3 Л2.4 Л2.7 Л2.9 Л2.11	Параллельные численные алгоритмы линейной алгебры. Способы разбиения матриц (горизонтальная, вертикальная, блочные схемы). Методы вычисления произведения матриц с использованием разных схем разбиения матриц. Обеспечение предельно допустимого параллелизма. Обращение матриц. Параллельные методы решения систем линейных уравнений. Параллельные численные алгоритмы решения дифференциальных уравнений в частных производных.

4.4	Параллельные алгоритмы /Лаб/	7	4	Л1.1 Л1.2 Л1.3Л1.4 Л2.2 Л2.3 Л2.4 Л2.7 Л2.9 Л2.11	Практическая реализация параллельных алгоритмов. Параллельные программы рекурсивных вычислений. Параллельные численные алгоритмы линейной алгебры. Практическая реализация алгоритмов разбиения матриц (горизонтальная, вертикальная, блочные схемы).
4.5	Параллельные алгоритмы /Ср/	7	2	Л1.1 Л1.2 Л1.3Л1.4 Л2.2 Л2.3 Л2.4 Л2.7 Л2.9 Л2.11	Практическая реализация параллельных алгоритмов. Параллельные программы рекурсивных вычислений. Параллельные численные алгоритмы линейной алгебры. Практическая реализация алгоритмов разбиения матриц (горизонтальная, вертикальная, блочные схемы).
4.6	Практические приемы параллельного решения уравнений /Лаб/	7	4	Л1.1 Л1.2 Л1.3Л1.4 Л2.2 Л2.3 Л2.4 Л2.7 Л2.9 Л2.11	Параллельные методы решения систем линейных уравнений. Параллельные численные алгоритмы решения дифференциальных уравнений в частных производных.
4.7	Практические приемы параллельного решения уравнений /Ср/	7	6	Л1.1 Л1.2 Л1.3Л1.4 Л2.2 Л2.3 Л2.4 Л2.7 Л2.9 Л2.11	Параллельные методы решения систем линейных уравнений. Параллельные численные алгоритмы решения дифференциальных уравнений в частных производных.
4.8	Работа над индивидуальным проектом /Ср/	7	4	Л1.1 Л1.2 Л1.3Л1.4 Л2.2 Л2.3 Л2.4 Л2.7 Л2.9 Л2.11	
4.9	Подготовка к зачету /Ср/	7	2	Л1.1 Л1.2 Л1.3Л1.4 Л2.2 Л2.3 Л2.4 Л2.7 Л2.9 Л2.11	Вопросы к зачету
4.10	Параллельное программирование и реализация параллельных алгоритмов /КСР/	7	2	Л1.1 Л1.2 Л1.3	

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

5.1. Типовые задания для проведения текущего контроля

Образцы заданий к лабораторным работам:

1. Написать программу, используя коммуникационные функции, передающие одномерные и двумерные массивы (вектора и матрицы) между двумя процессорами
2. Провести сравнение по скорости передачи данных в зависимости от применяемых функций и размера передаваемых данных
3. Написать программу, используя коммуникационную функцию, реализующую алгоритм передачи данных от 0 процесса всем остальным
4. Написать программу, используя коммуникационную функцию, реализующую алгоритм передачи частей массива от всех процессоров на 0.
5. Написать программу, используя коммуникационную функцию, реализующую алгоритм передачи частей массива от всех процессоров на все процессора.

Индивидуальное проектное задание заключается в разработке параллельной программы:

Каждый проект независимо от темы и задания должен предоставлять пользователю выбор количества узлов, на которых будет запущен процесс вычислений, а также выводить, время, за которое они произошли.

Индивидуальное проектное задание заключается в разработке параллельной программы:

Каждый проект независимо от темы и задания должен предоставлять пользователю выбор количества узлов, на которых будет запущен процесс вычислений, а также выводить, время, за которое они произошли.

1. Нахождение кратчайших путей для ориентированного графа

Исходной информацией для задачи является взвешенный граф, $G=(V,R)$ содержащий n вершин ($|V|=n$), в котором каждому ребру графа приписан неотрицательный вес. Граф будем полагать ориентированным, т.е., если из вершины i есть ребро в вершину j , то из этого не следует наличие ребра из j в i . В случае, если вершины все же соединены взаимнообратными ребрами, то веса, приписываемые им, могут не совпадать. Для имеющегося графа G требуется найти минимальные длины

путей между каждой парой вершин графа.

Входные данные начинаются N количеством вершин от 0 до 10000 и M количеством ребер в графе от 0 до 10000 через пробел, затем в M строках записаны номера вершин соединенных i-ым ребром и его вес от 0 до 10000, где i – номер строки, начиная счет со второй строки.

Выходные данные должны содержать все кратчайшие пути для всех пар вершин, если такой путь существует для пары. Каждый путь должен быть записан на отдельной строке в виде последовательности номеров ребер через пробел в том порядке, в котором они включены в путь, и к ним добавить последнее после знака «=» число, показывающее вес пути, в начале строки должны быть помещены номера вершин, соединенных этим путем («I J : »). В случае отсутствия пути между вершинами поставить вместо последовательности вершин прочерк.

Пример входных(input.txt) и выходных(output.txt) данных:

input.txt output.txt

```
5 4
1 2 1
1 3 8
1 4 3
2 3 2
3 1 3
4 2 1
1 2 : 1 = 1
1 3 : 1 4 = 4
1 4 : 3 = 3
1 5 : -
2 1 : -
2 3 : 4 = 3
2 4 : -
2 5 : -
3 1 : 5 = 3
3 2 : -
3 4 : -
3 5 : -
4 1 : -
4 2 : 6 = 1
4 3 : -
4 5 :-
5 1 : -
5 2 : -
5 3 : -
5 4 : -
```

Примечание: в случае если наименьший путь между парой вершин не является единственным, считать кратчайшим тот, который содержит наименьшее количество вершин, если же данное условие не дает единственное решение, вывести любой из них.

2. Решение систем линейных уравнений методом алгоритма Гаусса, рекурсивное удвоение Стоуна

Задача: дана расширенная (добавлен вектор-столбец свободных членов) матрица коэффициентов системы линейных уравнений, найти решение системы.

Входной файл в первой строке содержит N количество строк и M количество столбцов матрицы через пробел, далее в N содержится M элементов (числа от -10000 до 10000) матрицы через пробел. Порядок расположения элементов матриц внутри файла соответствует порядку расположения элементов в самой матрице.

$-10000 \leq N, M \leq 10000$

Выходной файл должен содержать значение переменных, совокупность которых является решением данной системы линейных уравнений, через пробел. Порядок расположения значений переменных должен соответствовать порядку номеров их переменных. В случае невозможности нахождения единственного решения вывести в файл сообщение «Error Data». В случае решения системы равного пустому множеству вывести сообщение «No Solutions»

Пример входных(input.txt) и выходных(output.txt) данных:

input.txt output.txt

```
1 2
-7 21
-3
input.txt output.txt
2 3
1 1 5
1 -2 -4
```

5.2. Типовые задания для проведения промежуточной аттестации

Вопросы к экзамену

1. Что такое параллельное программирование и суперкомпьютеры. Области в которых может возникать потребность в параллельных вычислениях. Особенности параллельных вычислений.
2. Увеличение производительности при параллельных вычислениях. К каким областям задач может быть

- применено распараллеливание к каким нет. Закон Амдала.
3. Ускорение параллельного алгоритма.
 4. Эффективность параллельного алгоритма.
 5. Распараллеливание арифметических выражений. Алгоритм Винограда.
 6. Разделение на подзадачи. Установление связей между отдельными подзадачами. Объединение мелких подзадач в большие, законченные счетные единицы (агломерация).
 7. Средства поддержки параллельной работы: параллелизм процессов, механизм синхронизации и механизм разделения ресурсов, семафоры, обмен сообщениями.
 8. Библиотека MPI. Инициализация, завершение. Базовые функции.
 9. Точечные обмены. Синхронный и асинхронный режим.
 10. Буферизация сообщений.
 11. Отложенные операции. Функции инициализации и завершения операции.
 12. Коллективные взаимодействия. Передача сообщений
 13. Управление группами и коммутатором.
 14. Коллективные взаимодействия. Прием сообщений
 15. Создание и освобождение коммутатора.
 16. Типы данных в MPI.
 17. Библиотека OpenMP. Инициализация, завершение. Базовые функции.
 18. Библиотека OpenMP. Управление потоками. Управление областями видимости
 19. Библиотека OpenMP. Распределение вычислений между потоками.
 20. Библиотека OpenMP. Синхронизация параллельных процессов.
 21. Простейшие параллельные алгоритмы и их степень параллелизма.
 22. Класс thread. Инициализация, завершение. Базовые методы.
 23. Класс thread. Управление потоками. Доступ к данным и объектам.
 24. Класс thread. Распределение вычислений между потоками.
 25. Класс thread. Синхронизация параллельных процессов.
 26. Простейшие параллельные алгоритмы и их степень параллелизма.
 27. Параллельный алгоритм скалярного умножения векторов и его ускорение по сравнению с последовательным алгоритмом.
 28. Параллельный алгоритм умножения матрицы на вектор и его ускорение по сравнению с последовательным алгоритмом.
 29. Параллельный алгоритм умножения матрицы на матрицу и его ускорение по сравнению с последовательным алгоритмом.
 30. Параллельный алгоритм решения СЛАУ прямым методом Гаусса и его ускорение по сравнению с последовательным алгоритмом.
 31. Технологии параллельного программирования.

5.3. Перечень видов оценочных средств

1. Лабораторные работы.
2. Индивидуальное проектное задание.
3. Вопросы к экзамену

5.4. Процедура применения оценочных материалов

Промежуточная аттестация может проводиться с применением электронного обучения и (или) дистанционных образовательных технологий в соответствии с "Порядком проведения промежуточной аттестации с применением электронного обучения и /или дистанционных образовательных технологий".

Описание балльно-рейтинговой системы по дисциплине.

Итоговая рейтинговая оценка по дисциплине складывается из следующих составляющих:

- 1) В течение семестра за выполнение заданий по курсу студент может максимально получить 60 баллов;
- 2) Обязательной формой текущей аттестации знаний является выполнение индивидуального проектного задания 20 баллов.
- 3) На зачете ответ студента может быть максимально оценен в 30 баллов.

Выполнение заданий:	70
Лабораторные работы	50
Индивидуальное проектное задание	20
Зачет	30
ИТОГО:	100

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Рекомендуемая литература

6.1.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание

	Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание
Л1.1	Гергель В. П.	Теория и практика параллельных вычислений: учебное пособие	, 2007	http://biblioclub.ru/index.php?page=book&id=233067
Л1.2	Савельев В. А., Штейнберг Б. Я.	Распараллеливание программ: учебник	, 2008	http://biblioclub.ru/index.php?page=book&id=240965
Л1.3	Биллиг В. А.	Параллельные вычисления и многопоточное программирование	, 2016	http://biblioclub.ru/index.php?page=book&id=428948

6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание
Л2.1	Туральчук К. А.	Параллельное программирование с помощью языка C#: учебное пособие	, 2016 ИНТУИТ	http://biblioclub.ru/index.php?page=book_red&id=429098&sr=1
Л2.2	Антонов А. С.	Параллельное программирование с использованием технологии MPI: курс	, 2008	http://biblioclub.ru/index.php?page=book&id=233577
Л2.3	Хиценко В. П.	Основы программирования: учебное пособие	НГТУ, 2015	http://biblioclub.ru/index.php?page=book&id=438365
Л2.4	Грузина Э. Э., Иванов К. С., Бондарева Л. В.	Программирование. C++: электронное учебное пособие	Кемеровский государственный университет, 2015	http://biblioclub.ru/index.php?page=book&id=481536
Л2.5	Кудрина Е. В., Огнева М. В.	Основы алгоритмизации и программирования на языке c#: Учебное пособие	, 2019	https://www.biblio-online.ru/book/osnovy-algoritmizacii-i-programmirovaniya-na-yazyke-c-431505
Л2.6	Тюкачев Н. А., Хлебостроев В. Г.	C#. Основы программирования: учебное пособие	, 2018	https://e.lanbook.com/book/104962
Л2.7	Малявко А. А.	Параллельное программирование на основе технологий OpenMP, MPI, CUDA: Учебное пособие для вузов	Москва: Юрайт, 2020	https://urait.ru/bcode/453248
Л2.8	Абрамян, М. Э.	Практикум по параллельному программированию с использованием электронного задачника Programming Taskbook for MPI	Ростов-на-Дону: Издательство Южного федерального университета, 2010	http://www.iprbookshop.ru/47085.html
Л2.9	Антонов, А. С.	Параллельное программирование с использованием технологии MPI	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016	http://www.iprbookshop.ru/73704.html
Л2.10	Туральчук, К. А.	Параллельное программирование с помощью языка C#	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019	http://www.iprbookshop.ru/79714.html
Л2.11	Малявко А. А.	Параллельное программирование на основе технологий openmp, cuda, opencl, mpi: Учебное пособие для вузов	Москва: Юрайт, 2021	https://urait.ru/bcode/467800

6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

Э1	Среда электронного обучения LMS Moodle [Электронный ресурс] URL: http://moodle.tspu.ru/
6.3. Информационные технологии	
6.3.1 Перечень лицензионного и свободно распространяемого программного обеспечения	
1.	Операционная система ROSA Enterprise Linux Desktop № RL00450-1-110518-01. RL00450-1-110518-17 от 11 мая 2018 г.
2.	Операционная система Microsoft Windows Professional 7 Russian. Лицензия №48497058 от 13.05.2011 г., договор № Пр/16/6 от 05 апреля 2016 г.
3.	Операционная система Microsoft Windows 10 Professional Russian. Контракт № ПР/ФЕН/15/18 от 23.10.2015 г., договор № Пр/16/6 от 05 апреля 2016 г.
4.	Текстовый редактор NotePad++. Свободно распространяемое ПО
5.	Файловый менеджер Far manager. Свободно распространяемое ПО
6.	Оболочка программирования Code: Blocks 17.12. Свободно распространяемое ПО
7.	Среда программирования и набор инструментов для программирования. MinGW 0.6.3 Свободно распространяемое ПО
6.3.2 Перечень информационных справочных систем и профессиональных баз данных	
1.	Полнотекстовый архив ведущих западных научных журналов на российской платформе Национального электронно-информационного консорциума (НЭИКОН)(http://neicon.ru)
2.	Базы данных издательства Springer (https://link.springer.com)
3.	Web of Science Core Collection – политематическая реферативно-библиографическая и наукометрическая (библиометрическая) база данных (http://webofscience.com)
4.	Портал Федеральных государственных образовательных стандартов высшего образования (http://fgosvo.ru)

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Ауд.	Назначение	Оборудование и технические средства обучения	Вид
4-306	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	
4-305	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, аудиоколонки учебные, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	
4-303	Помещение для самостоятельной работы	аудиоколонки, кондиционер, маркерная доска, столы компьютерные, столы учебные, компьютерная техника с возможностью подключения сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду университета	

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Приступая к изучению новой учебной дисциплины, студенты должны ознакомиться с учебной программой, учебной, научной и методической литературой, имеющейся в библиотеке университета, встретиться с преподавателем, ведущим дисциплину, получить в библиотеке рекомендованные учебники и учебно-методические пособия, осуществить запись на соответствующий курс в среде электронного обучения университета.

Глубина усвоения дисциплины зависит от активной и систематической работы студента на лекциях и лабораторных работ, а также в ходе самостоятельной работы, по изучению рекомендованной литературы.

На лекциях важно сосредоточить внимание на ее содержании. Это поможет лучше воспринимать учебный материал и уяснить взаимосвязь проблем по всей дисциплине. Основное содержание лекции целесообразнее записывать в тетради в виде ключевых фраз, понятий, тезисов, обобщений, схем, опорных выводов. Необходимо обращать внимание на термины, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации. Желательно оставлять в конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющей материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. С целью уяснения теоретических положений, разрешения спорных ситуаций необходимо задавать преподавателю уточняющие вопросы. Для закрепления содержания лекции в памяти, необходимо во время самостоятельной работы внимательно прочесть свой конспект и дополнить его записями из учебников и рекомендованной литературы. Конспектирование читаемых лекций и их последующая доработка способствует более глубокому усвоению знаний, и поэтому являются важной формой учебной деятельности студентов.

Прочное усвоение и долговременное закрепление учебного материала невозможно без продуманной самостоятельной работы. Такая работа требует от студента значительных усилий, творчества и высокой организованности. В ходе самостоятельной работы студенты выполняют следующие задачи: дорабатывают лекции, изучают рекомендованную литературу, готовятся к лабораторным работам, к коллоквиуму, контрольным работам по отдельным темам дисциплины. При этом эффективность учебной деятельности студента во многом зависит от того, как он распорядился выделенным для самостоятельной работы бюджетом времени.

Результатом самостоятельной работы является прочное усвоение материалов по предмету согласно программе дисциплины. В итоге этой работы формируются профессиональные умения и компетенции, развивается творческий подход

к решению возникших в ходе учебной деятельности проблемных задач, появляется самостоятельности мышления.

Целью лабораторных работ по данной дисциплине является закрепление теоретических знаний, полученных при изучении дисциплины.

При подготовке к лабораторной работе целесообразно выполнить следующие рекомендации: изучить основную литературу; ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях: журналах, газетах и т. д.; при необходимости доработать конспект лекций. При этом учесть рекомендации преподавателя и требования учебной программы.

При выполнении лабораторных работ основным методом обучения является самостоятельная работа студента под управлением преподавателя. На них пополняются теоретические знания студентов, их умение творчески мыслить, анализировать, обобщать изученный материал, проверяется отношение студентов к будущей профессиональной деятельности.

Оценка выполненной работы осуществляется преподавателем комплексно: по результатам выполнения заданий, устному сообщению и оформлению работы. После подведения итогов занятия студент обязан устранить недостатки, отмеченные преподавателем при оценке его работы.