МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное

учреждение высшего образования «Южный федеральный университет»

(ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Параллельные методы вычислений на СуперЭВМ

Ростов-на-Дону – Таганрог 2020

Содержание

[I. Цели и задачи освоения дисциплины 4](#_Toc57195560)

[II. Место дисциплины в структуре образовательной программы 4](#_Toc57195561)

[III. Требования к результатам освоения дисциплины 5](#_Toc57195562)

[IV. Содержание и структура дисциплины 6](#_Toc57195563)

[4.1. Содержание дисциплины, структурированное по темам 6](#_Toc57195564)

[4.2. План внеаудиторной самостоятельной работы 7](#_Toc57195565)

[4.3. Содержание учебного материала 8](#_Toc57195566)

[V. Образовательные технологии 9](#_Toc57195567)

[VI. Учебно-методическое обеспечение дисциплины 9](#_Toc57195568)

[6.1. Основная литература 9](#_Toc57195569)

[6.2. Дополнительная литература 9](#_Toc57195570)

[6.3 Периодические издания 10](#_Toc57195571)

[6.4 Интернет-ресурсы 10](#_Toc57195572)

[VII.Материально-техническое обеспечение дисциплины 10](#_Toc57195573)

[7.1 Учебно-лабораторное оборудование 10](#_Toc57195574)

[7.2 Программные средства 10](#_Toc57195575)

[7.3 Технические и электронные средства 10](#_Toc57195576)

[VIII. Методические указания для обучающихся по освоению дисциплины 10](#_Toc57195577)

[IX. Учебная карта дисциплины 11](#_Toc57195578)

[X. Фонд оценочных средств 12](#_Toc57195579)

[10.1. Паспорт фонда оценочных средств 12](#_Toc57195580)

[10.2. Контрольная работа № 1 (тестирование) 12](#_Toc57195581)

[Контрольная работа № 2 (тестирование) 12](#_Toc57195582)

[10.6 Примеры индивидуальных заданий 15](#_Toc57195583)

# Цели и задачи освоения дисциплины

Цели освоения дисциплины:

ознакомить магистрантов с теоретическими основами создания параллельных алгоритмов, современными вычислительными средами и параллельными языками программирования;

сформировать компетенции, необходимые для ведения научно-исследовательской деятельности в области параллельного анализа и вычислительной техники;

ознакомить с методологическими основами параллельной обработки информации.

Задачи освоения дисциплины:

1. Углубленное изучение основных теоретических аспектов в области создания параллельных методов решения вычислительно-трудоемких задач.
2. Исследование существующего математического аппарата в области параллельных вычислений.
3. Изучение современных подходов к созданию математического обеспечения современной вычислительной техники.
4. Формирование компетенций, направленных на осуществление научной деятельности в области параллельного программирования, вычислительной техники.

# Место дисциплины в структуре образовательной программы

Учебная дисциплина «Параллельные методы вычислений на СуперЭВМ» к факультативным дисциплинам образовательной программы и изучается в третьем семестре магистратуры.

Для изучения данной дисциплины необходимы знания, умения и навыки, формируемые предшествующими элементами образовательной программы:

| Наименование  дисциплины (модуля), практики | Требуемые знания, умения, навыки |
| --- | --- |
| Численные методы | Знания:   * Основных формул, правил и методов решения задач вычислительной математики |
| Умения:   * Правильно выбрать и использовать математические методы |
| Навыки: |
| Высокопроизводительные вычислительные системы | Умения:   * Разбираться в классификации современных вычислительных средств |
| Навыки:   * языков программирования |

Знания, умения и навыки, формируемые данной дисциплиной, потребуются при освоении следующих элементов образовательной программы:

производственная практика (НИР), преддипломная практика и при подготовке ВКР (магистерской диссертации) по тематикам разработки алгоритмов и программного обеспечения для высокопроизводительных вычислительных систем.

.

# Требования к результатам освоения дисциплины

Освоение дисциплины направлено на формирование следующих компетенций в соответствии с образовательным стандартом и образовательной программой:

Перечень планируемых результатов обучения по дисциплине, соотнесённых с индикаторами достижения компетенций

| Компетенция | Индикаторы достижения компетенции | Результаты обучения |
| --- | --- | --- |
| **ПК-1**. Способен проводить научно-исследовательскую работу в области высокопроизводительных вычислительных систем, включая системы квантовой обработки информации | **ПК-1.1** *Осуществляет обработку и анализ научно-технической информации и результатов исследования в области высокопроизводительных вычислительных систем и квантовой обработки информации* | Знания:  методологических основ теории параллельных методов решения вычислительно-трудоемких задач |
| Умения:  – осуществлять обобщение научно-технической информации, анализ отечественного и зарубежного опыта с помощью теоретических исследований в области параллельного программирования;  – выполнять постановку и формализацию задач в области параллельных вычислений;  – использовать методологические основы современного параллельного анализа при решении научных проблем. |
| Навыки:   * проведения научных исследований в области параллельных вычислений и решения задач на графах и гиперграфах с использованием современных информационно-коммуникационных технологий. |
| **ПК-1.2** *Осуществляет управление результатами исследований в области высокопроизводительных вычислительных систем* | Знания:  - направлений развития и современных проблем информационных технологий;  - моделей современных методов оценки сложности алгоритмов, облачных вычислений, параллельных вычислений; |
| Умения:  проводить теоретические и экспериментальные исследования в области построения параллельных алгоритмов и параллельных программ. |
| Навыки:   * разработки и проведения новых методов научных исследований в области информатики и вычислительной техники. |

# Содержание и структура дисциплины

Трудоёмкость дисциплины составляет 5 зачётных единиц, 180 часов.

Форма промежуточной аттестации: зачёт

## Содержание дисциплины, структурированное по темам

| № п/п | Темы дисциплины | Семестр | Виды учебной работы и их трудоёмкость, часы  (в том числе с использованием онлайн-курсов) | | | | Наименования оценочных средств |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Контактная работа | | | Самостоя­тельная работа |
| Лекции | Практические занятия | Лабораторные занятия |
| **Модуль 1. Параллельные методы решения задач** | | | | | | | |
| 1 | Тема 1. Параллельные алгоритмы. Основные понятия. Меры параллелизма. Распараллеливание выражений. | 3 | 2 | 4 |  | 14 | * индивидуальное задание |
| 2 | Тема 2. Построение параллельных алгоритмов для матричных вычислений. Прямые методы. | 3 | 2 | 6 |  | 16 | * выполнение отчета |
| 3 | Тема 3. Построение параллельных алгоритмов для матричных вычислений. Итерационные методы | 3 | 2 | 4 |  | 14 | * тестирование |
| 4 | Тема 4. Параллельные алгоритмы для прикладных задач. Моделирование задач на многопроцессорных системах | 3 | 4 | 4 |  | 18 | * контрольная работа №1 |
| **Модуль 2. Параллельное программирование** | | | | | | | |
| 5 | Тема 5. Параллельное программирование на основе МРI. Некоторые современные языки программирования. | 3 | 2 | 6 |  | 16 | – собеседование |
| 6 | Тема 6. Задача конструирования параллельной программы. | 3 | 2 | 4 |  | 16 | – индивидуальное задание |
| 7 | Тема 7. Взаимодействующие процессы | 3 | 2 | 4 |  | 16 | – презентация |
| 8 | Тема 8. Программирование взаимодействующих процессов | 3 | 2 | 4 |  | 16 | – контрольная работа №2 |
| **Итого часов** | |  | **18** | **36** | **0** | **126** | **–** |

## План внеаудиторной самостоятельной работы

| № п/п | Темы дисциплины | Семестр | Вид самостоятельной работы | Сроки выполнения (нед.) | Затраты времени (часы) | Учебно-методическое обеспечение |
| --- | --- | --- | --- | --- | --- | --- |
| **Модуль 1. Параллельные методы решения задач** | | | | | | |
| 1 | Тема 1. Параллельные алгоритмы. Основные определения | 3 | * проработка и повторение материала лекционных занятий; | 1–3 | 20 | [2], [3], [5] |
| 2 | Тема 2. Построение параллельных алгоритмов | 3 | * проработка и повторение материала лекционных занятий; * подготовка презентации | 4–6 | 22 | [2], [6], [8] |
| 3 | Тема 3. Параллельные алгоритмы для прикладных задач | 3 | * проработка и повторение материала лекционных занятий;   **–** подготовка к контрольной работе №1 | 7**–**9 | 22 | [2], [5], [9] |
| **Модуль 2. Параллельное программирование** | | | | | | |
| 4 | Тема 4. Основы MPI | 3 | * проработка и повторение материала лекционных занятий; | 10**–12** | 20 | [1], [8] |
| 5 | Тема 5. Граф алгоритма и параллельные вычисления | 3 | * проработка и повторение материала лекционных занятий;   **–**  подготовка презентации | 13**–15** | 20 | [2], [4] |
| 6 | Тема 6. Операторы индивидуальных и групповых взаимодействий | 3 | * **–** проработка и повторение материала лекционных занятий;   **–** подготовка к контрольной работе №2 | 16-18 | 22 | [5], [7] |
| **Общая трудоёмкость самостоятельной работы по дисциплине** | | | | | **126** | **–** |

## Содержание учебного материала

**Модуль 1. Параллельные методы решения задач**

**Тема1. *Параллельные алгоритмы. Основные понятия. Меры параллелизма. Распараллеливание выражений.***

Этапы разработки параллельных алгоритмов.Некоторые требования к параллельным алгоритмам: внутренний параллелизм, минимизация обменов между процессорами, логическая простота алгоритмов.

Типы параллелизма данных: функциональный параллелизм; геометрический параллелизм; алгоритмический параллелизм; конвейерный параллелизм; «беспорядочный» параллелизм.

Параллельный алгоритм суммирования. Параллельные формы рекурсивных алгоритмов. Каскадное суммирование

**Тема 2*. Построение параллельных алгоритмов для матричных вычислений. Прямые методы.***

Параллельная реализация операций матричной алгебры (расчет скалярного произведение векторов, умножение матрицы на вектор с прибавлением вектора, умножения матрицы на матрицу и т.п.) на МВС с различной топологией связей между процессорами.

Параллельная реализация метода Гаусса и LU-разложения для решения систем общего вида на МВС с жесткой и универсальной коммутацией. Анализ эффективности.

Алгоритм Н.Н. Яненко – А.Н. Коновалова распараллеливания прогонки для трехдиагональных систем. Анализ эффективности. Распараллеливания алгоритма матричной прогонки.

Последовательный и параллельный алгоритмы циклической редукции (CR) для скалярных трехточечных уравнений. Анализ эффективности.

**Тема 3*. Построение параллельных алгоритмов для матричных вычислений. Итерационные методы.***

Параллельные алгоритмы итерационных методов Якоби и Зейделя решения линейных систем на МВС с жесткой и универсальной коммутацией. Анализ эффективности.

Параллельная реализация некоторых алгоритмов решения задач на собственные значения.

Параллельные методы сортировки. Сортировка Шелла. Быстрая сортировка.

**Тема4*. Параллельные алгоритмы для прикладных задач. Моделирование задач на многопроцессорных системах***

Некоторые вопросы прикладной математики, ориентированные на использование в высокопроизводительных вычислениях.

Решение одномерных задач динамики на многопроцессорной вычислительной системе. Решение двумерных задач механики на многопроцессорной вычислительной системе. Решение трехмерных задач механики на многопроцессорной вычислительной.

**Модуль 2. Параллельное программирование**

**Тема 5*. Параллельное программирование на основе МРI. Некоторые современные языки программирования.***

Основные понятия и определения. Введение в разработку параллельных программ с использованием МРI.

Языки для параллельных вычислений.

**Тема 6*. Задача конструирования параллельной программы.***

Представление алгоритма. Требования к алгоритму. Граф алгоритма и параллельные вычисления. Параллельные формы графа. Ярус и высота. Понятие параллельного алгоритма. Рекомендации для создания параллельных алгоритмов.

**Тема 7*. Взаимодействующие процессы***

Последовательные процессы. Выполнение системы процессов. Операторы индивидуальных взаимодействий. Операторы групповых взаимодействий. Схема «рандеву». Задача взаимного исключения. Дедлоки. Задача о пяти обедающих философах. Задача производитель-потребитель.

**Тема 8*. Программирование взаимодействующих процессов***

Асинхронное программирование. Некорректное вычисление данных. Реализация управления взаимодействующими процессами. Семафоры. Задача взаимного исключения. Задача производитель-потребитель с ограниченным буфером.

# Образовательные технологии

По дисциплине предусмотрены следующие методы обучения и интерактивные формы проведения занятий:

**Наименование тем занятий с использованием активных форм обучения:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Тема занятия | Вид занятия | Форма / Методы интерактивного обучения | Кол-во часов |
| 1 | Все разделы | Лекции | лекции-визуализации, собеседования | 18 |
| Итого часов | | | | 18 |



Наряду с традиционными образовательными технологиями, для реализации дисциплины могут использоваться технологии электронного обучения и дистанционные образовательные технологий в электронной информационно-образовательной среде Южного федерального университета. Лекционные занятия и другие формы контактной работы обучающихся с преподавателем могут проводиться с использованием платформ Microsoft Teams, Cisco, Moodle (BigBlueButton) и др., что позволяет обеспечить онлайн и офлайн взаимодействие преподавателя с обучающимися в рамках дисциплины.

Основными методами текущего контроля являются электронный учёт и контроль учебных достижений студентов (использование средств сервиса балльно-рейтинговой системы; ведение электронного журнала успеваемости, проведение электронного тестирования и применение других средств контроля с использованием системы электронного обучения).

# Учебно-методическое обеспечение дисциплины

6.1. Основная литература**:**

1.Богачев К.Ю. Основы параллельного программирования. – М: Бином. Лаборатория знаний,2003. - 344 с.

2. Геркель В.П. Теория и практика параллельных вычислений. – М: Интуит,2016.-500 с.

3. Голуб Дж, Ван Лоун Ч. Матричные вычисления: Пер. с англ. – М.: Мир,1999.-548 с.

4. Миллер Р., Боксер Л. Последовательные и параллельные алгоритмы: Общий подход.-М: Бином. Лаборатория знаний,2006.-406с.

6.2. Дополнительная литература**:**

5. Воеводин В.В. Модели и методы в параллельных процессах. – М.: Наука,1986.

6. Дацюк В.Н. и др. Методическое пособие по курсу «Многопроцессорные системы и параллельное программирование». Ростов-на-Дону: РГУ,2000.

7. Корнеев. В.В. Параллельные вычислительные системы. М.: Нолидж,1999.

8. Параллельные вычисления. Под ред. Г.Родрига.-М.: Наука. Гл.ред.физ.-мат.лит.,1986.-376 с.

9. Системы параллельной обработки. Под ред.Д.Ивенса.- М.: Мир,1985.-416 с.

## Периодические издания

- Научный журнал «Известия ЮФУ. Технические науки»;

- Журнал «Параллельные вычислительные технологии». Труды конференций.

## Интернет-ресурсы

Информационно-аналитические материалы по параллельным вычислениям:

http://www/parallel.ru

# VII.Материально-техническое обеспечение дисциплины

## 7.1 Учебно-лабораторное оборудование

Лекционные занятия проводятся в аудиториях, оснащенных интерактивной доской типа SmartBoard (или иным аналогичным по функциональному назначению оборудованием с возможностью подключения ноутбука и удаленным Wi-Fi доступом).

## Программные средства

Для подготовки и демонстрации презентаций по курсу используются стандартные средства Microsoft Office (Microsoft Power Point).

## 7.3 Технические и электронные средства

Необходимая техника: проектор (электронная доска), компьютеры соответствующей комплектации с выходом в Интернет установлены в аудиториях института.

# VIII. Методические указания для обучающихся по освоению дисциплины

Дисциплина «Параллельные методы вычислений на СуперЭВМ» читается в 3-м семестре. Процесс обучения дисциплине включает в себя аудиторные занятия (лекции), лабораторные занятия и самостоятельную работу. Лектор контролирует посещение всех видов аудиторных занятий.

Используется интерактивный метод обучения, так как все лекции представлены в виде презентаций, что позволяет обучающимся конспектировать необходимую информацию, не перегружая их дополнительными сведениями.

При проверке знаний обучающихся преподавателем проводятся собеседования, включающие опрос обучающихся в устной форме. Самостоятельная работа обучающихся

направлена на решение следующих задач:

1) разработка алгоритмов для параллельной реализации программ;

2) создание программного обеспечения для многопроцессорных систем;

3) развитие и совершенствование творческих способностей при самостоятельном проведении научной деятельности.

На самостоятельную работу выносятся следующие виды деятельности: проработка лекций и подготовка к собеседованиям – чтение конспекта лекций, профессиональной литературы, периодических изданий;

Итоговый контроль усвоения материала осуществляется в форме зачета в конце семестра.

Для успешного усвоения курса необходимо своевременное посещение лекционных занятий и активная работа на семинарских занятиях.

# IX. Учебная карта дисциплины

Курс 2, семестр 3, очная форма обучения

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Виды контрольных мероприятий  (наименования оценочных средств) | Количество баллов | |
| Текущий контроль | Рубежный контроль |
| **Модуль 1. Параллельные методы решения задач** | | | |
| 1 | Индивидуальное задание | 30 | – |
| 2 | Контрольная работа № 1 (тестирование) | – | 20 |
| 3 |  |  |  |
| **Модуль 2. Параллельное программирование** | | | |
| 4 | Индивидуальное задание | 30 | – |
| 5 | Контрольная работа № 2 (тестирование) | – | 20 |
|  |  |  |  |
| **Всего** | | **60** | **40** |
| Бонусные баллы | | **до 10 баллов**  За академическую активность на лекционных занятиях | |
| **Промежуточная аттестация  в форме зачета** | | **Зачет по результатам текущего и рубежного контроля** | |

# Фонд оценочных средств

## Паспорт фонда оценочных средств

| № п/п | Индикатор достижения компетенции | Наименование оценочного средства |
| --- | --- | --- |
| 1 | **ПК-1.1** *Осуществляет обработку и анализ научно-технической информации и результатов исследования в области высокопроизводительных вычислительных систем и квантовой обработки информации (ТФ.В/02.6)* | * контрольная работа № 1 (тестирование); * вопросы зачета |
| 2 | **ПК-1.2** *Осуществляет управление результатами исследований в области высокопроизводительных вычислительных систем (ТФ.С/02.6)* | * контрольная работа № 2 (тестирование); * вопросы зачета |

## Контрольная работа № 1 (тестирование)

Построить и исследовать следующие параллельные алгоритмы для системы

алгебраических уравнений размерностью 10 строк на 10 столбцов:

Варианты заданий:

* + - 1. Параллельная реализация метода Гаусса.
      2. Параллельная реализация LU-разложения.
      3. Метод параллельной прогонки.
      4. Метод циклической редукции.
      5. Параллельный алгоритм метода Якоби.
      6. Параллельный алгоритм метода Зейделя.
      7. Параллельный алгоритм сортировки Шелла.

Матрицы строятся с помощью программы Mathcad.

## Контрольная работа № 2 (тестирование)

Варианты заданий.

**Вариант 1**

Используя функции асинхронной передачи данных. Написать программы и:

А) Построить графики зависимости времени передачи сообщения между парами узлов, в зависимости от объема сообщения и числа задействованных узлов в передаче. Объем сообщений изменяется в пределах [4 слова: 4 мегаслова]. Шаг выбрать самостоятельно.

Б) Построить график зависимости времени создания динамического массива данных типа double от числа элементов этого массива на одном узле кластера. Элементам массива присваиваются случайные значения. Определить дисперсию для полученного массива случайных чисел максимального объема в оперативной памяти.

В) Попытаться Уменьшить суммарное время выполнения на кластере заданий А) и Б) за счет использования функций асинхронной передачи данных (MPI\_Isend())

**Вариант 2**

Используя функции асинхронной передачи данных. Написать программы и:

А) Построить графики зависимости времени передачи сообщения от одного узла ко всем узлам, в зависимости от объема сообщения и числа задействованных узлов в передаче. Объем сообщений изменяется в пределах [4 слова: 4 мегаслова]. Шаг выбрать самостоятельно.

Б) Реализовать алгоритм Паркенсона для транспонирования матрицы распределенной в памяти МВС. Определить эффективность этого метода. Рассчитать необходимое число пересылок и объем каждой пересылки. Размеры матрицы NxN. N={0,125; 0,25; 0,5; 1; 2}\*10^4

В) Скалярное произведение векторов длинной {0,125;0,25;0,5;1;2;4}\*10^8

**Вариант 3**

Используя функции асинхронной передачи данных. Написать программы и:

А) Построить графики зависимости времени передачи сообщения от всех узлов к одному узлу узлам, в зависимости от объема сообщения и числа задействованных узлов в передаче. Объем сообщений изменяется в пределах [4 слова: 4 мегаслова]. Шаг выбрать самостоятельно.

Б) Реализовать алгоритм умножения матрицы на вектор на МВС. Определить вычислительную мощность узлов МВС. Результат представить в MFlops. Размер матрицы 512х512; 1024х1024; 2048х2048; 4096х4096; 8192х8192; 16384х16384

В) Сложение векторов векторов длинной {0,125;0,25;0,5;1;2;4}\*10^8

**Вариант 4**

Используя функции синхронной передачи данных. Написать программы и:

А) Построить графики зависимости времени передачи сообщения между парами узлов, в зависимости от объема сообщения и числа задействованных узлов в передаче. Объем сообщений изменяется в пределах [4 слова: 4 мегаслова]. Шаг выбрать самостоятельно.

Б) Сложение векторов векторов длинной {0,125;0,25;0,5;1;2;4}\*10^8

В) Попытаться Уменьшить суммарное время выполнения на кластере заданий А) и Б) за счет использования функций асинхронной передачи данных (MPI\_Isend())

**Вариант 5**

Используя функции синхронной передачи данных. Написать программы и:

А) Построить графики зависимости времени передачи сообщения от одного узла ко всем узлам, в зависимости от объема сообщения и числа задействованных узлов в передаче. Объем сообщений изменяется в пределах [4 слова: 4 мегаслова]. Шаг выбрать самостоятельно.

Б) Реализовать алгоритм Паркенсона для транспонирования матрицы распределенной в памяти МВС. Определить эффективность этого метода. Рассчитать необходимое число пересылок и объем каждой пересылки. Размеры матрицы NxN. N={0,125; 0,25; 0,5; 1; 2}\*10^4

В) Реализовать алгоритм умножения матрицы на вектор на МВС. Определить вычислительную мощность узлов МВС. Результат представить в MFlops. Размер матрицы 512х512; 1024х1024; 2048х2048; 4096х4096; 8192х8192; 16384х16384

**Вариант 6**

Используя функции синхронной передачи данных. Написать программы и:

А) Построить графики зависимости времени передачи сообщения от всех узлов к одному узлу узлам, в зависимости от объема сообщения и числа задействованных узлов в передаче. Объем сообщений изменяется в пределах [4 слова: 4 мегаслова]. Шаг выбрать самостоятельно.

Б) Построить график зависимости времени создания динамического массива данных типа double от числа элементов этого массива на одном узле кластера. Элементам массива присваиваются значения от функции randomize. Определить мат-ожидание для полученного массива случайных чисел максимального объема в оперативной памяти.

В) Скалярное произведение векторов длинной {0,125;0,25;0,5;1;2;4}\*10^8

**Критерии оценки контрольной работы:**

- оценка 9-10 баллов за работу («отлично») выставляется, если студент полностью и правильно решил задание;

- оценка 7-8 балла («хорошо») – если студент решил не менее 70 процентов задания;

- оценка 5-6 баллов («удовлетворительно») – выставляется при решении не менее 60 процентов задания;

- менее 5 баллов («неудовлетворительно») – если студент не ответил на вопросы.

**10.3 Тематика практических занятий по разделам и темам**

1.Расспараллеливание арифметических выражений.

2. Моделирование и анализ параллельных вычислений

3. Алгоритм скалярного произведения векторов.

4.Параллельные методы умножения матрицы на вектор.

5. Параллельные методы умножения матрицы на матрицу.

6.Параллельное решение систем линейных уравнений.

7. Распараллеливание сплайновых конструкций.

8. Параллельные методы сортировки.

9.Решение задач с помощью параллельной прогонки.

10. Решение задач с помощью алгоритма параллельной циклической редукции.

11. Параллельные алгоритмы итерационных методов Якоби и Зейделя.

12.Параллельное решение уравнений в частных производных.

13.Параллельные алгоритмы для задач аэродинамики.

14.Оценка эффективности прямых параллельных методов.

15. Языки параллельного программирования.

16. Методы распараллеливания вложенных циклов.

17. Языки с явным заданием параллелизма.

18. Языки с неявным заданием параллелизма.

**10.4 Методические рекомендации к практическим занятиям**

Практические занятия нацелены на закрепление теоретических знаний, развитие и формирование практических навыков и умений по курсу.

Рекомендации:

- при подготовке к практическому занятию необходимо руководствоваться содержанием тематического плана в рабочей программе дисциплины,

- при подготовке к соответствующему практическому занятию необходимо по лекционному конспекту просмотреть практические задания.

Необходимо: выполнять домашние задания, выдаваемые преподавателем в течение семестра.

10.5 Список вопросов к зачету по дисциплине «Параллельные методы вычислений на СуперЭВМ»

1. Понятие параллельного алгоритма. Основные требования, предъявляемые к параллельным алгоритмам.

2. Распараллеливание арифметических выражений. Модель вычислений в виде графа.

3. Параллельные методы в линейной алгебре. Умножение матрицы на вектор. Матричное умножение.

4. Алгоритм Фокса умножения матриц при блочном разделении данных.

5. Алгоритм Кэннона умножения матриц при блочном разделении данных

6. Параллельные алгоритмы решения СЛАУ прямыми методами.

7. Вычисление эффективности построенных алгоритмов.

8.Модифицированная каскадная схема. Вычисление всех частных сумм.

9.Оценки максимально достижимого параллелизма, Закон Амдаля. Анализ масштабируемости параллельных вычислений.

10. Параллельные алгоритмы решения СЛАУ итерационными методами.

11. Информационно логические задачи. Задачи сортировки.

12. Параллельные методы решения дифференциальных уравнений в частных производных.

13. Решения некоторых задач аэродинамики на матричных ЭВМ.

14.Библиотека MPI. Операции обмена типа точка-точка. Основные функции, их разновидности.

15.Коллективные операции. Функции MPI\_BARRIER, MPI\_BCAST, MPI\_GATHER, MPI\_GATHERV.

16.Коллективные операции. Функции MPI\_SCATTER, MPI\_SCATTERV, MPI\_ALLGATHER, MPI\_ALLGATHERV.

17.Коллективные операции. Функции MPI\_ALLTOALL, MPI\_ALLREDUCE, MPI\_REDUCE.

18. Языки параллельного программирования. Язык ОККАМ. Отечественные языки.

19. Операторы индивидуального и группового взаимодействия.

20. Методы распараллеливания вложенных циклов. Подходы для построения ветвей алгоритма, основанные на преобразовании цикла.

**Критерии оценки ответа магистранта:**

оценка «отлично» выставляется, если магистрант дал исчерпывающий ответ на все вопросы, не допустил ошибок, продемонстрировал понимание междисциплинарных связей, знание специальной литературы и дополнительных источников информации;

оценка «хорошо» – если магистрант раскрыл содержание вопросов, но есть незначительные ошибки при ответе на вопросы, которые он легко исправляет с помощью преподавателя, на дополнительные вопросы преподавателя ответил частично;

оценка «удовлетворительно» – обучающийся частично раскрыл содержание вопросов, или есть существенные замечания по содержанию ответа, при этом частично ответил на дополнительные вопросы преподавателя;

«неудовлетворительно» – обучающийся не раскрыл содержание вопросов и не ответил на дополнительные вопросы преподавателя.

## 10.6 Примеры индивидуальных заданий

**1. Асинхронный язык**

Цель проекта: студентам предлагается спроектировать и реализовать систему программирования на асинхронном языке. Результатом работы будет являться транслятор с асинхронного языка + исполнительная система асинхронных программ. После реализации транслятора и исполнительной системы, с их использованием будут решаться классические задачи параллельного программирования: задача "читатели/писатели", задача разделения множеств Дейкстры.

Краткое описание предметной области: Асинхронная модель вычислений является одной из наиболее общих моделей описания вычислений на мультикомпьютере и/или мультипроцессоре. Асинхронная программа определяет систему независимо исполняющихся и взаимодействующих процессов. Формулировка задачи в терминах асинхронного программирования позволяет описать весь внутренний параллелизм задачи, поскольку в общем случае в асинхронной программе отсутствуют ограничения на порядок выполнения операций и число одновременно исполняемых операций, обусловленные особенностями оборудования.

Чему научатся участники в процессе работы над проектом:

работать со строками и файлами в С/С++;

использовать Lex/YACC для создания трансляторов;

изучат алгоритмы планирования, используемые в современных операционных системах.

**2. Mini-MPI**

Цель проекта: студентам предлагается спроектировать и реализовать свою библиотеку, поддерживающую некоторый урезанный набор функций стандарта MPI. В качестве результата работы должна быть представлена собственно сама библиотека и некоторый модуль для запуска параллельной задачи на удаленном мультикомпьютере. Для проверки работоспособности библиотеки будет взята типовая задача пользователя, распараллеленная с помощью MPI.

Краткое описание предметной области: MPI (message passing interface) - это наиболее распространенный в мире промышленный стандарт на средства параллельных вычислений. Реализация MPI - это библиотека функций, обеспечивающая взаимодействие параллельных процессов с помощью механизма передачи сообщений. Программа, использующая MPI, легче отлаживается (сужается простор для совершения стереотипных ошибок параллельного программирования) и быстрее переносится на другие платформы (в идеале, простой перекомпиляцией).

Чему научатся участники в процессе работы над проектом? В процессе работы студенты разберутся и освоят следующие технологии и разделы программирования:

основы параллельного программирования;

распараллеливание методом передачи сообщений (MPI);

сетевое программирование (сокеты, TCP, технологию клиент-сервер);

многопоточное программирование;

обмен данными между различными потоками через разделяемую память (синхронизация, избежание deadlock).

**3. Введение в Grid технологии**

Цель проекта: студентам предлагается изучить основные аспекты grid технологий, освоить инструментальные средства пакета GlobusTookit, и реализовать программы для запуска, остановки параллельных расчетов, проверки состояния задачи пользователя в метакомпьютере и т.д.

Краткое описание предметной области: в настоящее время в мире GRID технологий стандартом де-факто для программирования является пакет Globus Toolkit. На основе Globus Toolkit уже сконструировано множество метакомпьютеров. Одним из крупных проектов использующих Globus Toolkit является проект CrossGrid, программное обеспечение которого установлено на кластере отдела МО ВВС.

Чему научатся участники в процессе работы над проектом:

изучат основные программные компоненты и функции, которые они выполняют: Систему безопасности: аутентификация и авторизация пользователей; Язык спецификации задачи;

систему управления ресурсами: сетевой доступ к ресурсам;

систему управления данными: сетевой доступ к файлам;

изучат основные GRID сервисы на примере CrossGrid, схема CrossGrid: роли компьютеров; запуск задач: брокер ресурсов: edg-job-submit; Система мониторинга: ocm-g, cg-gpm.

**4. Модельные численные задачи**

Цель: реализовать схемы решения уравнения переноса, сделать параллельную реализацию схем, написать программу визуализации.

Описание: численные методы решения уравнения переноса, методы параллельной реализации численных схем, средства визуализации средствами Win32API.

Научатся: познакомятся с конечно-разностными схемами решения диф. уравнений, параллельная реализация этих схем, визуализация средствами Win32API.

**5. Численное моделирование**

Цель: дать студентам начальное представление о численных методах.

В рамках этого проекта будет рассмотрено две довольно широко распространенные задачи: решение уравнения газовой динамики и уравнения Пуассона.

Во время работы студентам будут рассказаны основные методы решения уравнения Пуассона методом конечных разностей. А также студенты познакомятся с задачами газовой динамики и трудностями, которые возникают при решении уравнений газовой динамики.

**6. Моделирование процессов коагуляции**

Коагуляция - это процесс сцепления мелких частиц для образования более крупных. Для описания процесса коагуляции используется сложная модель, содержащая интегро-дифференциальные уравнения. При таком моделировании частицы описываются спектром масс, над которым проводятся специальные преобразования на каждом временном шаге. В итоге система эволюционирует до образования одной массивной суперчастицы, по массе равной суммарной массе вещества.

Цель: предлагается написать программное обеспечение для простого моделирования процессов коагуляции и их визуализации, чтобы на практике познакомиться с идеями прямого математического моделирования в физике и других дисциплинах, а также изучить простые средства для визуализации при таком моделировании