

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Макаренко Елена Николаевна

Должность: Ректор

Дата подписания: 14.04.2021 22:23:17

Уникальный программный ключ:

c098bc0c1041cb2a4c27c1171d67151b9a6ca00ad8c37b55cbe1e2db17c78

**Министерство образования и науки Российской Федерации**  
**Федеральное государственное бюджетное образовательное учреждение высшего образования**  
**«Ростовский государственный экономический университет (РИНХ)»**



**УТВЕРЖДАЮ**  
**Первый проректор –**  
**проректор по учебной работе**  
**Н.Г. Кузнецов**  
**«01» июня 2018г.**

**Рабочая программа дисциплины**  
**Параллельное программирование**

по профессионально-образовательной программе направление 09.03.04  
"Программная инженерия"

Квалификация

Бакалавр

Ростов-на-Дону  
2018 г.

## КАФЕДРА Информационные технологии и защита информации

### Распределение часов дисциплины по курсам

Курс	5		Итого	
	уп	рпд		
Лекции	6	6	6	6
Лабораторные	10	10	10	10
В том числе инт.	6	6	6	6
Итого ауд.	16	16	16	16
Контактная	16	16	16	16
Сам. работа	227	227	227	227
Часы на контроль	9	9	9	9
Итого	252	252	252	252

### ОСНОВАНИЕ

Федеральный государственный образовательный стандарт высшего образования по направлению подготовки 09.03.04 "Программная инженерия" (уровень бакалавриата) (приказ Минобрнауки России от 12.03.2015 г. № 229)


Рабочая программа составлена

по профессионально-образовательной программе направление  
09.03.04 "Программная инженерия"

Учебный план утвержден учёным советом вуза от 27.03.2018 протокол № 10.

Программу составил(и): к.э.н., доцент, Жилина Е.В.  11.05.18

Зав. кафедрой д.э.н., профессор Тищенко Е.Н.  11.05.18

Методическим советом направления к.ф.-м.н., декан, Карасев Д.Н.  16.05.18

Отделом образовательных программ и  
планирования учебного процесса Торопова Т.В.  30.05.18

Проректором по учебно-  
методической работе Джуха В.М.  31.05.18

---

**Визирование РПД для исполнения в очередном учебном  
году**

Отдел образовательных программ и планирования  
учебного процесса Торопова Т.В. \_\_\_\_\_

Рабочая программа пересмотрена, обсуждена и одобрена для  
исполнения в 2019-2020 учебном году на заседании  
кафедры **Информационные технологии и защита информации**

Зав. кафедрой д.э.н., профессор Тищенко Е.Н. \_\_\_\_\_

Программу составил(и) *ж.э.н., доцент, Жилина Е.В.* \_\_\_\_\_

---

**Визирование РПД для исполнения в очередном учебном  
году**

Отдел образовательных программ и планирования  
учебного процесса Торопова Т.В.

Рабочая программа пересмотрена, обсуждена и одобрена для  
исполнения в 2020-2021 учебном году на заседании  
кафедры **Информационные технологии и защита информации**

Зав. кафедрой д.э.н., профессор Тищенко Е.Н. \_\_\_\_\_

Программу составил(и): *к.э.н., доцент, Жилина Е.В.* \_\_\_\_\_

---

**Визирование РПД для исполнения в очередном учебном  
году**

Отдел образовательных программ и планирования  
учебного процесса Торопова Т.В.

Рабочая программа пересмотрена, обсуждена и одобрена для  
исполнения в 2021-2022 учебном году на заседании  
кафедры **Информационные технологии и защита информации**

Зав. кафедрой д.э.н., профессор Тищенко Е.Н. \_\_\_\_\_

Программу составил(и): *к.э.н., доцент, Жилина Е.В.* \_\_\_\_\_

---

**Визирование РПД для исполнения в очередном учебном  
году**

Отдел образовательных программ и планирования  
учебного процесса Торопова Т.В.

Рабочая программа пересмотрена, обсуждена и одобрена для  
исполнения в 2022-2023 учебном году на заседании  
кафедры **Информационные технологии и защита информации**

Зав. кафедрой д.э.н., профессор Тищенко Е.Н. \_\_\_\_\_

Программу составил(и): *к.э.н., доцент, Жилина Е.В.* \_\_\_\_\_

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ	
1.1	Цель освоения дисциплины: освоение базовых знаний по вопросам организации параллельных вычислительных систем, а также основных технологий организации параллельных вычислений на многопроцессорных вычислительных комплексах с распределенной или общей оперативной памятью.
1.2	Задачи освоения дисциплины: знакомство с основными направлениями в области организации параллельных вычислений на многопроцессорных вычислительных системах; освоение технологий параллельного программирования; приобретение навыков параллельного программирования с использованием интерфейса передачи сообщений; знакомство с технологией параллельного программирования на системах с общей оперативной памятью; приобретение навыков распараллеливания математических алгоритмов.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	
Цикл (раздел) ООП:	Б1.В.ДВ.04
2.1	<b>Требования к предварительной подготовке обучающегося:</b>
2.1.1	Необходимыми условиями для успешного освоения дисциплины являются знания, умения и навыки, полученные в результате изучения дисциплин:
2.1.2	Сети и системы передачи информации
2.1.3	Аппаратное обеспечение вычислительных систем
2.1.4	Инструменты и методы программной инженерии
2.1.5	Математический анализ
2.2	<b>Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:</b>
2.2.1	Научно-исследовательская работа
2.2.2	Преддипломная
2.2.3	Итоговая государственная аттестация
2.2.4	Преддипломная
2.2.5	Научно-исследовательская работа
2.2.6	Подготовка к сдаче и сдача государственного экзамена

3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ	
<b>ПК-1: готовностью применять основные методы и инструменты разработки программного обеспечения</b>	
<b>Знать:</b>	
инструменты и библиотеки разработки программного обеспечения с использованием параллельных вычислений	
<b>Уметь:</b>	
строить модель выполнения параллельных вычислений, оценивать эффективности параллельных вычислений с помощью инструментов разработки программного обеспечения	
<b>Владеть:</b>	
навыками использования различных методов межпроцессного взаимодействия	
<b>ПК-3: владением навыками использования различных технологий разработки программного обеспечения</b>	
<b>Знать:</b>	
основные подходы к разработке параллельных программ	
<b>Уметь:</b>	
использовать различные технологии разработки программного обеспечения с использованием параллельных вычислений	
<b>Владеть:</b>	
навыками реализации параллельных программ с помощью различных технологий	

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)							
Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетенции	Литература	Интер акт.	Примечание
	Раздел 1. 1. Введение в параллельные вычисления						

1.1	Тема 1.1 "Основы параллельных вычислений" Цели и задачи параллельной обработки данных. Необходимость и актуальность параллельных вычислений. Различия между многозадачным, параллельным и распределенным режимами выполнения программ. Закон Амдала. Закон Мура. Гипотеза Минского. Способы построения многопроцессорных вычислительных систем. Краткая история развития высокопроизводительных вычислений. Примеры параллельных вычислительных систем. /Лек/	5	2	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	1	
1.2	Тема 1.1 "Основы параллельных вычислений" Запуск MPI программ. Компиляция. Настройки для сборки и счета. Запуск в пакете. Шесть базовых вызовов. Сообщения MPI. Коммуникаторы. /Лаб/	5	2	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	1	
1.3	Тема 1.1 "Введение в параллельные вычисления" Рейтинги ведущих суперкомпьютеров /Ср/	5	2	ПК-1	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
1.4	Тема 1.2 "Обзор параллельных вычислительных систем и их классификация" Классификация параллельных вычислений по Флинну и Кришнамэрфи. Понятия мультипроцессора, мультикомпьютера, вычислительного кластера. Способы повышения производительности компьютера: конвейеризация, кэширование команд и данных, увеличение количества регистров, суперскалярность и микросуперскалярность процессоров. Модели многопроцессорных систем (переход от RAM к PRAM), локальная память, модули с шинной связью, разделяемая общая память. Топологии межмодульной связи: шина, двумерная сетка, гиперкуб, деревья, коммутирующий граф. Архитектура суперкомпьютеров. Транспьютерные сетевые архитектуры. Распределенные системы: кластеры и сети (динамические, статические). Метрические характеристики вычислительной сети: диаметр, связность, ширина сечения, поток через сечение, стоимость. Оценки метрик для различных сетей. Логическая и физическая организации параллельных вычислительных систем (ПВС). /Лек/	5	2	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	1	

1.5	Тема 1.2 "Обзор параллельных вычислительных систем и их классификация" Упражнение на тупик. Распаралеливание метода наименьших квадратов. /Лаб/	5	2	ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
1.6	Тема 1.2 "Обзор параллельных вычислительных систем и их классификация" Сериализуемость, требования когерентности, стоимость пересылки в параллельных системах. Маршрутизация. /Ср/	5	4	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
1.7	Тема 1.3 "Способы организации параллельной обработки данных" Особенности организации параллельных вычислений в системах с общей памятью (обеспечение однозначности кэш-памяти разных процессоров, синхронизация вычислений). Особенности организации параллельных вычислений в системах с распределенной памятью посредством передачи сообщений. Топологии сетей передачи данных в мультимпьютерах. Типовые схемы коммуникации. /Лек/	5	2	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
1.8	Тема 1.3 "Способы организации параллельной обработки данных" Расчет относительной эффективности способов коммуникации. Влияние замены блокирующего получения на неблокирующее получение на синхронизационную накладку. /Лаб/	5	2	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	1	
1.9	Тема 1.4 "Вычислительные кластеры" Основные функции коллективной коммуникации: операциями широкой рассылки (broadcast), сбора (gather) и глобального приведения (global reduction). Расчет параллельного волнового уравнения. /Лаб/	5	2	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	1	
1.10	Тема 1.4 "Вычислительные кластеры" Понятие кластера и кластерной архитектуры. Классификация кластерных вычислительных систем. Состав сетевой инфраструктуры кластера. Типы топологий и критерии эффективности коммуникационной сети кластера. Сетевые решения для кластерных систем. Основные критерии оценки кластерных систем. Типичный набор программно-аппаратного обеспечения кластеров. Особенности запуска задач на кластерах. Системы управления заданиями. Интегрированные наборы кластерного программного обеспечения. /Ср/	5	4	ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
	<b>Раздел 2. Параллельные алгоритмы. Стандарт MPI</b>						

2.1	<p>Тема 2.1 "Реализация параллельных алгоритмов"</p> <p>Отличия параллельных алгоритмов от последовательных алгоритмов. Постановка задачи распараллеливания существующих сериальных алгоритмов. Пример суммирования массива. Реализация параллельных алгоритмов с использованием процессов. Виды межпроцессной коммуникации (сигналы, семафоры, разделяемая память, каналы, сообщения). Оценка эффективности алгоритма с учетом времени доставки кода и данных вычислительному узлу. Законы Амдала. Целесообразность распараллеливания. Как анализировать зависимость данных и кода в последовательной программе. Порядок вычислений, граф зависимости. Модель «грубой силы». Примеры решений для PRAM, BSP, LogP. Преобразования параллелизации циклов. Доставка данных, отображение массива данных на распределенную систему, избыточность. /Ср/</p>	5	10	ПК-1	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
2.2	<p>Тема 2.1 "Реализация параллельных алгоритмов"</p> <p>Параллельные алгоритмы циклического сдвига на системе с распределенной памятью со структурой сети связей между процессами - "кольцо", в которых каждый процесс инициирует рассылку своего сообщения одновременно в каком-либо выбранном направлении по кольцу (в MPI):</p> <ul style="list-style-type: none"> <li>- с асинхронной блокированной передачей, блокированным приемом.</li> <li>- с асинхронной неблокированной передачей, неблокированным приемом.</li> <li>- с синхронной передачей, блокированным приемом.</li> </ul> <p>/Ср/</p>	5	4	ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
2.3	<p>Тема 2.1 "Реализация параллельных алгоритмов"</p> <p>Влияние аппаратной реализации PRAM и проблемы когерентности данных. Модели конкурентной и исключительной примитивных операций с памятью (EREW, CRCW, CREW, QRQW) /Ср/</p>	5	2	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
2.4	<p>Тема 2.2 "Моделирование параллельных программ. Реализация параллелизма различного вида"</p> <p>Показатели эффективности параллельного алгоритма и оценка максимально достижимого параллелизма. Параллелизм на примере модельных задач нахождения частных сумм последовательности числовых значений и умножения матриц. /Ср/</p>	5	4	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	

2.5	Тема 2.2 "Моделирование параллельных программ. Реализация параллелизма различного вида" Параллельные алгоритмы конвейерной передачи данных на системе с распределенной памятью со структурой сети связей между процессами - "линейка" (в MPI). /Лаб/	5	2	ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	1	
2.6	Тема 2.3 "Методы передачи данных. Стандарт MPI" MPI: основные понятия и определения. Базовый (минимальный) набор функций MPI, достаточный для разработки параллельных программ. Пример: программа вычисления числа $\rho$ . Операции передачи данных между двумя процессами. Коллективные операции передачи данных. Упаковка и распаковка разнотипных данных в MPI. Управление группами процессов и коммутаторами. Виртуальные топологии. Модельный пример: умножение матрицы на вектор /Ср/	5	2	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
2.7	Тема 2.3 "Методы передачи данных. Стандарт MPI" Параллельный алгоритм скалярного произведения векторов с использованием редуцированных операций (MPI_Reduce(...), MPI_Allreduce(...)). Параллельный алгоритм скалярного произведения векторов с использованием парных взаимодействий (MPI_Send(...), MPI_Recv(...)). Сравнить время выполнения алгоритмов (в MPI).  /Ср/	5	6	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
2.8	Тема 2.3 "Методы передачи данных. Стандарт MPI" Межпроцессное взаимодействие и операционных UNIX-средах. Осуществление распараллеливания на уровне процессов. Осуществление распараллеливания на уровне нитей. Синхронизация выполнения нитей в POSIX-совместимых системах.  /Ср/	5	4	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
2.9	Тема 2.4 "Протокол и библиотеки MPI" Состав функций, типы данных, организация обмена данными. Пример использования библиотеки MPI. Среда выполнения LAM. Установка, использование (компиляция и запуск программ, конфигурация вычислительной системы). /Ср/	5	4	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
2.10	Тема 2.4 "Протокол и библиотеки MPI" Параллельный алгоритм суммирования элементов векторов с использованием редуцированных операций (MPI_Reduce(...), MPI_Allreduce(...)) (в MPI). /Ср/	5	4	ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	



2.11	Тема 2.4 "Протокол и библиотеки MPI" Разработка распределенных приложений с использованием механизма передачи сообщений MPI. Сокращенный (минимальный) набор функций MPI. Пример использования. Функции MPI для обменов многие-к-одному и один-ко-многим. Особенности использования микросуперскалярности процессоров в компиляторах для получения эффективного кода. /Ср/	5	4	ПК-1	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
	<b>Раздел 3.3. Параллельные алгоритмы. Стандарт OpenMP</b>						
3.1	Тема 3.1 "Методика разработки параллельных алгоритмов" Принцип «Разделяй-и-властвуй». Виды декомпозиции задачи: по данным, по коду, по выполнению, комбинированные. Примеры различных декомпозиций. Сортировка, выпуклые оболочки, матричные операции: примеры декомпозиции рекурсивного вида. Матричные операции: пример декомпозиции по данным, одно-, двух- и трехмерное разбиения. Общая схема и методика разработки параллельных алгоритмов. Пример использования методики разработки параллельных алгоритмов для параллельного решения гравитационной задачи N тел. /Ср/	5	4	ПК-1	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
3.2	Тема 3.1 "Методика разработки параллельных алгоритмов" Выполнить вычислительный эксперимент для оценки эффективности параллельных вычислений (обработка двумерного массива: перестановки элементов, умножение матриц и векторов) с распараллеливанием вычислений с помощью технологии OpenMP. Варьировать размер массива от 10 до 10000, число потоков – 1,2,4,8. /Ср/	5	6	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
3.3	Тема 3.2 "Базовые средства параллельного программирования вычислительных кластеров" Общая характеристика методов передачи данных, оценка времени выполнения коммуникационных операций. Оценка трудоемкости операций передачи данных для кластерных систем. Модель Хокни. /Ср/	5	8	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
3.4	Тема 3.2 "Базовые средства параллельного программирования вычислительных кластеров" Выполнить вычислительный эксперимент для оценки эффективности параллельных вычислений (алгоритмы сортировки) с распараллеливанием вычислений с помощью технологии OpenMP. Варьировать размер массива от 10 до 10000, число потоков – 1,2,4,8. /Ср/	5	4	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	

3.5	<p>Тема 3.3 "Базовые средства параллельного программирования систем с общей памятью. Стандарт OpenMP"</p> <p>Общие сведения. Структура стандарта OpenMP. Достоинства технологии OpenMP. Модель параллелизма OpenMP. Модель памяти OpenMP. Директивы OpenMP. Типы директив. Формат записи директив. Определение параллельной области. Распределение вычислений между потоками. Директивы синхронизации. Директивы управления областью видимости данных. Совместимость директив и их параметров. Библиотека функций OpenMP. Функции для контроля/запроса параметров среды исполнения. Функции синхронизации. Переменные среды исполнения. Пример программы произведения матриц. Сравнение технологий MPI и OpenMP для SMP-систем. Гибридный (MPI+OpenMP) подход для SMP-кластеров. /Ср/</p>	5	6	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
3.6	<p>Тема 3.3 "Базовые средства параллельного программирования систем с общей памятью. Стандарт OpenMP"</p> <p>Выполнить вычислительный эксперимент для оценки эффективности параллельных вычислений (решение системы линейных алгебраических уравнений методом Гаусса) с распараллеливанием вычислений с помощью технологии многопоточности на основе классов .NET (Thread, Task, Parallel). Варьировать размер сетки от 100 до 10000 точек, число потоков – 1,2,4,8. /Ср/</p>	5	4	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
3.7	<p>Тема 3.3 "Базовые средства параллельного программирования систем с общей памятью. Стандарт OpenMP"</p> <p>Компиляторы Intel с поддержкой OpenMP. Инструментальные средства разработки и отладки многопоточных приложений. Примеры /Ср/</p>	5	10	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
3.8	<p>Тема 3.4 "Гибридная модель параллельного программирования"</p> <p>Совместное использование технологий программирования MPI, OpenMP. Решение задач с привлечением обеих технологий — MPI и OpenMP. Разбор особенностей совместного использования. /Ср/</p>	5	6	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
3.9	<p>Тема 3.4 "Гибридная модель параллельного программирования"</p> <p>Параллельные алгоритмы скалярного произведения векторов: с использованием редуцированной операции (reduction(...)); с использованием директив critical{...} или atomic. /Ср/</p>	5	6	ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	

3.10	Тема 3.4 "Гибридная модель параллельного программирования" Архитектура приложений в DCOM и CORBA. Сравнение технологий CORBA и DCOM. MICO – свободно распространяемая версия CORBA. Программирование для CORBA. Язык описания интерфейсов IDL в CORBA. /Ср/	5	8	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
	<b>Раздел 4. 4. Алгоритмы распараллеливания сортировки. Параллельные численные методы</b>						
4.1	Тема 4.1 "Принципы распараллеливания сортировки" Основные последовательные алгоритмы сортировки данных. Масштабирование параллельных вычислений. Параллельные методы сортировки. Пузырьковая сортировка. Сортировка Шелла. Быстрая сортировка. Обобщенный алгоритм. /Ср/	5	4	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
4.2	Тема 4.1 "Принципы распараллеливания сортировки" Матрица M1 распределена по процессам горизонтальными ленточными полосами, матрица M2 распределена вертикальными ленточными полосами. Матрицы M1 и M2 обе распределены по процессам горизонтальными ленточными полосами. /Ср/	5	4	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
4.3	Тема 4.2 "Многопоточное программирование с использованием классов MS Visual Studio .NET" Параллельное программирование в среде Microsoft Visual Studio. Многопоточная обработка. Методы и свойства класса Thread. Фоновые потоки. Класс BackgroundWorker. Многозадачность на основе класса Task. Фабрика задач. Обобщенные классы задач и фабрики задач Task TResult, TaskFactory TResult, методы и свойства классов. Параллелизм задач. Параллелизм данных. Класс Parallel; методы Parallel.Invoke; Parallel.For; Parallel.ForEach. /Ср/	5	6	ПК-1	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	

4.4	<p>Тема 4.2 "Многопоточное программирование с использованием классов MS Visual Studio .NET" Параллельный алгоритм умножения матрицы <math>M</math> на вектор <math>V</math> с использованием директивы распараллеливания параметрических циклов <code>#pragma omp for</code>. Параллельный алгоритм умножения матрицы <math>M</math> на вектор <math>V</math> с использованием "ручного" задания работ (распараллеливания циклов). Параллельный алгоритм умножения матрицы на матрицу с использованием директивы распараллеливания параметрических циклов <code>#pragma omp for</code>. Сравнить показатели ускорения и эффективности приведенных алгоритмов, а так же сравнить их с аналогичными показателями этих же алгоритмов реализованных на системах с распределенной памятью.</p> <p>/Ср/</p>	5	6	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
4.5	<p>Тема 4.3 "Параллельные численные методы"</p> <p>Умножение матрицы на вектор. Произведение матриц (алгоритм Штрассена). Параллельная реализация волнового алгоритма. Быстрое преобразование Фурье. Решение системы линейных уравнений с трехдиагональной матрицей — метод редукции. Анализ вычислительной сложности алгоритмов, анализ накладных расходов на синхронизацию параллельных подзадач. Формулировка ряда численных методов, допускающих параллельное исполнение. Умножение матрицы на вектор. Произведение матриц напрямую, формулировка и обсуждение вариантов параллельной реализации алгоритма Штрассена. Параллельная реализация волнового алгоритма поиска кратчайшего пути в графе для случая неориентированного планарного графа, топологически эквивалентного шахматной доске. /Ср/</p>	5	8	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
4.6	<p>Тема 4.3 "Параллельные численные методы"</p> <p>Решение системы линейных алгебраических уравнений <math>Ax = f</math> методом простой итерации. /Ср/</p>	5	4	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	

4.7	Тема 4.3 "Параллельные численные методы" Параллельная реализация быстрого преобразования Фурье. Решение системы линейных уравнений с трехдиагональной матрицей — метод редукции. Анализ вычислительной сложности алгоритмов, анализ накладных расходов на синхронизацию параллельных подзадач. /Ср/	5	12	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
	<b>Раздел 5. 5. Высокопроизводительные вычисления</b>						
5.1	Тема 5.1 "Высокоуровневые средства программирования многопроцессорных систем. DVM-система" DVM-система. Общие сведения, цели создания, принципы построения. Модель параллелизма, модель выполнения и модель программирования DVM. Языки программирования DVM. Директивы DVM (на примере языка C-DVM). Сравнение размеров и эффективности MPI- и DVM-программ. Переносимость и повторное использование DVM-программ. Средства функциональной отладки, анализа и прогноза производительности DVM-программ. Особенности компиляции и запуска DVM-программ. Примеры программ на языке C-DVM. /Ср/	5	6	ПК-1	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
5.2	Тема 5.1 "Высокоуровневые средства программирования многопроцессорных систем. DVM-система" Решение системы линейных алгебраических уравнений $Ax = f$ методом Гаусса. /Ср/	5	4	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
5.3	Тема 5.2 "Высокопроизводительные вычисления с применением графических процессоров (GPU). Технология NVidia CUDA" Введение в вычисления общего назначения с использованием GPU. Основные архитектурные отличия GPU от CPU. Архитектура современных GPU. Технология CUDA. Модели и шаблоны программирования с использованием технологии CUDA. Модель памяти CUDA. Типы памяти. Оптимизация CUDA-приложений. Модель исполнения CUDA. Компиляция CUDA-приложений. CUDA-расширение языка C (спецификаторы функций, спецификаторы переменных, встроенные переменные, директивы запуска ядра). Некоторые функции API CUDA Runtime. Пример программы на CUDA. Производство матриц. Ускорение Matlab-расчетов на GPU. /Ср/	5	5	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	

5.4	Тема 5.2 "Высокопроизводительные вычисления с применением графических процессоров (GPU). Технология NVidia CUDA" Параллельные алгоритмы строятся одновременно в MPI и OpenMP. Параллельные вычисления внутри узла вычислительного кластера осуществляются в OpenMP, а взаимодействия процессов, расположенных в разных узлах, осуществляются в MPI: Параллельные алгоритмы вычисления определенного интеграла $y=f(x)dx$ с использованием метода прямоугольников (трапеций) для $f(x)=\sin(x)$ , $f(x)=x^2$ , $f(x)=e^x$ /Ср/	5	6	ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
5.5	К разделу 5. mpC – Расширение языка Си для параллельных вычислений. Cilk – расширение языка Си, синтаксис и семантика. Понятие процесса и базовые процессы в языке OCCAM. /Ср/	5	10	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	
5.6	Контрольная работа. Перечень тем представлен в Приложении 1 к рабочей программе дисциплины. /Ср/	5	36	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4	0	
5.7	/Экзамен/	5	9	ПК-1 ПК-3	Л1.1 Л1.2 Л1.3 Л2.1 Л2.2 Л2.3 Л3.1 Л3.2 Л3.3 Э1 Э2 Э3 Э4 Э5 Э6 Э7 Э8 Э9	0	

## 5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

### 5.1. Фонд оценочных средств для проведения промежуточной аттестации

Вопросы к экзамену, 5 курс:

1. Основы MPI
2. Запуск MPI программ
3. Группы коммутаторов и процессов.
4. Сообщения MPI.
5. Коммутаторы.
6. Блокирующее поведение
7. Неблокирующее поведение. Неблокирующие вызовы
8. Способы коммуникации.
9. Синхронная коммуникация.
10. Коммуникация по готовности.
11. Буферизованная коммуникация.
12. Стандартизованная коммуникация.
13. Основные функции коллективной коммуникации.
14. Операция широкой рассылки (broadcast).
15. Операция сбора (gather).
16. Операция глобального приведения (global reduction).
17. Изменение кода параллельной программы.
18. Использование функции MPI\_Scatterv.
19. Использование функции MPI\_Scatter.
20. Использование функции MPI\_Gatherv.
21. Основные функции, необходимые для создания групп и коммутаторов/
22. Основные функции, необходимые для получения информации о коммутаторах.
23. Основные типы данных MPI.

24. Производные типы данных.
25. Общие типы данных и карты типов. Диапазон типа данных.
26. Общая схема и методика разработки параллельных алгоритмов. Принцип «Разделяй-и-властвуй». Виды декомпозиции задачи: по данным, по коду, по выполнению, комбинированные. Примеры различных декомпозиций.
27. Сортировка, выпуклые оболочки, матричные операции: примеры декомпозиции рекурсивного вида.
28. Матричные операции: пример декомпозиции по данным, одно-, двух- и трехмерное разбиения.
29. Общая схема и методика разработки параллельных алгоритмов.
30. Оценка трудоемкости операций передачи данных для кластерных систем. Модель Хокни.
31. Базовые средства параллельного программирования систем с общей памятью. Стандарт OpenMP. Общие сведения. Структура стандарта OpenMP. Достоинства технологии OpenMP.
32. Модель параллелизма OpenMP. Модель памяти OpenMP. Директивы OpenMP. Типы директив. Формат записи директив.
33. OpenMP. Определение параллельной области. Распределение вычислений между потоками. Директивы синхронизации. Директивы управления областью видимости данных. Совместимость директив и их параметров.
34. Библиотека функций OpenMP. Функции для контроля/запроса параметров среды исполнения. Функции синхронизации. Переменные среды исполнения.
35. Сравнение технологий MPI и OpenMP. Гибридный (MPI+OpenMP) подход для SMP-кластеров. Инструментальные средства разработки и отладки многопоточных приложений.
36. Принципы распараллеливания сортировки. Основные последовательные алгоритмы сортировки данных. Масштабирование параллельных вычислений.
37. Параллельные методы сортировки. Пузырьковая сортировка. Сортировка Шелла. Быстрая сортировка. Обобщенный алгоритм.
38. Многопоточное программирование с использованием классов MS Visual Studio .NET. Многопоточная обработка. Методы и свойства класса Thread. Фоновые потоки. Класс BackgroundWorker.
39. Многозадачность на основе класса Task. Фабрика задач. Обобщенные классы задач и фабрики задач Task TResult, TaskFactory TResult, методы и свойства классов.
40. Параллелизм задач. Параллелизм данных. Класс Parallel; методы Parallel.Invoke; Parallel.For; Parallel.Foreach.
41. Параллельные численные методы. Умножение матрицы на вектор. Произведение матриц (алгоритм Штрассена).
42. Параллельная реализация волнового алгоритма поиска кратчайшего пути в графе для случая неориентированного планарного графа.
43. Параллельная реализация быстрого преобразования Фурье.
44. Высокоуровневые средства программирования многопроцессорных систем. DVM-система. Общие сведения, цели создания, принципы построения.
45. Модель параллелизма, модель выполнения и модель программирования DVM.
46. Языки программирования DVM. Директивы DVM (на примере языка C-DVM).
47. Высокопроизводительные вычисления с применением графических процессоров (GPU). Архитектура современных GPU.
48. Технология NVidia CUDA. Модели и шаблоны программирования с использованием технологии CUDA. Модель памяти CUDA. Типы памяти.
49. Введение в вычисления общего назначения с использованием GPU. Основные архитектурные отличия GPU от CPU.

## 5.2. Фонд оценочных средств для проведения текущего контроля

Структура и содержание фонда оценочных средств представлены в Приложении 1 к рабочей программе дисциплины.

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

### 6.1. Рекомендуемая литература

#### 6.1.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год	Колич-во
ЛП.1	Калачев А. В.	Многоядерные процессоры: учебное пособие	Москва: Интернет-Университет Информационных Технологий, 2011	<a href="http://biblioclub.ru/">http://biblioclub.ru/</a> - неограниченный доступ для зарегистрированных пользователей
ЛП.2	Алексеев А. А.	Основы параллельного программирования с использованием Visual Studio 2010	Москва: Национальный Открытый Университет «ИНТУИТ», 2016	<a href="http://biblioclub.ru/">http://biblioclub.ru/</a> - неограниченный доступ для зарегистрированных пользователей
ЛП.3	Николаев Е. И.	Параллельные вычисления: учебное пособие	Ставрополь: СКФУ, 2016	<a href="http://biblioclub.ru/">http://biblioclub.ru/</a> - неограниченный доступ для зарегистрированных пользователей

#### 6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год	Колич-во
Л2.1	Левин М. П.	Параллельное программирование с использованием OpenMP: учебное пособие	Москва: Интернет-Университет Информационных Технологий, 2008	<a href="http://biblioclub.ru/">http://biblioclub.ru/</a> - неограниченный доступ для зарегистрированных пользователей
Л2.2	Антонов А. С.	Параллельное программирование с использованием технологии MPI: курс	Москва: Интернет-Университет Информационных Технологий, 2008	<a href="http://biblioclub.ru/">http://biblioclub.ru/</a> - неограниченный доступ для зарегистрированных пользователей
Л2.3	Биллинг В. А.	Параллельные вычисления и многопоточное программирование	Москва: Национальный Открытый Университет «ИНТУИТ», 2016	<a href="http://biblioclub.ru/">http://biblioclub.ru/</a> - неограниченный доступ для зарегистрированных пользователей

### 6.1.3. Методические разработки

	Авторы, составители	Заглавие	Издательство, год	Колич-во
Л3.1	Савельева Н. Г., Веретенникова Е. Г., Мирошниченко И. И., Гречкина В. Ю.	Работа в среде Visual Studio 2013 (C++): лаборатор. практикум по программированию	Ростов н/Д: Изд-во РГЭУ (РИНХ), 2014	53
Л3.2	Савельева Н. Г., Жилина Е. В.	Работа в среде Visual Studio 2013 (C++): лаборатор. практикум по объектно-ориентир. программированию	Ростов н/Д: Изд-во РГЭУ (РИНХ), 2014	63
Л3.3	Абрамян М. Э.	Практикум по параллельному программированию с использованием электронного задачника Programming Taskbook for MPI: учебное пособие	Ростов на Дону: Издательство Южного федерального университета, 2010	<a href="http://biblioclub.ru/">http://biblioclub.ru/</a> - неограниченный доступ для зарегистрированных пользователей

### 6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

Э1	Введение в методы параллельного программирования. Лекция 1. Введение. Презентация [Электронный ресурс] / М.:Национальный Открытый Университет «ИНТУИТ»,2014. -22с. - <a href="http://biblioclub.ru/index.php?page=book&amp;id=237055">http://biblioclub.ru/index.php?page=book&amp;id=237055</a>			
Э2	Введение в методы параллельного программирования. Лекция 2. Принципы построения параллельных вычислительных систем. Презентация [Электронный ресурс] / М.:Национальный Открытый Университет «ИНТУИТ»,2014. -61с. - <a href="http://biblioclub.ru/index.php?page=book&amp;id=237056">http://biblioclub.ru/index.php?page=book&amp;id=237056</a>			
Э3	Введение в методы параллельного программирования. Лекция 3. Моделирование и анализ параллельных вычислений. Презентация [Электронный ресурс] / М.:Национальный Открытый Университет «ИНТУИТ»,2014. - 60с. - <a href="http://biblioclub.ru/index.php?page=book&amp;id=237057">http://biblioclub.ru/index.php?page=book&amp;id=237057</a>			
Э4	Введение в методы параллельного программирования. Лекция 5. Параллельное программирование на основе MPI. Презентация [Электронный ресурс] / М.:Национальный Открытый Университет «ИНТУИТ»,2014. -58с. - <a href="http://biblioclub.ru/index.php?page=book&amp;id=237059">http://biblioclub.ru/index.php?page=book&amp;id=237059</a>			
Э5	Введение в методы параллельного программирования. Лекция 6. Параллельное программирование на основе MPI. Презентация [Электронный ресурс] / М.:Национальный Открытый Университет «ИНТУИТ»,2014. -51с. - <a href="http://biblioclub.ru/index.php?page=book&amp;id=237064">http://biblioclub.ru/index.php?page=book&amp;id=237064</a>			
Э6	Введение в методы параллельного программирования. Лекция 7. Параллельное программирование на основе MPI. Презентация [Электронный ресурс] / М.:Национальный Открытый Университет «ИНТУИТ»,2014. -33с. - <a href="http://biblioclub.ru/index.php?page=book&amp;id=237092">http://biblioclub.ru/index.php?page=book&amp;id=237092</a>			
Э7	Введение в методы параллельного программирования. Лекция 14. Параллельные методы решения дифференциальных уравнений в частных производных. Презентация [Электронный ресурс] / М.:Национальный Открытый Университет «ИНТУИТ»,2014. -69с. - <a href="http://biblioclub.ru/index.php?page=book&amp;id=237098">http://biblioclub.ru/index.php?page=book&amp;id=237098</a>			
Э8	Введение в методы параллельного программирования. Лекция 15. Параллельные методы многоэкстремальной оптимизации. Презентация [Электронный ресурс] / М.:Национальный Открытый Университет «ИНТУИТ»,2014. - 43с. - <a href="http://biblioclub.ru/index.php?page=book&amp;id=237167">http://biblioclub.ru/index.php?page=book&amp;id=237167</a>			
Э9	Введение в методы параллельного программирования. Лекция 16. Программная система ПараЛаб для изучения и исследования методов параллельных вычислений. Презентация [Электронный ресурс] / М.:Национальный Открытый Университет «ИНТУИТ»,2014. -57с. - <a href="http://biblioclub.ru/index.php?page=book&amp;id=237100">http://biblioclub.ru/index.php?page=book&amp;id=237100</a>			

### 6.3. Перечень программного обеспечения

6.3.1	MS Visual Studio
6.3.2	MS Visio



**6.4 Перечень информационных справочных систем**

6.4.1 Консультант +

6.4.2 Гарант

**7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)**


- 7.1 Помещения для проведения всех видов работ, предусмотренных учебным планом, укомплектованы необходимой специализированной учебной мебелью и техническими средствами обучения. Для проведения лекционных занятий используется демонстрационное оборудование. Лабораторные занятия проводятся в компьютерных классах, рабочие места в которых оборудованы необходимыми лицензионными программными средствами и выходом в Интернет.

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

Методические указания по освоению дисциплины представлены в Приложении 2 к рабочей программе дисциплины.

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Ростовский государственный экономический университет (РИНХ)»

Рассмотрено и одобрено  
на заседании кафедры Информационные технологии и  
защита информации

Протокол № 10 от 11.05.2018 г.  
Зав.кафедрой  Тищенко Е.Н.

## МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Инженерное программирование

Направление подготовки

09.03.04 Программная инженерия

Уровень образования

Бакалавриат

Составитель

(подпись)

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

Жилина Е.В., доцент, к.э.н.

Ростов-на-Дону, 2018

Методические указания по освоению дисциплины «Параллельное программирование» адресованы студентам всех форм обучения.

Учебным планом по направлению подготовки 09.03.04 «Программная инженерия» предусмотрены следующие виды занятий:

лекционные  
лабораторные

В ходе лекционных занятий рассматриваются основные теоретические вопросы, даются рекомендации для самостоятельной работы и подготовке к лабораторным занятиям.

При подготовке к лабораторным занятиям каждый студент должен:

- изучить рекомендованную учебную литературу;
- изучить конспекты лекций;
- подготовить ответы на все вопросы по изучаемой теме;
- письменно решить домашнее задание, рекомендованные преподавателем при изучении каждой темы.

По согласованию с преподавателем студент может подготовить реферат, доклад или сообщение по теме занятия. В процессе подготовки к лабораторным занятиям студенты могут воспользоваться консультациями преподавателя.

Вопросы, не рассмотренные на аудиторных занятиях, должны быть изучены студентами в ходе самостоятельной работы. Контроль самостоятельной работы студентов над учебной программой курса осуществляется в ходе занятий методом устного опроса или контрольной работы. В ходе самостоятельной работы каждый студент обязан прочитать основную и по возможности дополнительную литературу по изучаемой теме, дополнить конспекты недостающим материалом, выписками из рекомендованных первоисточников. Выделить непонятные термины, найти их значение в энциклопедических словарях.


Студент должен готовиться к предстоящим лабораторным занятиям по всем, обозначенным в рабочей программе дисциплины вопросам.

При реализации различных видов учебной работы используются разнообразные (в т.ч. интерактивные) методы обучения, в частности:

- интерактивная доска для подготовки и проведения лекционных занятий.

Для подготовки к занятиям, текущему контролю и промежуточной аттестации студенты могут воспользоваться электронной библиотекой ВУЗа <http://library.rsue.ru/>. Также обучающиеся могут взять на дом необходимую литературу на абонементе вузовской библиотеки или воспользоваться читальными залами вуза.

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Ростовский государственный экономический университет (РИНХ)»

Рассмотрено и одобрено  
на заседании кафедры Информационные  
технологии и защита информации  
Протокол № 10 от 11.05.2018 г.  
Зав.кафедрой  Тищенко Е.Н.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ  
ПО ДИСЦИПЛИНЕ**

Параллельное программирование

Направление подготовки

09.03.04 Программная инженерия

Уровень образования

Бакалавриат

Составитель



Жилина Е.В., доцент, к.э.н.

---

(подпись) Ф.И.О., должность, ученая степень, ученое звание

Ростов-на-Дону, 2018

## Оглавление

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.....	3
2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания.....	3
3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы.....	4
4. Методические материалы, определяющие процедуры оценивания результатов освоения образовательной программы .....	16

## 1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Перечень компетенций с указанием этапов их формирования представлен в п. 3. «Требования к результатам освоения дисциплины» рабочей программы дисциплины.

## 2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

### 2.1 Показатели и критерии оценивания компетенций:

ЗУН, составляющие компетенцию	Показатели оценивания	Критерии оценивания	Средства оценивания
<b>ПК 1 - готовностью применять основные методы и инструменты разработки программного обеспечения</b>			
З. инструменты и библиотеки разработки программного обеспечения с использованием параллельных вычислений	актуальность темы исследования и ее научно-практическая новизна	полнота и содержательность ответа умение приводить примеры	РЗ – расчетные задачи, ЛЗ – лабораторные задания, КР- контрольная работа
У. строить модель выполнения параллельных вычислений, оценивать эффективности параллельных вычислений с помощью инструментов разработки программного обеспечения	использование современных информационно-коммуникационных технологий	полнота и содержательность ответа умение приводить примеры умение самостоятельно находить решение поставленных задач	
В. навыками использования различных методов межпроцессного взаимодействия	использование современных информационно-коммуникационных технологий	полнота и содержательность ответа умение приводить примеры умение самостоятельно находить решение поставленных задач	
<b>ПК 3 - владением навыками использования различных технологий разработки программного обеспечения</b>			
З. основные подходы к разработке параллельных программ	актуальность темы исследования и ее научно-практическая новизна	полнота и содержательность ответа умение приводить примеры	РЗ – расчетные задачи, ЛЗ – лабораторные задания, КР- контрольная работа
У. использовать различные технологии разработки программного обеспечения с использованием параллельных вычислений	использование современных информационно-коммуникационных технологий	полнота и содержательность ответа умение приводить примеры умение самостоятельно находить решение поставленных задач	
В. навыками реализации параллельных программ с помощью различных технологий	использование современных информационно-коммуникационных технологий	полнота и содержательность ответа умение приводить примеры умение самостоятельно находить решение поставленных задач	

## 2.2 Шкалы оценивания:

Текущий контроль успеваемости и промежуточная аттестация осуществляется в рамках накопительной балльно-рейтинговой системы в 100-балльной шкале

6 семестр – очная форма обучения:

- 50-100 баллов («зачет») – изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса целями и задачами обучения; правильные, уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой; наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целями обучения, правильные действия по применению знаний на практике, четкое изложение материала, допускаются отдельные логические и стилистические погрешности, обучающийся усвоил основную литературу, рекомендованную в рабочей программе дисциплины; наличие твердых знаний в объеме пройденного курса в соответствии с целями обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению знаний на практике;

- 0-49 баллов («незачет») – ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

7 семестр – очная форма обучения, 5 курс (заочная форма обучения):

- 84-100 баллов (оценка «отлично») – изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса целями и задачами обучения; правильные, уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой;

- 67-83 баллов (оценка «хорошо») – наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целями обучения, правильные действия по применению знаний на практике, четкое изложение материала, допускаются отдельные логические и стилистические погрешности, обучающийся усвоил основную литературу, рекомендованную в рабочей программе дисциплины;

- 50-66 баллов (оценка «удовлетворительно») – наличие твердых знаний в объеме пройденного курса в соответствии с целями обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению знаний на практике;

- 0-49 баллов (оценка «неудовлетворительно») – ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

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

В разделе приводятся типовые варианты оценочных средств: вопросы к зачету, экзамену, лабораторные задания, расчетные задачи.

**Вопросы к зачету, 6 семестр (очная форма обучения)  
по дисциплине Параллельное программирование**

1. Основы MPI
2. Запуск MPI программ
3. Группы коммутаторов и процессов.
4. Сообщения MPI.
5. Коммутаторы.
6. Блокирующее поведение
7. Неблокирующее поведение. Непрокирующиеся вызовы
8. Способы коммуникации.
9. Синхронная коммуникация.
10. Коммуникация по готовности.
11. Буферизованная коммуникация.
12. Стандартизованная коммуникация.
13. Основные функции коллективной коммуникации.
14. Операция широкой рассылки (broadcast).
15. Операция сбора (gather).
16. Операция глобального приведения (global reduction).
17. Изменение кода параллельной программы.
18. Использование функции MPI\_Scatterv.
19. Использование функции MPI\_Scatter.
20. Использование функции MPI\_Gatherv.
21. Основные функции, необходимые для создания групп и коммутаторов/
22. Основные функции, необходимые для получения информации о коммутаторах.
23. Основные типы данных MPI.
24. Производные типы данных.
25. Общие типы данных и карты типов. Диапазон типа данных.

*Критерии оценивания:*

- 50-100 баллов («зачет») – изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса целями и задачами обучения; правильные, уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой; наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целями обучения, правильные действия по применению знаний на практике, четкое изложение материала, допускаются отдельные логические и стилистические погрешности, обучающийся усвоил основную литературу, рекомендованную в рабочей программе дисциплины; наличие твердых знаний в объеме пройденного курса в соответствии с целями обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению знаний на практике;

- 0-49 баллов («незачет») – ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

**Вопросы к экзамену, 7 семестр (очная форма обучения)  
по дисциплине Параллельное программирование**

1. Общая схема и методика разработки параллельных алгоритмов. Принцип «Разделяй-и-властвуй». Виды декомпозиции задачи: по данным, по коду, по выполнению, комбинированные. Примеры различных декомпозиций.



2. Сортировка, выпуклые оболочки, матричные операции: примеры декомпозиции рекурсивного вида.
3. Матричные операции: пример декомпозиции по данным, одно-, двух- и трехмерное разбиения.
4. Общая схема и методика разработки параллельных алгоритмов.
5. Оценка трудоемкости операций передачи данных для кластерных систем. Модель Хокни.
6. Базовые средства параллельного программирования систем с общей памятью. Стандарт OpenMP. Общие сведения. Структура стандарта OpenMP. Достоинства технологии OpenMP.
7. Модель параллелизма OpenMP. Модель памяти OpenMP. Директивы OpenMP. Типы директив. Формат записи директив.
8. OpenMP. Определение параллельной области. Распределение вычислений между потоками. Директивы синхронизации. Директивы управления областью видимости данных. Совместимость директив и их параметров.
9. Библиотека функций OpenMP. Функции для контроля/запроса параметров среды исполнения. Функции синхронизации. Переменные среды исполнения.
10. Сравнение технологий MPI и OpenMP. Гибридный (MPI+OpenMP) подход для SMP-кластеров. Инструментальные средства разработки и отладки многопоточных приложений.
11. Принципы распараллеливания сортировки. Основные последовательные алгоритмы сортировки данных. Масштабирование параллельных вычислений.
12. Параллельные методы сортировки. Пузырьковая сортировка. Сортировка Шелла. Быстрая сортировка. Обобщенный алгоритм.
13. Многопоточное программирование с использованием классов MS Visual Studio .NET. Многопоточная обработка. Методы и свойства класса Thread. Фоновые потоки. Класс BackgroundWorker.
14. Многозадачность на основе класса Task. Фабрика задач. Обобщенные классы задач и фабрики задач Task TResult, TaskFactory TResult, методы и свойства классов.
15. Параллелизм задач. Параллелизм данных. Класс Parallel; методы Parallel.Invoke; Parallel.For; Parallel.Foreach.
16. Параллельные численные методы. Умножение матрицы на вектор. Произведение матриц (алгоритм Штрассена).
17. Параллельная реализация волнового алгоритма поиска кратчайшего пути в графе для случая неориентированного планарного графа.
18. Параллельная реализация быстрого преобразования Фурье.
19. Высокоуровневые средства программирования многопроцессорных систем. DVM-система. Общие сведения, цели создания, принципы построения.
20. Модель параллелизма, модель выполнения и модель программирования DVM.
21. Языки программирования DVM. Директивы DVM (на примере языка C-DVM).
22. Высокопроизводительные вычисления с применением графических процессоров (GPU). Архитектура современных GPU.
23. Технология NVidia CUDA. Модели и шаблоны программирования с использованием технологии CUDA. Модель памяти CUDA. Типы памяти.
24. Введение в вычисления общего назначения с использованием GPU. Основные архитектурные отличия GPU от CPU.
25. Оптимизация CUDA-приложений. Модель исполнения CUDA. Компиляция CUDA-приложений.

*Критерии оценивания:*

- 84-100 баллов (оценка «отлично») – изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса целями и задачами обучения; правильные, уверенные

действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой;

- 67-83 баллов (оценка «хорошо») – наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целями обучения, правильные действия по применению знаний на практике, четкое изложение материала, допускаются отдельные логические и стилистические погрешности, обучающийся усвоил основную литературу, рекомендованную в рабочей программе дисциплины;

- 50-66 баллов (оценка удовлетворительно) – наличие твердых знаний в объеме пройденного курса в соответствии с целями обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению знаний на практике;

- 0-49 баллов (оценка неудовлетворительно) – ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

### **Вопросы к экзамену, 5 курс (заочная форма обучения) по дисциплине Параллельное программирование**

1. Основы MPI
2. Запуск MPI программ
3. Группы коммутаторов и процессов.
4. Сообщения MPI.
5. Коммутаторы.
6. Блокирующее поведение
7. Неблокирующее поведение. Непрокирующие вызовы
8. Способы коммуникации.
9. Синхронная коммуникация.
10. Коммуникация по готовности.
11. Буферизованная коммуникация.
12. Стандартизованная коммуникация.
13. Основные функции коллективной коммуникации.
14. Операция широкой рассылки (broadcast).
15. Операция сбора (gather).
16. Операция глобального приведения (global reduction).
17. Изменение кода параллельной программы.
18. Вспользование функции MPI\_Scatterv.
19. Использование функции MPI\_Scatter.
20. Использование функции MPI\_Gatherv.
21. Основные функции, необходимые для создания групп и коммутаторов/
22. Основные функции, необходимые для получения информации о коммутаторах.
23. Основные типы данных MPI.
24. Производные типы данных.
25. Общие типы данных и карты типов. Диапазон типа данных.
26. Общая схема и методика разработки параллельных алгоритмов. Принцип «Разделяй-и-властвуй». Виды декомпозиции задачи: по данным, по коду, по выполнению, комбинированные. Примеры различных декомпозиций.
27. Сортировка, выпуклые оболочки, матричные операции: примеры декомпозиции рекурсивного вида.
28. Матричные операции: пример декомпозиции по данным, одно-, двух- и трехмерное разбиения.
29. Общая схема и методика разработки параллельных алгоритмов.

30. Оценка трудоемкости операций передачи данных для кластерных систем. Модель Хокни.
31. Базовые средства параллельного программирования систем с общей памятью. Стандарт OpenMP. Общие сведения. Структура стандарта OpenMP. Достоинства технологии OpenMP.
32. Модель параллелизма OpenMP. Модель памяти OpenMP. Директивы OpenMP. Типы директив. Формат записи директив.
33. OpenMP. Определение параллельной области. Распределение вычислений между потоками. Директивы синхронизации. Директивы управления областью видимости данных. Совместимость директив и их параметров.
34. Библиотека функций OpenMP. Функции для контроля/запроса параметров среды исполнения. Функции синхронизации. Переменные среды исполнения.
35. Сравнение технологий MPI и OpenMP. Гибридный (MPI+OpenMP) подход для SMP-кластеров. Инструментальные средства разработки и отладки многопоточных приложений.
36. Принципы распараллеливания сортировки. Основные последовательные алгоритмы сортировки данных. Масштабирование параллельных вычислений.
37. Параллельные методы сортировки. Пузырьковая сортировка. Сортировка Шелла. Быстрая сортировка. Обобщенный алгоритм.
38. Многопоточное программирование с использованием классов MS Visual Studio .NET. Многопоточная обработка. Методы и свойства класса Thread. Фоновые потоки. Класс BackgroundWorker.
39. Многозадачность на основе класса Task. Фабрика задач. Обобщенные классы задач и фабрики задач Task TResult, TaskFactory TResult, методы и свойства классов.
40. Параллелизм задач. Параллелизм данных. Класс Parallel; методы Parallel.Invoke; Parallel.For; Parallel.Foreach.
41. Параллельные численные методы. Умножение матрицы на вектор. Произведение матриц (алгоритм Штрассена).
42. Параллельная реализация волнового алгоритма поиска кратчайшего пути в графе для случая неориентированного планарного графа.
43. Параллельная реализация быстрого преобразования Фурье.
44. Высокоуровневые средства программирования многопроцессорных систем. DVM-система. Общие сведения, цели создания, принципы построения.
45. Модель параллелизма, модель выполнения и модель программирования DVM.
46. Языки программирования DVM. Директивы DVM (на примере языка C-DVM).
47. Высокопроизводительные вычисления с применением графических процессоров (GPU). Архитектура современных GPU.
48. Технология NVidia CUDA. Модели и шаблоны программирования с использованием технологии CUDA. Модель памяти CUDA. Типы памяти.
49. Введение в вычисления общего назначения с использованием GPU. Основные архитектурные отличия GPU от CPU.
50. Оптимизация CUDA-приложений. Модель исполнения CUDA. Компиляция CUDA-приложений.

*Критерии оценивания:*

- 84-100 баллов (оценка «отлично») – изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса целями и задачами обучения; правильные, уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой;

- 67-83 баллов (оценка «хорошо») – наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целями обучения, правильные

действия по применению знаний на практике, четкое изложение материала, допускаются отдельные логические и стилистические погрешности, обучающийся усвоил основную литературу, рекомендованную в рабочей программе дисциплины;

- 50-66 баллов (оценка удовлетворительно) – наличие твердых знаний в объеме пройденного курса в соответствии с целями обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению знаний на практике;

- 0-49 баллов (оценка неудовлетворительно) – ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

### **Расчетные задачи по дисциплине Параллельное программирование**

Расчетная задача № 1

Постановка задачи распараллеливания существующих сериальных алгоритмов. Пример суммирования массива. Реализация параллельных алгоритмов с использованием процессов.

Расчетная задача № 2

Расчет относительной эффективности способов коммуникации.

Расчетная задача № 3

Основные функции коллективной коммуникации: операциями широкой рассылки (broadcast), сбора (gather) и глобального приведения (global reduction). Расчет параллельного волнового уравнения. Представление результатов расчета

Расчетная задача № 4

Программа вычисления числа  $\pi$ . Операции передачи данных между двумя процессами. Коллективные операции передачи данных.

Расчетная задача № 5

Управление группами процессов и коммутаторами. Модельный пример: умножение матрицы на вектор

Расчетная задача № 6

Разработка параллельных алгоритмов для параллельного решения гравитационной задачи N тел.

Расчетная задача № 7

Оценка трудоемкости операций передачи данных для кластерных систем.

Расчетная задача № 8

Пузырьковая сортировка.

Расчетная задача № 9

Сортировка Шелла.

Расчетная задача № 10

Быстрая сортировка.

Расчетная задача № 11.

Решение системы линейных алгебраических уравнений  $Ax = f$  методом простой итерации.

Расчетная задача № 12.

Решение системы линейных алгебраических уравнений  $Ax = f$  методом Гаусса.

Критерии оценки:

- 84-100 баллов (оценка «отлично») – изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса целями и задачами обучения; правильные, уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой;

- 67-83 баллов (оценка «хорошо») – наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целями обучения, правильные действия по применению знаний на практике, четкое изложение материала, допускаются отдельные логические и стилистические погрешности, обучающийся усвоил основную литературу, рекомендованную в рабочей программе дисциплины;

- 50-66 баллов (оценка удовлетворительно) – наличие твердых знаний в объеме пройденного курса в соответствии с целями обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению знаний на практике;

- 0-49 баллов (оценка неудовлетворительно) – ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

### **Лабораторные задания по дисциплине Параллельное программирование**

Лабораторное задание № 1

Запуск MPI программ. Компиляция. Настройки для сборки и счета. Запуск в пакете. Шесть базовых вызовов. Сообщения MPI. Коммуникаторы.

Лабораторное задание № 2

Распараллеливание метода наименьших квадратов.

Лабораторное задание № 3

Параллельные алгоритмы циклического сдвига на системе с распределенной памятью со структурой сети связей между процессами - “кольцо”, в которых каждый процесс инициирует рассылку своего сообщения одновременно в каком-либо выбранном направлении по кольцу (в MPI):

- с асинхронной блокированной передачей, блокированным приемом.
- с асинхронной неблокированной передачей, неблокированным приемом.
- с синхронной передачей, блокированным приемом.

Лабораторное задание № 4

Параллельные алгоритмы конвейерной передачи данных на системе с распределенной памятью со структурой сети связей между процессами - “линейка” (в MPI).

Лабораторное задание № 5

Параллельный алгоритм скалярного произведения векторов с использованием редуцированных операций (MPI\_Reduce(...), MPI\_Allreduce(...)).

Параллельный алгоритм скалярного произведения векторов с использованием парных взаимодействий (MPI\_Send(...), MPI\_Recv(...)).

Сравнить время выполнения алгоритмов (в MPI).

Лабораторное задание № 6

Параллельный алгоритм суммирования элементов векторов с использованием редуцированных операций (MPI\_Reduce(...), MPI\_Allreduce(...)) (в MPI).

Лабораторное задание № 7

Выполнить вычислительный эксперимент для оценки эффективности параллельных вычислений (обработка двумерного массива: перестановки элементов, умножение матриц и векторов) с

распараллеливанием вычислений с помощью технологии OpenMP. Варьировать размер массива от 10 до 10000, число потоков – 1,2,4,8.

#### Лабораторное задание № 8

Выполнить вычислительный эксперимент для оценки эффективности параллельных вычислений (алгоритмы сортировки) с распараллеливанием вычислений с помощью технологии OpenMP. Варьировать размер массива от 10 до 10000, число потоков – 1,2,4,8.

#### Лабораторное задание № 9

Выполнить вычислительный эксперимент для оценки эффективности параллельных вычислений (решение системы линейных алгебраических уравнений методом Гаусса) с распараллеливанием вычислений с помощью технологии многопоточности на основе классов .NET (Thread, Task, Parallel). Варьировать размер сетки от 100 до 10000 точек, число потоков – 1,2,4,8.

#### Лабораторное задание № 10

Параллельные алгоритмы скалярного произведения векторов:

- с использованием редуцированной операции (reduction(...)).
- с использованием директив critical{...} или atomic.

#### Лабораторное задание № 11.

Матрица M1 распределена по процессам горизонтальными ленточными полосами, матрица M2 распределена вертикальными ленточными полосами.

Матрицы M1 и M2 обе распределены по процессам горизонтальными ленточными полосами.

#### Лабораторное задание № 12.

Параллельный алгоритм умножения матрицы M на вектор V с использованием директивы распараллеливания параметрических циклов #pragma omp for.

Параллельный алгоритм умножения матрицы M на вектор V с использованием “ручного” задания работ (распараллеливания циклов).

Параллельный алгоритм умножения матрицы на матрицу с использованием директивы распараллеливания параметрических циклов #pragma omp for.

Сравнить показатели ускорения и эффективности приведенных алгоритмов, а так же сравнить их с аналогичными показателями этих же алгоритмов реализованных на системах с распределенной памятью.

#### Лабораторное задание № 13.

Параллельные алгоритмы строятся одновременно в MPI и OpenMP. Параллельные вычисления внутри узла вычислительного кластера осуществляются в OpenMP, а

взаимодействия процессов, расположенных в разных узлах, осуществляются в MPI: Параллельные алгоритмы вычисления определенного интеграла  $y=f(x)dx$  с использованием метода прямоугольников (трапеций) для  $f(x)=\sin(x)$ ,  $f(x)=x^2$ ,  $f(x)=e^x$

### 2. Методические рекомендации по выполнению лабораторных заданий

Лабораторные задания выполняются с учетом приобретенных знаний по предшествующим дисциплинам, теоретического материала дисциплины, с помощью и консультациями (при необходимости) преподавателя на занятиях.

### 3. Критерии оценки:

- 84-100 баллов (оценка «отлично») – изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса целями и задачами обучения; правильные, уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой;

- 67-83 баллов (оценка «хорошо») – наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целями обучения, правильные действия по применению знаний на практике, четкое изложение материала, допускаются отдельные логические и

стилистические погрешности, обучающийся усвоил основную литературу, рекомендованную в рабочей программе дисциплины;

- 50-66 баллов (оценка удовлетворительно) – наличие твердых знаний в объеме пройденного курса в соответствии с целями обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению знаний на практике;

- 0-49 баллов (оценка неудовлетворительно) – ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

### **Контрольная работа (заочная форма обучения) по дисциплине Параллельное программирование**

Целью контрольной работы является практическая реализация освоенных принципов разработки алгоритмов с параллельными вычислениями, а также приобретение обучающимися навыков тестирования, отладки и документирования разработанных проектов.

Типовая тематика контрольной работы:

1. Параллельные алгоритмы строятся одновременно в MPI и OpenMP. Параллельные вычисления внутри узла вычислительного кластера осуществляются в OpenMP, а взаимодействия процессов, расположенных в разных узлах, осуществляются в MPI: Параллельные алгоритмы вычисления определенного интеграла  $y=f(x)dx$  с использованием метода прямоугольников (трапеций) для  $f(x)=\sin(x)$ ,  $f(x)=x^2$ ,  $f(x)=e^x$

2. Параллельный алгоритм умножения матрицы  $M$  на вектор  $V$  с использованием директивы распараллеливания параметрических циклов `#pragma omp for`.

3. Параллельный алгоритм умножения матрицы  $M$  на вектор  $V$  с использованием “ручного” задания работ (распараллеливания циклов).

4. Параллельный алгоритм умножения матрицы на матрицу с использованием директивы распараллеливания параметрических циклов `#pragma omp for`.

5. Матрица  $M1$  распределена по процессам горизонтальными ленточными полосами, матрица  $M2$  распределена вертикальными ленточными полосами.

6. Матрицы  $M1$  и  $M2$  обе распределены по процессам горизонтальными ленточными полосами.

7. Параллельные алгоритмы скалярного произведения векторов: с использованием редуцированной операции (`reduction(...)`).

8. Параллельные алгоритмы скалярного произведения векторов: с использованием директив `critical{...}` или `atomic`.

9. Выполнить вычислительный эксперимент для оценки эффективности параллельных вычислений (алгоритмы сортировки) с распараллеливанием вычислений с помощью технологии OpenMP. Варьировать размер массива от 10 до 10000, число потоков – 1,2,4,8.

10. Выполнить вычислительный эксперимент для оценки эффективности параллельных вычислений (решение системы линейных алгебраических уравнений методом Гаусса) с распараллеливанием вычислений с помощью технологии многопоточности на основе классов .NET (Thread, Task, Parallel). Варьировать размер сетки от 100 до 10000 точек, число потоков – 1,2,4,8.

#### 2. Методические рекомендации по выполнению контрольной работы

Контрольная работа выполняется с учетом приобретенных знаний по предшествующим дисциплинам, теоретического материала дисциплины.

Основными этапами выполнения контрольной работы являются:

1. Постановка задачи.
2. Построение модели.
3. Разработка алгоритма.
4. Реализация алгоритма.
5. Проверка программы.

## 6. Составление отчета.

Постановка задачи. Это начальный этап и начинается он с ознакомления задачей и рекомендуемой литературой. Прежде чем решать задачу, необходимо ее точно сформулировать. Процесс точной формулировки задачи сводится к постановке правильных вопросов:

Понятна ли терминология, используемая в предварительной формулировке?

Что дано? Что нужно найти?

Как определить решение?

Каких данных не хватает и все ли они нужны?

Являются ли какие-то имеющиеся данные бесполезными? Какие сделаны допущения?

Возможны и другие вопросы в зависимости от конкретной задачи. Точную постановку задачи необходимо сформулировать в процессе консультации с преподавателем.

Построение модели. Задача четко поставлена, нужно сформулировать для нее математическую модель. Это очень важный шаг в процессе решения, и его надо хорошо обдумать. Выбор модели существенно влияет на остальные этапы в процессе решения. Большинство задач должно рассматриваться индивидуально.

Приступая к разработке модели, следует задать по крайней мере несколько основных вопросов:

Существует ли математическая величина, ассоциируемая с искомым результатом?

Какие математические структуры больше всего подходят для задачи?

Имеются ли какие-нибудь полезные отношения между объектами модели?

Существуют ли решенные аналогичные задачи?

Большинство решаемых задач, как правило, являются модификациями ранее решенных и для продвижения вперед приходится руководствоваться накопленным опытом.

Сделав выбор математической структуры, задачу следует переформулировать в терминах соответствующих математических объектов.

Разработка алгоритма. Как только задача четко поставлена и для нее построена модель, необходимо приступить к разработке алгоритма ее решения. Выбор метода разработки, сильно зависящий от выбора модели, может в значительной степени повлиять на эффективность алгоритма решения. Два разных алгоритма могут быть правильными, но очень сильно отличаться по эффективности. Доказательство правильности алгоритма — это один из наиболее трудных этапов создания алгоритма. Вероятно, наиболее распространенная процедура доказательства правильности программы — это прогон ее на разных тестах. Если выданные программой ответы могут быть подтверждены известными или вычисленными вручную данными, возникает вывод, что программа «работает». Однако этот метод редко исключает все сомнения; может существовать случай, в котором программа «не сработает».

Можно предложить следующую общую методику доказательства правильности алгоритма. Предположим, что алгоритм описан в виде последовательности шагов, скажем, от шага 0 до шага  $n$ . Необходимо предложить некое обоснование правомерности для каждого шага. В частности, может потребоваться лемма об условиях, действующих до и после пройденного шага. Затем необходимо предложить доказательство конечности алгоритма, при этом будут проверены все подходящие входные данные и получены все подходящие выходные данные.

Реализация алгоритма. Необходимо построить целую систему структур данных (классов) для представления важных аспектов используемой модели. Руководствуясь технологией ООП программа должна быть представлена совокупностью взаимодействующих объектов. Следует внимательно рассмотреть приведенные выше примеры с тем, чтобы попытаться применить разработанные в них классы в качестве прототипов для разработки путем наследования собственных объектов. При этом желательно исключить использование многих глобальных



переменных и строгую локализацию данных и действий рамками отдельных объектов, что является основой высокой надежности программы.

Проверка программы. Проверка программы может быть охарактеризована как экспериментальное подтверждение того факта, что программа делает именно то, что должна делать. Проверка программы является также экспериментальной попыткой установить границы использования алгоритма (проекта). Недостаточно доказать правильность алгоритма. Окончательная программа должна быть тщательно проверена и оттестирована. Как выбрать входные данные для тестирования? На этот вопрос невозможно дать общего ответа. Для любого алгоритма ответ зависит от сложности программы, имеющегося ресурса времени, а также от числа вводов (т. е. вариантов входных данных), для которых можно установить правильность выводов, и т.д. Обычно множество всех вводов огромно, и полная проверка практически невозможна. Необходимо выбрать множество вводов, которые проверяют каждый участок программы.

Составление текстовой части контрольной работы. Контрольная работа должна включать следующие разделы.

1. Содержание, включающее наименование всех разделов и пунктов с указанием номеров страниц.

2. Введение.

2.1 Дается характеристика предметной области, к которой относится решаемая задача и обосновывается ее актуальность.

2.2 Цель работы.

Формулируется цель выполнения контрольной работы.

2.3 Постановка задач.

В этом разделе требуется формализовать задачи, указать возможные ограничения на их решение, CASE-средства, ИТ-технологии и т.п.

3. Техническое задание.

В соответствии с вариантом формулируется задание по контрольной работы.

4. Теоретическая часть, освещающую теоретические аспекты темы;

5. Практическая часть, в которой разработка программы и ее результаты.

5.1 Анализ и разработка проекта (основной раздел отчета).

Должен отражать результаты анализа возможных вариантов решения задачи и выбора среди них наиболее рационального. Приводятся математические выкладки и рисунки, поясняющие зависимости параметров решения задачи от данных. Обосновывается выбор структур данных и основных операций над ними. Приводятся результаты моделирования программ-прототипов (если они использовались для разработки алгоритма). Определяются источники и форматы исходных данных и содержание вывода программы.

5.2 Алгоритм работы проекта.

В этом разделе приводится блок-схема алгоритма программы.

### 5.3 Текст программы.

Приводится текст программы с комментариями.

### 5.4 Результаты моделирования программы на ПК.

Должны быть приведены результаты тестирования программы с различными наборами данных, в том числе и с некорректными данными.

### 6. Выводы.

Приводятся комментарии к результатам и рекомендации к дальнейшему совершенствованию программы.

### 7. Использованная литература.

#### 8. Приложения

Включают материалы иллюстративного и вспомогательного характера (таблицы большого формата; дополнительные расчеты; распечатки и проч.) Приложения обозначаются русскими заглавными буквами - А, Б, В и т.д. (например, «Приложение А»), располагаются в виде заголовка, по центру.

Таблицы, рисунки, формулы оформляются в соответствии с внутривузовским изданием для нормоконтроля. На все таблицы, рисунки, литературные источники, приложения в тексте должны быть ссылки.

Оформление контрольной работы должно соответствовать требованиям государственных стандартов, в т.ч. и методических рекомендаций вуза (кафедры). Текст работы должен быть набран на белой бумаге формата А4 с одной стороны листа. Размер шрифта: 12-14, интервал: 1,5. Поля: левое – 30 мм, правое – 10 мм, верхнее – 20 мм, нижнее – 20 мм.

### 3. Критерии оценки контрольной работы:

- 50-100 баллов («зачет») – изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса целями и задачами обучения; правильные, уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой; наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целями обучения, правильные действия по применению знаний на практике, четкое изложение материала, допускаются отдельные логические и стилистические погрешности, обучающийся усвоил основную литературу, рекомендованную в рабочей программе дисциплины; наличие твердых знаний в объеме пройденного курса в соответствии с целями обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению знаний на практике;

- 0-49 баллов («незачет») – ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

#### **4. Методические материалы, определяющие процедуры оценивания результатов освоения образовательной программы**

Процедуры оценивания включают в себя текущий контроль и промежуточную аттестацию.

**Текущий контроль** успеваемости проводится с использованием оценочных средств, представленных в п. 3 данного приложения. Результаты текущего контроля доводятся до сведения студентов до промежуточной аттестации.

**Промежуточная аттестация** проводится в форме зачета, экзамена.

Зачет проводится по окончании теоретического обучения до начала экзаменационной сессии.

Экзамен проводится по расписанию экзаменационной сессии в устном виде. Количество вопросов в экзаменационном задании – 3. Объявление результатов производится в день экзамена.

Результаты аттестации заносятся в экзаменационную ведомость и зачетную книжку студента. Студенты, не прошедшие промежуточную аттестацию по графику сессии, должны ликвидировать задолженность в установленном порядке.