



Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Дагестанский государственный технический университет»

РЕКОМЕНДОВАНО
К УТВЕРЖДЕНИЮ:
Декаан, факультета
магистерской подготовки


Ашуралиева Р.К.
Подпись Ф.И.О.
20.09 2018г.

УТВЕРЖДАЮ:
Проректор по учебной работе,
председатель методического
совета ДГТУ


Н.С.Суракатов
Подпись Ф.И.О.
24.09 2018г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЬ)

Дисциплина М1.В.ОД.2 Теория параллельных вычислений
наименование дисциплины по ООП и код по ФГОС

для направления 09.04.04 «Программная инженерия»
шифр и полное наименование направления (специальности)

по профилю «Разработка программно-информационных систем»

факультет магистерской подготовки

кафедра Программное обеспечение вычислительной техники и автоматизированных систем

наименование кафедры, за которой закреплена дисциплина

Квалификация выпускника магистр

Форма обучения очная, курс 2, семестр 3;
очная, заочная, др.


Всего трудоемкость в зачетных единицах (часах) 5 (180 ч.);

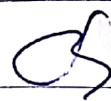
лекции - 9 (ч); экзамен - 3 (1 ЗЕТ-36ч.);
(семестр)

практические занятия - (час); зачет - ;
(семестр)

лабораторные занятия – 17 (ч); самостоятельная работа 118 (ч);

курсовой проект (работа, РГР) - 3 сем;

Зав. кафедрой 
подпись

Начальник УО 
подпись

В.Б. Мелехин
Ф.И.О


Э.В. Магомаева
Ф.И.О




Рабочая программа составлена в соответствии с требованиями ФГОС ВО по направлению 09.04.04 «Программная инженерия» магистерская программа «Разработка программно-информационных систем»

Программа одобрена на заседании выпускающей кафедры «Программное обеспечение вычислительной техники и автоматизированных систем»

Зав. выпускающей кафедрой по данному направлению (профилю)


_____ (Мелехин В.Б.)

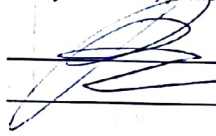
ОДОБРЕНО
Методической комиссией
по укрупненным группам специальностей и направлений
подготовки
09.00.00 Программная инженерия»
Председатель МК


_____ А.М.Абдулгалимов
Подпись, ФИО

12.09.2018г.

АВТОР ПРОГРАММЫ:

Магомедов И.А. к.т.н., доц.
Ф.И.О уч. степень, ученое звание, подпись



1. Цели освоения дисциплины

В результате изучения дисциплины студент должен приобрести:

- знания о принципах формализации предметной области программного проекта и разработки спецификации для компонентов программного продукта, формальном представлении алгоритмов и программ, алгоритмов параллельной обработки, средств их представления, методов отображения алгоритмов на регулярные матричные структуры, методов отображения матричных структур в среду процессорных элементов; устройства ЭВМ и систем; технологии распределенной обработки данных; инструментальных средствах исследования объектов профессиональной деятельности; технологии программирования параллельных программ.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, самостоятельная работа студента, консультации.

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости в форме выполнения лабораторных работ и отчетов по ним, рубежный контроль в форме коллоквиумов и промежуточный контроль в форме экзамена.

Общая трудоемкость освоения дисциплины составляет 5 зачетных единицы, 180 часов.

2. Место дисциплины в структуре ООП

Дисциплина М1.В.ОД.2 «Теория параллельных вычислений» относится к вариативной части обязательных дисциплин учебного плана. Программа базируется на дисциплинах:

машинно-зависимые языки программирования, сети и телекоммуникации;

Знания, полученные в результате изучения этой дисциплины, будут использоваться при работе над диссертационной работой магистра

Основными видами рубежного контроля знаний является экзамен.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины «Теория параллельных вычислений»

Процесс изучения дисциплины направлен на формирование следующих компетенций:

- способностью воспринимать математические, естественнонаучные, социально-экономические и профессиональные знания, умением самостоятельно приобретать, развивать и применять их для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте (ОПК-1);
 - культурой мышления, способностью выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных их разных областей науки и техники, выносить суждения на основании неполных данных (ОПК-2);
 - способностью проектировать распределенные информационные системы, их компоненты и протоколы их взаимодействия (ПК-7);
 - способностью проектировать системы с параллельной обработкой данных и
 - высокопроизводительные системы, и их компоненты (ПК-8);
- владением навыками программной реализации систем с параллельной обработкой данных и высокопроизводительных систем (ПК-14);

В результате освоения дисциплины студент должен:

- **Знать:**

- основные направления развития высокопроизводительных компьютеров;
- основные классификации многопроцессорных вычислительных систем;
- основные подходы к разработке параллельных программ
- основные технологии и модели параллельного программирования.
- методы параллельных вычислений для задач вычислительной математики (матричные вычисления, решение систем линейных уравнений, сортировка, обработки графов, уравнения в частных производных, многоэкстремальная оптимизация);

• **Уметь:**

- создавать параллельные программы для вычислительных систем с распределенной, общей оперативной памятью;
- проводить распараллеливание вычислительных алгоритмов;
- строить модель выполнения параллельных программ;
- оценивать эффективности параллельных вычислений;
- анализировать сложность вычислений и возможность распараллеливания разрабатываемых алгоритмов;
- применять общие схемы разработки параллельных программ для реализаций собственных алгоритмов;
- оценивать основные параметры получаемых параллельных программ, таких как ускорение, эффективность и масштабируемость.

• **Владеть:**

- создание параллельных программ для вычислительных систем с распределенной, общей оперативной памятью;
- построение параллельных аналогов вычислительных алгоритмов.

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 5 зачетных единиц – 180 часов; в том числе - 9 лекций; 17- лабораторных; СРС - 118 часов; форма отчетности: 3 семестр – зачет.

4.1.Содержание дисциплины

| № п/п | Наименование раздела дисциплины по семестрам | Семестр | Неделя | Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах) | | | | Формы текущего* контроля успеваемости (по срокам текущих аттестаций в семестре) Форма промежуточной аттестации (по семестрам) |
|-------|--|---------|--------|--|----|----|-----|--|
| | | | | ЛК | ПЗ | ЛР | СРС | |
| | | | | | | | | |

| | | | | | | | | |
|----|--|---|---|---|--|---|---|--------------------------------------|
| 1. | Системы с распределенной, общей памятью, примеры систем. Массивно-параллельные системы (MPP). Компьютерные кластеры – специализированные и полнофункциональные. Основные принципы организации параллельной обработки данных: модели, методы и технологии параллельного программирования. | 3 | 1 | 2 | | 2 | 7 | Входная КР |
| 2. | Симметричные мультипроцессорные системы (SMP). Параллельные векторные системы (PVP). Системы с неоднородным доступом к памяти (Numa), примеры систем. | 3 | 2 | | | | 7 | |
| 3. | Классификация Флинна, Шора и т.д. Организация межпроцессорных связей – коммуникационные топологии. Примеры сетевых решений для создания кластерных систем. | 3 | 3 | | | 2 | 7 | |
| 4. | Функциональный параллелизм, параллелизм по данным. Парадигма master-slave. Парадигма SPMD. Модель обмена сообщениями – MPI. Модель общей памяти – OpenMP. | 3 | 4 | | | | 7 | |
| 5. | Концепция виртуальной, разделяемой памяти – Linda. Российские разработки – T-система, система DVM. Проблемы создания средства автоматического распараллеливания программ. | 3 | 5 | 2 | | 2 | 7 | Аттестационная контрольная работа №1 |

| | | | | | | | | |
|-----|---|---|----|---|--|---|---|--------------------------------------|
| 6. | Параллельное программирование с использованием интерфейса передачи сообщений MPI. Параллельное программирование на системах с общей памятью (OpenMP). Использование многопоточности при программировании для многоядерных платформ. | 3 | 6 | | | | 7 | |
| 7. | Библиотека MPI. Модель SIMD. Инициализация и завершение MPI-приложения. Точечные обмены данными между процессами MPI-программы. Режимы буферизации. Управление группами и коммутаторами в MPI. | 3 | 7 | | | 2 | 7 | |
| 8. | Введение в OpenMP. Стандарты программирования для систем с разделяемой памятью. Создание многопоточных приложений. | 3 | 8 | | | | 7 | |
| 9. | Синхронизация данных между ветвями в параллельной программе. Директивы языка OpenMP. | 3 | 9 | 2 | | 2 | 7 | |
| 10. | Параллельное программирование многоядерных GPU. Кластеры из GPU и суперкомпьютеры на гибридной схеме. Существующие многоядерные системы. GPU как массивно-параллельный процессор. | 3 | 10 | | | | 7 | Аттестационная контрольная работа №2 |

| | | | | | | | | |
|-----|--|---|--------|---|--|---|---|--------------------------------------|
| 11. | Архитектура GPU и модель программирования CUDA. Иерархия памяти CUDA. Глобальная, константная, текстурная, локальная, разделяемая и регистровая память. | 3 | 1 1 | | | 2 | 7 | |
| 12. | Программирование многоядерных GPU. Кластеры из GPU. Кластеры и суперкомпьютеры на гибридной схеме. Использование OpenMP и MPI технологий совместно с CUDA. Вопросы оптимизации приложений на CUDA. | 3 | 1 2 | | | | 7 | |
| 13. | Параллельные численные алгоритмы для решения типовых задач вычислительной математики. Степень параллелизма численного алгоритма. | 3 | 1 3 | 2 | | 2 | 7 | |
| 14. | Средняя степень параллелизма численного алгоритма. Зернистость алгоритма. Ускорение и эффективность. Закон Амдала. Алгоритм исследования свойств параллельного алгоритма. | 3 | 1 4 | | | | 7 | |
| 15. | Матричные вычисления (матрично-векторное умножение, умножение матриц, решение систем линейных уравнений). | 3 | 1 5 | | | 2 | 7 | Аттестационная контрольная работа №3 |
| 16. | Параллельный алгоритм умножения матрицы на вектор и его ускорение по сравнению с последовательным алгоритмом. | 3 | 1 6 | | | | 7 | |

| | | | | | | | | |
|-----|---|---|--------|---|---|----|-----|--|
| 17. | Параллельный алгоритм умножения матрицы на матрицу и его ускорение по сравнению с последовательным алгоритмом. Параллельный алгоритм решения СЛАУ прямым методом Гаусса и его ускорение | 3 | 1 7 | 1 | | 1 | 6 | |
| | Итого | | | 9 | - | 17 | 118 | |

4.2 Содержание лабораторных занятий

| № п/п | № лекции из рабочей программы | Наименование лабораторного занятия | Кол-во часов | Рекомендуемая литература и методические разработки (№ источника из списка литературы) |
|-------|-------------------------------|--|--------------|---|
| 1 | 1 | Основные принципы организации параллельной обработки данных: модели, методы и технологии параллельного программирования. | 2 | 1-2 |
| 2 | 3 | Примеры сетевых решений для создания кластерных систем | 2 | 1-4 |
| 3 | 5 | Проблемы создания средства автоматического распараллеливания программ. | 2 | 1-4 |
| 4 | 7 | Режимы буферизации. Управление группами и коммутаторами в MPI. | 2 | 1-3 |
| 5 | 9 | Синхронизация данных между ветвями в параллельной программе. Директивы языка OpenMP. | 2 | 1-5 |
| 6 | 11 | Архитектура GPU и модель программирования CUDA. Иерархия памяти CUDA. | 2 | 1-5 |
| 7 | 13 | Параллельные численные алгоритмы для решения типовых задач вычислительной математики | 2 | 1-5 |
| 8 | 15 | Матричные вычисления (матрично-векторное умножение, умножение матриц, решение систем линейных уравнений). | 2 | 1-5 |
| 9 | 17 | Параллельный алгоритм решения СЛАУ прямым методом Гаусса и его ускорение | 1 | 1-5 |
| | | Итого | 17 | |

4.3 Тематика для самостоятельной работы студента

| №№ п/п | Темы для самостоятельного изучения | Кол-во часов | Рекомендуемая литература и источники информации | Формы контроля СРС |
|-----------|--|-----------------|--|--------------------------|
| 1 | Компьютерные кластеры – специализированные и полнофункциональные. Основные принципы организации параллельной обработки данных: модели, методы и технологии параллельного программирования. | 7 | 1-2 | Реферат |
| 2 | Симметричные мультипроцессорные системы (SMP). | 7 | 1-2 | |
| 3 | Классификация Флинна, Шора и т.д. Организация межпроцессорных связей – коммуникационные топологии. | 7 | 1-3 | |
| 4 | Функциональный параллелизм, параллелизм по данным. Парадигма master-slave. | 7 | 1-4 | Реферат |
| 5 | Концепция виртуальной, разделяемой памяти – Linda. Российские разработки – T-система, система DVM. | 7 | 1-4 | |
| 6 | Параллельное программирование с использованием интерфейса передачи сообщений MPI. Параллельное программирование на системах с общей памятью (OpenMP). | 7 | 1-5 | |
| 7 | Точечные обмены данными между процессами MPI-программы. Режимы буферизации. Управление группами и коммутаторами в MPI. | 7 | 1-6 | |
| 8 | Введение в OpenMP. Стандарты программирования для систем с разделяемой памятью.. | 7 | 1-7 | Реферат |
| 9 | Синхронизация данных между ветвями в параллельной программе. Директивы языка OpenMP. | 7 | 1-7 | |
| 10 | Кластеры из GPU и суперкомпьютеры на гибридной схеме. | 7 | 1-7 | |
| 11 | Архитектура GPU и модель программирования CUDA. Иерархия памяти CUDA | 7 | 1-7 | |
| 12 | Программирование многоядерных GPU. Кластеры из GPU. Кластеры и суперком- | 7 | 1-9 | |

| | | | | |
|----|--|------------|------|--|
| | пьютеры на гибридной схеме. | | | |
| 13 | Параллельные численные алгоритмы для решения типовых задач вычислительной математики. | 7 | 1-9 | |
| 14 | Средняя степень параллелизма численного алгоритма. Зернистость алгоритма. | 7 | 1-9 | |
| 15 | Матричные вычисления (матрично-векторное умножение, умножение матриц, решение систем линейных уравнений). | 7 | 1-9 | |
| 16 | Параллельный алгоритм умножения матрицы на вектор и его ускорение по сравнению с последовательным алгоритмом. | 7 | 1-10 | |
| 17 | Параллельный алгоритм умножения матрицы на матрицу и его ускорение по сравнению с последовательным алгоритмом. | 6 | 1-10 | |
| | ВСЕГО | 118 | | |

5. Образовательные технологии

5.1. При выполнении лабораторных работ используются алгоритмические языки программирования, а также учебно-исследовательская программная система Параллельная Лаборатория (ПараЛаб), которая может быть использована для организации лабораторного практикума для изучения и исследования эффективности параллельных алгоритмов.

5.2. При чтении лекций используются активные формы, то есть привлекаются студенты в качестве экспертов для ответов на вопросы при рассмотрении принципов параллельного программирования с использованием технологии OpenMP. Это позволяет более детально понять излагаемый материал. Удельный вес занятий, проводимых в интерактивной форме составляет 20% (10 часов) аудиторных занятий.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

6.1. Вопросы текущих контрольных работ

Входная контрольная работа

1. Основные направления развития высокопроизводительных компьютеров. Пути достижения параллелизма: независимость функционирования отдельных функциональных устройств, избыточность элементов вычислительной системы, дублирование устройств.
2. Векторная и конвейерная обработка данных.
3. Многопроцессорная и многомашинная, параллельная обработка данных.
4. Закон Мура, сдерживающие факторы наращивания количества транзисторов на кристалле и частоты процессоров. Сдерживающие факторы повсеместного внедрения параллельных вычислений.

5. Перечень критических задач, решение которых без использования параллельных вычислений затруднено или вовсе невозможно.
6. Поясните понятие суперкомпьютера
7. Возможно ли увеличение производительности суперкомпьютера прямо пропорционально увеличению количества процессорных элементов?
8. В чем заключаются основные способы достижения параллелизма?
9. В чем могут состоять различия параллельных вычислительных систем?
10. Что положено в основу классификация Флинна?

Аттестационная контрольная работа №1

1. Какие классы систем известны для мультикомпьютеров?
2. Что такое массивно-параллельный компьютер?
3. Что такое векторно-конвейерный компьютер?
4. В чем состоят положительные и отрицательные стороны кластерных систем?
5. Каковы причины появления концепции метакомпьютинга?
6. Каковы причины появления Grid проектов?
7. Сравните метакомпьютинг и Grid технологии.
8. Что общего и в чем различия между традиционной общей памятью в SMP-компьютерах и пространством кортежей в системе Linda?
9. Необходимо написать программу для компьютера с общей памятью. Чему отдать предпочтение: OpenMP или Linda? Сравните технологии с различных точек зрения.

Аттестационная контрольная работа № 2

1. Какой минимальный набор средств является достаточным для организации параллельных вычислений в системах с распределенной памятью?
2. В чем различие парных и коллективных операций передачи данных?
3. Какие режимы передачи данных поддерживаются в MPI?
4. Как организуется неблокирующий обмен данными в MPI?
5. Каковы преимущества программирования с использованием OpenMP?
6. Как достигается балансировка нагрузки в OpenMP?
7. Какие особенности организации параллельного цикла в OpenMP?
8. Какие похожие операции есть в MPI и OpenMP?
9. OpenMP: Директивы OpenMP, Переменные окружения.
10. OpenMP: Библиотечные функции. Средства синхронизации.

Аттестационная контрольная работа № 3

1. Поясните понятие суперкомпьютера
2. Возможно ли увеличение производительности суперкомпьютера прямо пропорционально увеличению количества процессорных элементов?
3. Дайте определение вычислительного кластера.
4. Опишите виды кластеров, их особенности

5. В распоряжении программистов есть, с одной стороны, MPI и OpenMP, а с другой стороны, компьютеры с общей и распределенной памятью.
6. Какая технология программирования какой архитектуре лучше соответствует?
7. Как описываются в MPI передаваемые сообщения?
8. В чем различие понятий процесса и процессора?
9. В чем состоит концепция нитей?
10. Каковы преимущества программирования на MPI?

6.2 Вопросы к экзамену

1. Основные направления развития высокопроизводительных компьютеров. Пути достижения параллелизма: независимость функционирования отдельных функциональных устройств, избыточность элементов вычислительной системы, дублирование устройств.
2. Векторная и конвейерная обработка данных.
3. Многопроцессорная и многомашинная, параллельная обработка данных.
4. Закон Мура, сдерживающие факторы наращивания количества транзисторов на кристалле и частоты процессоров. Сдерживающие факторы повсеместного внедрения параллельных вычислений.
5. Перечень критических задач, решение которых без использования параллельных вычислений затруднено или вовсе невозможно.
6. Однопроцессорная оптимизация. Архитектурно-зависимая оптимизация; отличия развертывания циклов для векторных и кэш-ориентированных архитектур. Конвейерная обработка данных. Зависимость производительности процессора от способа описания и хранения данных.
7. Стандартные методики измерения производительности MIPS, MFLOPS и т.д.
8. Классификация многопроцессорных вычислительных систем
9. Парадигмы, модели и технологии параллельного программирования
10. Параллельное программирование с использованием интерфейса передачи сообщений MPI
11. Параллельное программирование на системах с общей памятью (OpenMP)
12. Параллельное программирование на системах смешанного типа.
13. Классификация ошибок параллельных программ (сильные, слабые ошибки ...). Особенности отладки параллельных приложений. Трассировка.
14. Степень параллелизма численного алгоритма. Средняя степень параллелизма численного алгоритма. Зернистость алгоритма. Ускорение и эффективность. Закон Амдала.
15. Определение параллелизма: анализ задачи с целью выделить подзадачи, которые могут выполняться одновременно. Выявление параллелизма: изменение структуры задачи таким образом, чтобы можно было эффективно выполнять подзадачи. Выражение параллелизма: реализация параллельного алгоритма в исходном коде с помощью системы обозначений параллельного программирования.
16. Параллельный алгоритм умножения матрицы на вектор и его ускорение по сравнению с последовательным алгоритмом.
17. Параллельный алгоритм умножения матрицы на матрицу и его ускорение по сравнению с последовательным алгоритмом.

6.4. Темы курсовых работ

1. Алгоритм решения систем линейных уравнений методом Гаусса и построение параллельного алгоритма решения методом Гаусса.
2. Умножение матрицы на вектор способом декомпозиции.
3. Разработка параллельного алгоритма умножения матриц при ленточном разбиении матрицы.
4. Последовательное умножение матрицы на вектор.
5. Разработка параллельного алгоритма умножения матрицы на вектор при разделении по столбцам.

6.5. Вопросы для проверки остаточных знаний студентов.

1. Основные направления развития высокопроизводительных компьютеров. Пути достижения параллелизма: независимость функционирования отдельных функциональных устройств, избыточность элементов вычислительной системы, дублирование устройств.
2. Векторная и конвейерная обработка данных.
3. Многопроцессорная и многомашинная, параллельная обработка данных.
4. Закон Мура, сдерживающие факторы наращивания количества транзисторов на кристалле и частоты процессоров. Сдерживающие факторы повсеместного внедрения параллельных вычислений.
5. Перечень критических задач, решение которых без использования параллельных вычислений затруднено или вовсе невозможно.
6. Умножение матрицы на вектор способом декомпозиции.
7. Каковы преимущества программирования на MPI?
8. Поясните понятие суперкомпьютера
9. Возможно ли увеличение производительности суперкомпьютера прямо пропорционально увеличению количества процессорных элементов?
10. Дайте определение вычислительного кластера.
11. Опишите виды кластеров, их особенности
12. В распоряжении программистов есть, с одной стороны, MPI и OpenMP, а с другой стороны, компьютеры с общей и распределенной памятью.
13. Какая технология программирования какой архитектуре лучше соответствует?
14. Как описываются в MPI передаваемые сообщения?
15. Перечень критических задач, решение которых без использования параллельных вычислений затруднено или вовсе невозможно.
16. Поясните понятие суперкомпьютера
17. Возможно ли увеличение производительности суперкомпьютера прямо пропорционально увеличению количества процессорных элементов?
18. В чем заключаются основные способы достижения параллелизма?
19. В чем могут состоять различия параллельных вычислительных систем?
20. Что положено в основу классификация Флинна?

7. Учебно-методическое и информационное обеспечение дисциплины

И.О. Заб. Сед. А.М.Б.

| № | Виды занятия | Комплект необходимой литературы | Автор | Издание и год издания | Количество пособий, учебников | |
|----------------------------------|--------------|---|---------------------------------------|---|-------------------------------|---------|
| | | | | | в библ. | на каф. |
| Основная литература | | | | | | |
| 1 | лк, пз, срс | Высокопроизводительные вычисления | Баденко В. Л. | СПб.: Изд-во Политехн. ун-та, 2010. – 180 с. | 1 | 1 |
| 2 | лк, срс | Теория и практика параллельных вычислений | Гергель В.П. | М.:Интернет-Университет, БИНОМ. Лаборатория знаний, 2007. | 25 | 1 |
| 3 | лк, пз, срс | Программирование многопроцессорных вычислительных систем. (http://rsusul.rnd.runnet.ru/tutor/method/index.html) | Букатов А.А., Дацюк В.Н., Жегуло А.И. | Ростов-на-Дону. Издательство ООО «ЦВВР», 2008, 208с. | 5 | 1 |
| 4 | лк, срс | Параллельные вычисления | Воеводи н В.В., Воеводи н Вл.В. | СПб.:БХВ-Петербург, 2008. | | 1 |
| 5 | лк, срс | Параллельное программирование для многопроцессорных вычислительных систем. | Немнюгин С., Стесик О. | СПб.: БХВ-Петербург, 2012, 400 с | 50 | 1 |
| 7 | лк, лб, срс | Параллельное программирование с использованием технологии OpenMP: Учебное пособие.- | Антонов А.С. | М.: Изд-во МГУ, 2009 | | 1 |
| Дополнительная литература | | | | | | |
| 8 | лк, лб, срс | Введение в параллельные и векторные методы решения линейных систем | Ортега Дж. | Пер. с англ. М.: Мир, 1991. 367с. | | |
| 9 | лк, лб, срс | Параллельное программирование с использованием технологии MPI | Антонов А.С. | Учебное пособие. // http://rsusul.rnd.runnet.ru/tutor/antonov/ | | |
| 10 | лк, лб, срс | Основы многопоточного, параллельного программирования. | Эндрюс Г.Р | Пер. с англ. М.: Издательский дом “Вильямс”, 2011. 512с. | | |

8. Материально-техническое обеспечение дисциплины (модуля)

Лекционные занятия проходят в мультимедийной аудитории, оснащенной компьютером и проектором. Практические занятия проходят в компьютерном классе. Первая часть занятия посвящена разбору нового материала. Вторая часть – выполнению практических заданий с целью закрепления материала.

Для практических занятий необходимы:

- Компьютерные классы, оснащенные ПЭВМ Intel Pentium 4.
- Программное обеспечение: Microsoft Windows XP/2000, Windows 7.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.04.04 «Программная инженерия» магистерская программа «Разработка программно-информационных систем»

Рецензент от выпускающей кафедры (работодателя) по направлению


Подпись,


ФИО