

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)



УТВЕРЖДАЮ

Проректор по УМР

А.А. Панфилов

« 17 » 04 2015 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Параллельное программирование

(НАИМЕНОВАНИЕ ДИСЦИПЛИНЫ)

Направление подготовки: 02.03.03 Математическое обеспечение и администрирование информационных систем

Профиль/программа подготовки: _____

Уровень высшего образования: бакалавриат

Форма обучения очно-заочная (ускоренное обучение на базе СПО)

Семестр	Трудоемкость зач. ед./час.	Лекций, час.	Практич. занятий, час.	Лаборат. работы, час.	СРС, час	Форма промежуточного контроля (экз./зачет)
4	6/216	18	18	36	108	Экзамен (36), КР
Итого	6/216	18	18	36	108	Экзамен (36), КР

Владимир, 2015 г.

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Потребность решения сложных прикладных задач с большим объемом вычислений и принципиальная ограниченность максимального быстродействия "классических" - по схеме фон Неймана - ЭВМ привели к появлению многопроцессорных вычислительных систем (МВС). Использование таких средств вычислительной техники позволяет существенно увеличивать производительность ЭВМ при любом существующем уровне развития компьютерного оборудования. При этом, однако, необходимо "параллельное" обобщение традиционной - последовательной - технологии решения задач на ЭВМ. Так, численные методы в случае МВС должны проектироваться как системы параллельных и взаимодействующих между собой процессов, допускающих исполнение на независимых процессорах. Применяемые алгоритмические языки и системное программное обеспечение должны обеспечивать создание параллельных программ, организовывать синхронизацию и взаимоисключение асинхронных процессов и т.п.

Предметом рассмотрения настоящего курса и является изучение перечисленного круга вопросов. Цель курса состоит в изложении математических моделей и методов параллельного программирования для многопроцессорных вычислительных систем, а также практическое освоение приемов и техник разработки параллельных программ.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина «Параллельное программирование» находится в вариативной части основной профессиональной образовательной программы относится к дисциплинам по выбору.

Дисциплина логически и содержательно-методически связана с рядом теоретических дисциплин и практик предшествующего периода обучения (Архитектура компьютеров, Алгоритмы и алгоритмические языки, Теория вычислительных процессов и структур, Структуры и алгоритмы компьютерной обработки данных, Системные и математические основы суперкомпьютерных технологий). Для успешного освоения курса студенты должны: знать устройство и принципы функционирования ЭВМ, иметь представление о базовых алгоритмах и структурах данных, уметь применять языки программирования высокого уровня.

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

В результате освоения дисциплины обучающийся должен демонстрировать следующие профессиональные компетенции:

- готовность анализировать проблемы и направления развития технологий программирования (ОПК-3);

- способность применять в профессиональной деятельности основные методы и средства автоматизации проектирования, производства, испытаний и оценки качества программного обеспечения (ОПК-4).

В результате освоения дисциплины обучающийся должен продемонстрировать следующие результаты образования:

1. **Знать:** классификацию распределенных вычислительных систем, методы повышения производительности вычислительных систем, подходы к построению параллельных алгоритмов (ОПК-3, ОПК-4);
2. **Уметь:** самостоятельно извлекать полезную научно-техническую информацию из различных источников, применять изученные методы при проектировании распределенных алгоритмов (ОПК-3, ОПК-4);
3. **Владеть:** навыками использования современных средств решения вычислительных задач для систем с распределенной памятью (ОПК-3, ОПК-4).

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Общая трудоемкость дисциплины составляет 6 зачетных единиц, 216 часов.

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)					Объем учебной работы, с применением интерактивных методов (в часах / %)	Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
				Лекции	Практич. занятия	Лаб. работы	СРС	КП / КР		
1.	Цели и задачи параллельной обработки данных	4	1-3	3	3	6	9		6 / 50%	Рейтинг-контроль №1
2.	Принципы построения параллельных вычислительных систем	4	4-6	3	3	6	13		5 / 41%	
3.	Модели параллельных вычислительных систем	4	7-9	3	3	6	19		6 / 50%	Рейтинг-контроль №2
4.	Принципы разработки параллельных алгоритмов и программ	4	10-13	3	3	6	17		5 / 41%	
5.	Системы разработки параллельных программ	4	14-15	3	3	6	23		6 / 50%	Рейтинг-контроль №3
6.	Параллельные численные алгоритмы для	4	16-18	3	3	6	27		6 / 50%	

решения типовых задач вычислительной математики										
Всего:	4	18	18	18	36	108	КР	34 / 47%	экзамен (36)	

ЛЕКЦИИ

1. Цели и задачи введения параллельной обработки данных

1.1. Необходимость

Ограничение максимальной производительности однопроцессорных ЭВМ. Постоянная необходимость решения задач, превышающих возможности современных ЭВМ (проблемы "большого вызова"). Необходимость коллективного режима решения задач. Автоматизация управления распределенных технических систем. Технические требования по снижению стоимости и повышению надежности.

1.2. История введения параллелизма

ENIAC, IBM-701, 704, 709, ATLAS, CDC 6600, 7600, ILLIAC IV, Cray-1, Эльбрус.

1.3. Различие многозадачных, параллельных и распределенных вычислений

1.4. Проблемы использования параллелизма

Существование последовательных алгоритмов (закон Амдаля). Повышение производительности последовательных компьютеров (закон Мура). Потери на взаимодействие и передачу данных (гипотеза Минского). Высокая стоимость параллельных систем (закон Гроша). "Последовательность" существующих алгоритмов и программного обеспечения. Зависимость эффективности параллельных вычислений от учета особенностей аппаратуры. Сложность разработки параллельных алгоритмов. Трудоемкость проверки правильности параллельных программ.

2. Принципы построения параллельных вычислительных систем

2.1. Пути достижения параллелизма

Функциональные вычислительные устройства. Многоуровневая и модульная память. Конвейерные и векторные вычисления. Процессорные матрицы. Многопроцессорные вычислительные системы с общей и распределенной памятью (мультипроцессоры и мультикомпьютеры). Микропроцессорные системы.

2.2. Способы построения многопроцессорных вычислительных систем

Схемы коммутации (полная коммутация - общая память, перекрестные коммутаторы, локальные схемы коммутации - общая шина, решетки, кластеры). Анализ параллельных алгоритмов и типовые топологии схем коммутации – кольцо, линейка, решетки, полный граф, гиперкуб, тор, дерево. Аппаратная реализация и программная эмуляция топологий.

2.3. Виды параллельных вычислительных систем

СуперЭВМ. Многопроцессорные вычислительные комплексы (МВС). Многомашинные вычислительные комплексы. Сети ЭВМ.

Примеры современных высокопроизводительных вычислительных систем (Cray T932, IBM SP2, HP Exemplar, ASCI RED). Суперкомпьютерные вычислительные системы в России.

2.4. Классификация МВС

Систематики Флинна и Шора. Потоки данных (команд). Структурная нотация Хокни и Джесхоупа.

2.5. Оценка производительности МВС

Общее выражение для оценки производительности для разного типа МВС. Максимальная (пиковая) производительность. Степень параллелизма (длина полупроизводительности). Удельная производительность. Значения показателей для ряда МВС.

3. Моделирование и анализ параллельных вычислений

3.1. Модели параллельных вычислительных систем

Компьютер с неограниченным параллелизмом (паракомпьютер). Модели многопроцессорных систем с общей и распределенной памятью. Модель конвейерной системы.

3.2. Модель алгоритма в виде графа "операнд - операции"

Представление алгоритма в виде графа потока данных. Расписание параллельных вычислений. Показатель временной сложности алгоритма. Оценка времени выполнения алгоритма для паракомпьютера (предельное распараллеливание) и для систем с конечным количеством процессоров. Зависимость оценок от топологии графа алгоритма и необходимость оптимизации структуры графа. Способы получения оптимального расписания вычислений.

3.3. Модель параллельных вычислений в виде сети Петри

Основные понятия теории сетей Петри. Использование сетей Петри для описания параллельных вычислений. Демонстрация основных проблем параллельных вычислений: синхронизация, взаимоисключение, блокировка (тупики).

3.4. Модель параллельных вычислений в виде графа "процесс-ресурс"

Понятие процесса. Синхронизация параллельных процессов. Аппарат событий. Пример реализации в операционной системе Unix.

Взаимоисключение параллельных процессов. Концепция ресурса. Механизмы взаимоисключения: алгоритм Деккера, семафоры (Дейкстра), мониторы (Вирт). Примеры решения стандартных задач взаимоисключения: кольцевой буфер, проблема "читатели и писатели".

Взаимодействие параллельных процессов посредством механизма передачи сообщений. Механизмы передачи: очереди, почтовые ящики, порты. Принцип рандеву в языках Ада и ОККАМ.

Проблемы взаимодействия процессов. Понятие тупика и условия его возникновения. Предотвращение тупиков. Алгоритм банкира. Обнаружение тупиков и восстановление состояния процессов.

Многозадачный режим работы ЭВМ как частный случай параллельной обработки.

4. Принципы разработки параллельных алгоритмов и программ

4.1. Оценка эффективности параллельных вычислений

Показатель эффекта распараллеливания (ускорение). Эффективность использования вычислительной системы. Способы оценки показателей. Основные характеристики вычислительной системы, влияющие на величины ускорения и эффективности (архитектура, количество процессоров, топология каналов передачи данных).

4.2. Оценка коммуникационной трудоемкости параллельных алгоритмов

Характеристики топологий сети передачи данных. Алгоритмы маршрутизации. Методы передачи данных.

Анализ трудоемкости основных операций передачи данных. Передача данных между двумя процессорами сети. Одиночная и множественная рассылка сообщений. Операция циклического сдвига.

Методы логического представления топологии коммуникационной среды. Отображение кольцевой топологии и топологии решетки на гиперкуб.

4.3. Уровни распараллеливания вычислений

Распараллеливание вычислений на уровне команд, выражений, программных модулей, отдельно выполняемых заданий.

4.4. Этапы построения параллельных алгоритмов и программ

Выбор параллельного алгоритма. Реализация алгоритма в виде параллельной программы. Построение исполняемой программы для параллельной вычислительной системы. Параллельное исполнение машинной программы. Частные постановки: выбор оптимального алгоритма для конкретной вычислительной системы, нахождение наилучшей топологии вычислительной системы для решения определенной задачи, распараллеливание существующего алгоритма.

4.5. Технологические аспекты распараллеливания

Декомпозиция алгоритма на параллельно исполняемые фрагменты вычислений. Распределение заданий по процессорам и балансировка. Синхронизация и взаимоисключение. Организация взаимодействия.

5. Системы разработки параллельных программ

5.1. Создание специализированных языков программирования

Алгоритмический язык ОККАМ. Общая характеристика транспьютера. Концепция процесса в ОККАМ. Методы конструирования агрегированных процессов. Принципы передачи данных между процессами. Пример программ на алгоритмическом языке ОККАМ.

5.2. Расширение существующих языков программирования

Общая характеристика параллельных расширений алгоритмического языка Фортран. Автоматическая векторизация и распараллеливание. Проблемно-ориентированные компиляторы.

Общая характеристика стандарта OpenMP. Создание параллельных областей. Разделение вычислительной нагрузки между потоками. Работа с данными. Синхронизация. Функции и переменные окружения.

Сравнительная характеристика подходов параллельного программирования для систем с распределенной и общей памятью.

5.3. Разработка специализированных библиотек

Система PVM. Концепция параллельной виртуальной вычислительной машины и ее представление в виде распределенной неоднородной системы компьютеров.

Представление программной системы на виртуальной машине. Основные программные примитивы системы PVM. Пример использования.

Система MPI. Общая характеристика. Поддержка модели взаимодействия параллельных вычислителей при помощи передачи сообщений. Основные программные примитивы системы MPI. Пример использования.

Организация вычислений на многопроцессорной системе Parsytec PowerXplorer. Использование технологии кросс эмуляции при разработке параллельных программ. Концепция виртуальных процессоров и каналов передачи данных. Виртуальные топологии системы: кольцо, линейка, звезда, решетка, дерево. Основные программные примитивы. Пример использования.

6. Параллельные численные алгоритмы для решения типовых задач вычислительной математики

6.1. Общие способы распараллеливания алгоритмов

Выявление функциональной независимости отдельных фрагментов алгоритма (параллелизм команд). Геометрическое разделение вычислений (параллелизм данных). Иерархическая декомпозиция обработки данных.

6.2. Организация параллельного исполнения рекурсивных вычислений

Проблема рекурсивной зависимости этапов обработки данных. Каскадная схема. Подход для получения асимптотически ненулевой эффективности. Метод Оутса. Пример для вычисления частичных и общей сумм.

6.3. Параллельные численные алгоритмы линейной алгебры

Способы разбиения матриц (горизонтальная, вертикальная, блочные схемы). Методы вычисления произведения матриц с использованием разных схем разбиения матриц. Обеспечение предельно допустимого параллелизма. Обращение матриц. Параллельные методы решения систем линейных уравнений.

6.4. Параллельные численные алгоритмы решения дифференциальных уравнений в частных производных

Параллельная реализация прямых и итерационных методов решения дифференциальных уравнений в частных производных. Анализ разностных схем для эффективного разделения области определения решаемых задач.

6.5. Параллельные численные алгоритмы многомерной многоэкстремальной оптимизации

Характеристическая схема представления методов глобального поиска. Общий принцип распараллеливания методов. Оценка эффективности

введения параллелизма: эффективность и безызбыточность. Синхронные и асинхронные варианты алгоритмов. Определение наилучших топологий вычислительной системы для реализации методов.

ЛАБОРАТОРНЫЕ И ПРАКТИЧЕСКИЕ РАБОТЫ

1. Разработка параллельных программ с использованием интерфейса передачи сообщений MPI

1) Начальная параллельная программа (печать идентификаторов процессов) - запуск (локальный, распределенный). Оценка времени выполнения программы, синхронизация, коллективные операции.

2) Численное интегрирование - создание проекта на основе готового исходного текста программы, настройка опций, компиляция, запуск.

3) Скалярное произведение векторов.

2. Разработка параллельных программ с использованием технологии OpenMP

1) Скалярное произведение векторов. Глобальные и локальные данные потоков. Критические секции доступа к разделяемым данным. Синхронизация.

2). Умножение матриц (варианты распараллеливания вложенных циклов).

2. Практикум по разработке параллельных алгоритмов и программ для решения задач вычислительной математики

Обзор библиотеки MPI: установка, настройка, схема функционирования.

4. Практикум по оценке эффективности параллельных методов для разных топологий многопроцессорных вычислительных систем

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

- лекционно-семинарская система обучения (традиционные лекционные и лабораторные занятия);

- обучение в малых группах (выполнение лабораторных работ в группах из двух или трёх человек);

- мастер-классы (демонстрация на лабораторных занятиях принципов расчета и проектирования оптических деталей и оптических систем);

- применение мультимедиа технологий (проведение лекционных занятий с применением компьютерных презентаций и демонстрационных роликов с помощью проектора или ЭВМ);

- информационно-коммуникационные технологии (применение информационных технологий для мониторинга текущей успеваемости студентов и контроля знаний).

6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО- МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

а) Вопросы к рейтинг-контролю:

Рейтинг-контроль №1

1. CISC- и RISC-процессоры. Основные черты RISC- архитектуры.
2. Повышение производительности процессоров за счет конвейеризации. Условия оптимального функционирования конвейера.
3. Суперконвейерные и суперскалярные процессоры. Выделение независимо работающих устройств: IU, FPU, MMU, BU.
4. Методы уменьшения негативного влияния инструкций перехода на производительность процессора.
5. Повышение производительности процессоров за счет введения кэш-памяти. Кэши: единый, Гарвардский, с прямой записью, с обратной записью.
6. Организация кэш-памяти. Алгоритмы замены данных в кэш-памяти. Специальные кэши.
7. Согласование кэшей в мультипроцессорных системах с общей памятью.
8. Виды многопроцессорных архитектур.
9. Поддержка многозадачности и многопроцессорности специальными инструкциями процессора. Организация данных во внешней памяти.
10. Программа, процессор, процесс. Основные составляющие процесса, состояния процесса. Стек, виртуальная память, механизмы трансляции адреса.
11. Механизмы взаимодействия процессов. Разделяемая память, семафоры, сигналы, почтовые ящики, события. Задачи (threads). Сравнение с процессами. Ресурсы, приоритеты. Параллельные процессы. Связывание. Статическое и динамическое связывание.
12. Виды ресурсов: аппаратные, программные, активные, пассивные, локальные, разделяемые, постоянные, временные, некритичные, критичные.

Рейтинг-контроль №2

13. Типы взаимодействия процессов: сотрудничающие и конкурирующие процессы. Критические секции, взаимное исключение процессов (задач).
14. Проблемы, возникающие при синхронизации задач и идеи их разрешения.
15. Состояния процесса и механизмы перехода из одного состояния в другое.
16. Стандарты на UNIX-системы.
17. Управление процессами. Функции fork, execl, execv, waitpid. Примеры использования.
18. Работа с сигналами. Функция signal. Пример использования.
19. Разделяемая память. Функции shmget, shmat, shmctl. Примеры использования.
20. Семафоры. Функции semget, semop, semctl. Примеры использования.
21. События. Примитивные операции. Организация взаимодействия клиент- сервер с помощью событий.
22. Очереди сообщений. Функции msgget, msgsnd, msgrcv, msgctl. Примеры использования.

23. Управление задачами (threads). Функции `pthread_create`, `pthread.join`, `sched.yield`.
Примеры использования.
24. Объекты синхронизации типа `mutex`. Функции `pthread_mutex_init`, `pthread_mutex_lock`, `pthread_mutex_trylock`, `pthread_mutex_unlock`, `pthread_mutex_destroy`. Примеры использования.
25. Объекты синхронизации типа `condvar`. Функции `pthread_cond_init`, `pthread_cond_signal`, `pthread_cond_broadcast`, `pthread_cond_wait`, `pthread_cond_destroy`. Примеры использования.
26. Пример `multithread`-программы умножения матрицы на вектор.

Рейтинг-контроль №3

27. Message Passing Interface (MPI). Общая структура MPI-программы. Функции `MPI_Init`, `MPI_Finalize`. Сообщения и их виды.
28. Коммуникаторы. Функции `MPI_Comm_size`, `MPI_Comm_rank`. Примеры использования.
29. Парный обмен сообщениями. Функции `MPI_Send`, `MPI_Recv`. Примеры использования.
30. Операции ввода-вывода в MPI-программах. Примеры.
31. Дополнительные возможности для парного обмена сообщениями. Функции `MPI_Sendrecv`, `MPI_Sendrecv_replace`.
32. Дополнительные возможности для парного обмена сообщениями. Функции `MPI_Isend`, `MPI_Irecv`, `MPI_Test`, `MPI_Testany`, `MPI_Wait`, `MPI_Waitany`.
33. Коллективный обмен сообщениями. Функции `MPI_Barrier`, `MPI_Abort`, `MPI_Bcast`.
34. Коллективный обмен сообщениями. Функции `MPI_Reduce`, `MPI_Allreduce`, `MPI_Or_create`, `MPI_Or_free`. Пример MPI-программы, вычисляющей определенный интеграл.
35. Время в MPI-программах. Функции `MPI_Wtime`, `MPI_Wtick`. Пример использования.
36. Пример MPI-программы умножения матрицы на вектор.
37. Дополнительные возможности для коллективного обмена массивами данных. Функции `MPI_Gather`, `MPI_Allgather`, `MPI_Scatter`,
38. Пересылка структур данных. Создание нового MPI-типа данных с помощью `MPI_Type_struct`. Функции `MPI_Address`, `MPI_Type_commit`, `MPI_Type_free`. Пример использования.

б) Вопросы к экзамену:

39. CISC- и RISC-процессоры. Основные черты RISC- архитектуры.
40. Повышение производительности процессоров за счет конвейеризации.
Условия оптимального функционирования конвейера.
41. Суперконвейерные и суперскалярные процессоры. Выделение независимо работающих устройств: IU, FPU, MMU, BU.
42. Методы уменьшения негативного влияния инструкций перехода на производительность процессора.
43. Повышение производительности процессоров за счет введения кэш-памяти. Кэши: единый, Гарвардский, с прямой записью, с обратной записью.
44. Организация кэш-памяти. Алгоритмы замены данных в кэш-памяти.
Специальные кэши.

- 45.Согласование кэшей в мультипроцессорных системах с общей памятью.
- 46.Виды многопроцессорных архитектур.
- 47.Поддержка многозадачности и многопроцессорности специальными инструкциями процессора. Организация данных во внешней памяти.
- 48.Программа, процессор, процесс. Основные составляющие процесса, состояния процесса. Стек, виртуальная память, механизмы трансляции адреса.
- 49.Механизмы взаимодействия процессов. Разделяемая память, семафоры, сигналы, почтовые ящики, события. Задачи (threads). Сравнение с процессами. Ресурсы, приоритеты. Параллельные процессы. Связывание. Статическое и динамическое связывание.
- 50.Виды ресурсов: аппаратные, программные, активные, пассивные, локальные, разделяемые, постоянные, временные, некритичные, критичные.
- 51.Типы взаимодействия процессов: сотрудничающие и конкурирующие процессы. Критические секции, взаимное исключение процессов (задач).
- 52.Проблемы, возникающие при синхронизации задач и идеи их разрешения.
- 53.Состояния процесса и механизмы перехода из одного состояния в другое.
- 54.Стандарты на UNIX-системы.
- 55.Управление процессами. Функции fork, execl, execv, waitpid. Примеры использования.
- 56.Работа с сигналами. Функция signal. Пример использования.
- 57.Разделяемая память. Функции shmget, shmat, shmctl. Примеры использования.
- 58.Семафоры. Функции semget, semop, semctl. Примеры использования.
- 59.События. Примитивные операции. Организация взаимодействия клиент-сервер с помощью событий.
- 60.Очереди сообщений. Функции msgget, msgsnd, msgrcv, msgctl. Примеры использования.
- 61.Управление задачами (threads). Функции pthread_create, pthread_join, sched_yield. Примеры использования.
- 62.Объекты синхронизации типа mutex. Функции pthread_mutex_init, pthread_mutex_lock, pthread_mutex_trylock, pthread_mutex_unlock, pthread_mutex_destroy. Примеры использования.
- 63.Объекты синхронизации типа condvar. Функции pthread_cond_init, pthread_cond_signal, pthread_cond_broadcast, pthread_cond_wait, pthread_cond_destroy. Примеры использования.
- 64.Пример multithread-программы умножения матрицы на вектор.
- 65.Message Passing Interface (MPI). Общая структура MPI-программы. Функции MPI_Init, MPI_Finalize. Сообщения и их виды.
- 66.Коммуникаторы. Функции MPI_Comm_size, MPI_Comm_rank. Примеры использования.
- 67.Попарный обмен сообщениями. Функции MPI_Send, MPI_Recv. Примеры использования.
- 68.Операции ввода-вывода в MPI-программах. Примеры.

69. Дополнительные возможности для попарного обмена сообщениями. Функции `MPI_Sendrecv`, `MPI_Sendrecv_replace`.
70. Дополнительные возможности для попарного обмена сообщениями. Функции `MPI_Isend`, `MPI_Irecv`, `MPI_Test`, `MPI_Testany`, `MPI_Wait`, `MPI_Waitany`
71. Коллективный обмен сообщениями. Функции `MPI_Barrier`, `MPI_Abort`, `MPI_Bcast`.
72. Коллективный обмен сообщениями. Функции `MPI_Reduce`, `MPI_Allreduce`, `MPI_Op_create`, `MPI_Op_free`. Пример MPI-программы, вычисляющей определенный интеграл.
73. Время в MPI-программах. Функции `MPI_Wtime`, `MPI_Wtick`. Пример использования.
74. Пример MPI-программы умножения матрицы на вектор.
75. Дополнительные возможности для коллективного обмена массивами данных. Функции `MPI_Gather`, `MPI_Allgather`, `MPI_Scatter`,
76. Пересылка структур данных. Создание нового MPI-типа данных с помощью `MPI_Type_struct`. Функции `MPI_Address`, `MPI_Type_commit`, `MPI_Type_free`. Пример использования.

в) Экзаменационные задачи:

1. Напишите программу, используя блокирующие коммуникационные функции (`MPI_Send`, `MPI_Recv`), реализующую следующий алгоритм: на нулевом процессоре инициализируется переменная (`float a`); нулевой процессор рассылает переменную `a` всем процессорам, включая самого себя; после получения переменной `a` все процессоры прибавляют к ней свой индивидуальный номер и передают на нулевой процессор; нулевой процессор получает от всех процессоров данные и выводит на экран в формате: номер процессора, пересланное им значение переменной `a`.
2. Напишите программу, используя блокирующие коммуникационные функции (`MPI_Send`, `MPI_Recv`), реализующую алгоритм передачи данных по кольцу: очередной процессор дожидается сообщения от предыдущего и потом посылает следующему процессу.
3. Напишите программу, используя коммуникационные функции (`MPI_Isend`, `MPI_Irecv`), передающую двумерный массив между двумя процессорами.
4. Напишите программу, используя коммуникационную функцию (`MPI_Sendrecv`), реализующую алгоритм передачи данных по кольцу: очередной процессор дожидается сообщения от предыдущего и потом посылает следующему процессу.

г) Примерные темы курсовых работ:

1. Решение задач распознавания образов с использованием параллельных вычислений.
2. Умножение матриц, ленточный алгоритм.
3. Умножение матриц, блочные схемы распределения данных (алгоритмы Фокса и Кеннона).

4. Параллельная сортировка: алгоритмы пузырьковой сортировки, сортировки Шелла и быстрой сортировки.
5. Задачи обработки графов: построение минимального охватывающего дерева, поиск кратчайших путей

д) Самостоятельная работа студентов:

- 1) Работа с материалами, вынесенными на самостоятельное изучение; подготовка к промежуточной аттестации.
- 2) Подготовка к лабораторным работам и оформление отчетов по результатам их выполнения. Контроль осуществляется на занятиях в виде устных ответов на вопросы преподавателя по содержанию отчета.
- 3) Выполнение задания курсовой работы и подготовка отчета. Защита курсовой работы (зачет/незачет) осуществляется в конце семестра в форме демонстрационного запуска на ЭВМ реализованной программы и ответов на вопросы по содержанию отчета.

Распределение видов самостоятельной работы по разделам дисциплины:

№ п/п	Раздел дисциплины	Вид СРС		
		(1)	(2)	(3)
1.	Цели и задачи параллельной обработки данных	4	—	—
2.	Принципы построения параллельных вычислительных систем	5	—	—
3.	Модели параллельных вычислительных систем	5	—	—
4.	Принципы разработки параллельных алгоритмов и программ	5	6	—
5.	Системы разработки параллельных программ	7	6	—
6.	Параллельные численные алгоритмы для решения типовых задач вычислительной математики	12	14	44
	Всего	38 ч.	26 ч.	44 ч.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

а) основная литература

1. Антонов, Александр Сергеевич. Технологии параллельного программирования MPI и OpenMP : учебное пособие для вузов по направлениям 010400 "Прикладная математика и информатика" и 010300 "Фундаментальная информатика и информационные технологии" / А. С. Антонов ; Московский государственный университет имени М. В. Ломоносова (МГУ) ; авт. предисл. В. А. Садовничий .— Москва : Московский университет (МГУ), 2012 .— 339 с. : ил. — (Суперкомпьютерное образование) .— Библиогр.: с. 333-334 .— ISBN 978-5-211-06343-3.

2. Основы параллельного программирования [Электронный ресурс] / Богачёв К.Ю. - М. : БИНОМ, 2013. - <http://www.studentlibrary.ru/book/ISBN9785996309399.html>
3. Модели параллельного программирования [Электронный ресурс] / Федотов И.Е. - М. : СОЛОН-ПРЕСС, 2012. - <http://www.studentlibrary.ru/book/ISBN9785913591029.html>
4. Федотов И.Е. Модели параллельного программирования [Электронный ресурс] / Федотов И.Е.— Электрон. текстовые данные.— М.: СОЛОН-ПРЕСС, 2012.— 384 с.— Режим доступа: <http://www.iprbookshop.ru/20877>.— ЭБС «IPRbooks», по паролю

б) дополнительная литература:

1. Куликов И.М. Технологии разработки программного обеспечения для математического моделирования физических процессов. Часть 1. Использование суперкомпьютеров, оснащенных графическими ускорителями [Электронный ресурс]: учебное пособие/ Куликов И.М.— Электрон. текстовые данные.— Новосибирск: Новосибирский государственный технический университет, 2013.— 40 с.— Режим доступа: <http://www.iprbookshop.ru/45044>.— ЭБС «IPRbooks», по паролю
2. Модели распределенных вычислений [Электронный ресурс] / Топорков В.В. - М. : ФИЗМАТЛИТ, 2011. - <http://www.studentlibrary.ru/book/ISBN5922104950.html>
3. Богачёв К.Ю. Основы параллельного программирования [Электронный ресурс]: учебное пособие/ Богачёв К.Ю.— Электрон. текстовые данные.— М.: БИНОМ. Лаборатория знаний, 2015.— 343 с.— Режим доступа: <http://www.iprbookshop.ru/20702>.— ЭБС «IPRbooks», по паролю

в) Программное обеспечение и Интернет-ресурсы:


1. Лаборатория Параллельных информационных технологий Научно-исследовательского вычислительного центра МГУ <http://parallel.ru>
2. MPICH: a high performance and widely portable implementation of the Message Passing Interface (MPI) standard. <https://www.mpich.org/>
3. Оригиналы стандарта MPI: <http://www.mpi-forum.org/docs/docs.html>.


8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Лекционные аудитории, оснащённые доской (для мела или маркера), экраном для проекционных систем, проектором и ноутбуком (420-3, 430-3).

Аудитории для проведения занятий, оснащённые современными персональными компьютерами, объединёнными в локальную вычислительную сеть и укомплектованными необходимым системным и прикладным программным обеспечением (511-3, 100-3).

Рабочая программа дисциплины составлена в соответствии с требованиями ФГОС ВО по направлению 02.03.03 Математическое обеспечение и администрирование информационных систем

Рабочую программу составил: доцент кафедры ФиПМ  А.С. Голубев

Рецензент (представитель работодателя) директор ООО "РС Сервис" 

(место работы, должность, ФИО, подпись)

Программа рассмотрена и одобрена на заседании кафедры ФиПМ протокол № 11А от « 17 » апреля 2015 года.

Заведующий кафедрой  С.М. Аракелян

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии направления 02.03.03 Математическое обеспечение и администрирование информационных систем

протокол № 11А от « 17 » апреля 2015 года.

Председатель комиссии  С.М. Аракелян

**ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ
РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ (МОДУЛЯ)**

Рабочая программа одобрена на _____ учебный год
Протокол заседания кафедры № _____ от _____ года
Заведующий кафедрой _____

Рабочая программа одобрена на _____ учебный год
Протокол заседания кафедры № _____ от _____ года
Заведующий кафедрой _____

Рабочая программа одобрена на _____ учебный год
Протокол заседания кафедры № _____ от _____ года
Заведующий кафедрой _____