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

УТВЕРЖДАЮ
Проректор по ОД
А.А. Панфилов
«03» 03 2018г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Параллельное программирование

(НАИМЕНОВАНИЕ ДИСЦИПЛИНЫ)

Направление подготовки: 01.03.02 Прикладная математика и информатика

Профиль/программа подготовки: _____

Уровень высшего образования: бакалавриат

Форма обучения очная

Семестр	Трудоемкость зач. ед./час.	Лекций, час.	Практич. занятий, час.	Лаборат. работы, час.	СРС, час	Форма промежуточного контроля (экз./зачет)
8	5/180	32	—	24	97	Экзамен / 27, КР
Итого	5/180	32	—	24	97	Экзамен / 27, КР

Владимир, 2018г.

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 и относится к дисциплинам по выбору.

Дисциплина логически и содержательно-методически связана с рядом теоретических дисциплин и практик предшествующего периода обучения (Архитектура компьютеров, Алгоритмы и анализ сложности, Языки и методы программирования, Системное и прикладное программное обеспечение, Операционные системы, Объектно-ориентированное программирование). 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.

Дисциплина формирует знания и навыки, необходимые в практической деятельности квалифицированного специалиста, изучается в конце теоретического курса. В рамках учебного процесса они могут быть использованы при подготовке выпускной квалификационной работы.

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

В результате освоения дисциплины обучающийся должен демонстрировать следующие компетенции:

- способность использовать базовые знания естественных наук, математики и информатики, основные факты, концепции, принципы теорий, связанных с прикладной математикой и информатикой (ОПК-1);
- способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей,

образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям (ОПК-3);

- способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения (ПК-7).

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования:

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

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Общая трудоемкость дисциплины составляет 5 зачетных единиц, 180 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	8	1	4	—	—	6		2/50	Рейтинг-контроль №1
2.	Принципы построения параллельных вычислительных систем	8	2	4	—	—	10		2/50	
3.	Модели параллельных вычислительных систем	8	3-4	6	—	7	12		8/61	Рейтинг-контроль №2
4.	Принципы разработки параллельных алгоритмов и программ	8	5-6	6	—	5	14		6/54	
5.	Системы разработки параллельных программ	8	6-8	6	—	6	13		6/50	Рейтинг-контроль №3
6.	Параллельные численные алгоритмы для решения типовых задач вычислительной математики	8	8-9	6	—	6	42		6/50	
	Всего:	8	9	32	—	24	97	КР	30/53	Экзамен, 27

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 РАБОТЫ

1. Разработка параллельных программ с использованием интерфейса передачи сообщений MPI

Мини-MPI (старт, финиш, передача и прием сообщений).

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

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

Пример 3: Скалярное произведение векторов (самостоятельное задание 1).

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

Обзор библиотеки MPI: установка, настройка, схема функционирования.

Пример 4: Решение задач распознавания образов (выполнение задания под руководством преподавателя).

Пример 5. Умножение матриц, ленточный алгоритм (самостоятельное задание 2).

Пример 6. Умножение матриц, блочные схемы распределения данных (алгоритмы Фокса и Кеннона).

3. Практикум по использованию библиотек параллельных методов ParaLib для решения задач вычислительной

Обзор библиотеки ParaLib: назначение, классы решаемых задач, принципы разработки.

Пример 7. Функции библиотеки ParaLib для умножения матриц при блочном способе распределения данных" (алгоритмы Фокса и Кеннона): схема реализации, теоретическая оценка эффективности, вычислительные эксперименты.

Анализ интенсивности потоков передачи данных, осуществляемых в ходе параллельных вычислений в системах с распределенной памятью, с использованием инструментальной библиотеки ParaScope.

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

Характеристика программной лаборатории ParaLab как интегрированной системы для проведения вычислительных экспериментов с параллельными методами для оценки их эффективности:

- Моделирование многопроцессорных вычислительных систем (выбор топологии, задание количества и производительности процессоров, выбор метода передачи данных и задание коммуникационных характеристик сети),
- Определение класса решаемых задач и задание параметров задачи,
- Выбор параллельного метода решения задачи и настройка значений его параметров,
- Установка графических индикаторов для наблюдения за процессом параллельных вычислений (состояние данных на процессорах системы, передача информации по сети, текущая оценка решения исходной вычислительной задачи),
- Проведение экспериментов в режиме имитации вычислений; пошаговый, последовательный (непрерывный) и циклический (серийный) способы проведения экспериментов; одновременное выполнение нескольких экспериментов в режиме разделения времени для разных вариантов топологии вычислительной системы, параметров задачи, количества процессоров и т.п.,
- Анализ результатов с использованием сведений из журнала экспериментов; оценка времени решения задач в зависимости от размерности задачи и количества процессоров; построение зависимостей ускорения и эффективности параллельных вычислений,
- Проведение экспериментов в режиме реальных параллельных вычислений; выполнение параллельных программ в виде множества независимых процессов на одном процессоре; удаленный доступ к многопроцессорной вычислительной системе (кластеру); сравнение теоретических оценок и результатов реальных вычислительных экспериментов.

5. Разработка параллельных программ с использованием технологии OpenMP

Общая характеристика технологии OpenMP: потоки, параллельные области, распределение вычислений между потоками.

Пример 8: Скалярное произведение векторов. Глобальные и локальные данные потоков. Критические секции доступа к разделяемым данным. Синхронизация.

Пример 9. Умножение матриц (варианты распараллеливания вложенных циклов, самостоятельное задание 3).

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

Пример 10. Параллельная сортировка: алгоритмы пузырьковой сортировки, сортировки Шелла и быстрой сортировки (выполнение задания под руководством преподавателя).

Пример 11. Задачи обработки графов: построение минимального охватывающего дерева, поиск кратчайших путей (самостоятельное задание 4)

7. Практикум по методам параллельных вычислений для решения дифференциальных уравнений в частных производных

Общая характеристика методом конечных разностей (метод сеток) для численного решения дифференциальных уравнений.

Методы распараллеливания метода конечных разностей для вычислительных систем с общей памятью (организация взаимоисключения для оценки погрешности решения, избыток синхронизации, проблема сериализации и блокировки, обеспечение тождественности последовательного и параллельного вариантов расчетов, волновые схемы вычислений, блочная схема распределения данных, балансировка, использование очереди заданий).

Методы распараллеливания метода конечных разностей для вычислительных систем с распределенной памятью (ленточная и блочная схемы распределения данных, волновые схемы вычислений, оценка трудоемкости передачи данных).

Практикум по решению дифференциальных уравнений в частных производных с использованием учебно-исследовательской системы ParaGrid (постановка задачи, выполнение вычислений, визуализация данных, анализ результатов расчетов, изменение параметров и продолжение вычислений).

8. Практикум по использованию библиотек параллельных методов для решения сложных научно-технических задач

Обзор библиотеки PLAPACK: структура, объекты, методы, использование. Пример 12: Умножение матриц - использование PLAPACK (выполнение задания под руководством преподавателя).

9. Практикум по методам параллельных вычислений для решения задач многомерной многоэкстремальной оптимизации

Общая характеристика предметной области (постановка задачи глобальной оптимизации, редукция размерности для сведения многомерных задач к одномерным постановкам, информационно-статистические алгоритмы глобального поиска, примеры).

Использование множественных разверток типа кривой Пеано для построения различных сеток в области решения оптимизационной задачи. Сведение проблемы многомерной оптимизации к семейству одномерных информационно-совместимых задач. Параллельное решение задач порождаемого семейства и схема информационных обменов.

Общность рассмотренного подхода для решения ряда вычислительно-трудоемких научно-технических задач (интегрирование, решение систем нелинейных уравнений, восстановление зависимостей, поиск решений, оптимальных по набору критериев (многокритериальная оптимизация) и т.д.).

Практикум по решению задач многоэкстремальной оптимизации с использованием системы параллельных вычислений Абсолют Эксперт (постановка задачи, выполнение вычислений, визуализация данных, анализ результатов расчетов, изменение параметров и продолжение вычислений).

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

1. Типы взаимодействия процессов: сотрудничающие и конкурирующие процессы. Критические секции, взаимное исключение процессов (задач).

2. Проблемы, возникающие при синхронизации задач и идеи их разрешения.
3. Состояния процесса и механизмы перехода из одного состояния в другое.
4. Стандарты на UNIX-системы.
5. Управление процессами. Функции fork, execl, execv, waitpid. Примеры использования.
6. Работа с сигналами. Функция signal. Пример использования.
7. Разделяемая память. Функции shmget, shmat, shmctl. Примеры использования.
8. Семафоры. Функции semget. semop, semctl. Примеры использования.
9. События. Примитивные операции. Организация взаимодействия клиент- сервер с помощью событий.
10. Очереди сообщений. Функции msgget, msgsnd, msgrcv, msgctl. Примеры использования.
11. Управление задачами (threads). Функции pthread_create, pthread.join, sched.yield. Примеры использования.
12. Объекты синхронизации типа mutex. Функции pthread_mutex_init, pthread_mutex_lock, pthread_mutex_trylock, pthread_mutex_unlock, pthread_mutex_destroy. Примеры использования.
13. Объекты синхронизации тина condvar. Функции pthread_cond_init, pthread_cond_signal, pthread_cond_broadcast, pthread_cond_wait, pthread_cond_destroy. Примеры использования.
14. Пример multithread-программы умножения матрицы на вектор.

Рейтинг-контроль №3

1. Message Passing Interface (MPI). Общая структура MPI-программы. Функции MPI_Init, MPI_Finalize. Сообщения и их виды.
2. Коммуникаторы. Функции MPI_Comm_size, MPI_Comm_rank. Примеры использования.
3. Попарный обмен сообщениями. Функции MPI_Send, MPI__Recv. Примеры использования.
4. Операции ввода-вывода в MPI-программах. Примеры.
5. Дополнительные возможности для попарного обмена сообщениями. Функции MPI_Sendrecv, MPI_Sendrecv_replace.
6. Дополнительные возможности для попарного обмена сообщениями. Функции MPI_Isend, MPI_Irecv, MPI_Test, MPI_Testany, MPI_Wait, MPI_Waitany
7. Коллективный обмен сообщениями. Функции MPI_Barrier, MPI_Abort, MPI_Bcast.
8. Коллективный обмен сообщениями. Функции MPI_Reduce, MPI_Allreduce, MPI_0p_create, MPI_0p_free. Пример MPI-программы, вычисляющей определенный интеграл.
9. Время в MPI-программах. Функции MPI_Wtime, MPI_Wtick. Пример использования.
10. Пример MPI-программы умножения матрицы на вектор.
11. Дополнительные возможности для коллективного обмена массивами данных. Функции MPI_Gather, MPI_Allgather, MPI_Scatter,
12. 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) 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.

в) Примерные темы курсовых работ:

1. Численное решение средствами MPI/OpenMP двумерного нелинейного уравнения теплопроводности методом продольно-поперечной прогонки
2. Численное решение средствами MPI/OpenMP двумерного нелинейного уравнения теплопроводности с помощью схемы расщепления
3. Численное решение средствами MPI/OpenMP двумерного нелинейного уравнения теплопроводности с помощью схемы предиктор-корректор
4. Численное решение средствами MPI/OpenMP трехмерного линейного уравнения теплопроводности с помощью явной разностной схемы
5. Численное решение средствами MPI/OpenMP двумерного уравнения Пуассона итерационным методом Зейделя
6. Численное решение средствами MPI/OpenMP двумерного уравнения Пуассона методом блочных итераций
7. Моделирование средствами MPI/OpenMP процесса взаимодействия твердых частиц с потоком газа
8. Численное решение средствами MPI/OpenMP двумерного уравнения переноса
9. Вычисление ранга произвольной матрицы средствами MPI/OpenMP
10. Моделирование средствами MPI/OpenMP процесса прохождения нейтронов через пластину

г) Самостоятельная работа студентов:

1. Подготовка к текущему контролю и промежуточной аттестации.
2. Подготовка к лабораторным работам и оформление отчетов по результатам из выполнения. Контроль осуществляется на занятиях в виде устных ответов на вопросы преподавателя по содержанию отчета.
3. Работа с дополнительной литературой по вопросам, вынесенным на самостоятельное изучение. Контроль осуществляется на экзамене.
    1) Достоинства и недостатки SMP- и MPP- архитектур вычислительных систем.
    2) Отличие понятий процесса и потока в операционных системах.
    3) Проблемы синхронизации задач при параллельном программировании.
    4) Стандарт OpenMP. Назначение.
    5) Существующие реализации стандарта MPI.

6) Протоколы обмена данными между процессами.
7) Использование утилиты MPIRun.
8) Отладка параллельных приложений.
9) Методика оценки эффективности вычислений.
10) Реализация матричных алгоритмов средствами MPI
11) Распределенное решение дифференциальных уравнений.

4. 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)	(4)
1.	Цели и задачи параллельной обработки данных	2	—	4	—
2.	Принципы построения параллельных вычислительных систем	4	—	6	—
3.	Модели параллельных вычислительных систем	4	6	6	—
4.	Принципы разработки параллельных алгоритмов и программ	4	4	6	—
5.	Системы разработки параллельных программ	6	5	4	—
6.	Параллельные численные алгоритмы для решения типовых задач вычислительной математики	6	6	4	20
	Всего	26 ч.	21 ч.	30 ч.	20 ч.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

а) основная литература

1. Антонов, А. С. Параллельное программирование с использованием технологии MPI [Электронный ресурс] / А. С. Антонов. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 83 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/73704.html

2. Левин, М. П. Параллельное программирование с использованием OpenMP [Электронный ресурс] / М. П. Левин. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 133 c. — 978-5-94774-857-4. — Режим доступа: http://www.iprbookshop.ru/52216.html

3. Туральчук, К. А. Параллельное программирование с помощью языка C# [Электронный ресурс] / К. А. Туральчук. — 3-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019. — 189 c. — 978-5-4486-0506-2. — Режим доступа: http://www.iprbookshop.ru/79714.html

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 c.
2. Модели распределенных вычислений [Электронный ресурс] / Топорков В.В. - М.: ФИЗМАТЛИТ, 2011.
3. Технология CUDA в примерах: введение в программирование графических процессоров [Электронный ресурс] / Сандерс Дж., Кэндрот Э. - М. : ДМК Пресс, 2011.

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.

Аудитории для проведения занятий, оснащённые современными персональными компьютерами, объединёнными в локальную вычислительную сеть и укомплектованными необходимым системным и прикладным программным обеспечением.

Рабочая программа дисциплины составлена в соответствии с требованиями ФГОС ВО по направлению 01.03.02 Прикладная математика и информатика

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

Рецензент (представитель работодателя) Ген. директор ООО «ФС Сервис» Квасов Д.С.
(место работы, должность, ФИО, подпись)

Программа рассмотрена и одобрена на заседании кафедры ФиПМ
протокол №1 от 03.09.2018 года.

Заведующий кафедрой _____ С.М. Аракелян

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии направления 01.03.02 Прикладная математика и информатика
протокол №1 от 03.09.2018 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 _____