

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 образовательной деятельности



А.А. Папфилов

« 31 » 08 2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ И ОСНОВЫ СУПЕРКОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Профиль/программа подготовки: Мобильные и Интернет-технологии

Уровень высшего образования: бакалавриат

Форма обучения: очная (ускоренное обучение)

Семестр	Трудоемкость зач. ед./ час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	СРС, час.	Форма промежуточной аттестации (экзамен/зачет/зачет с оценкой)
5	5 / 180	36	—	36	81	экзамен (27 ч.)
Итого	5 / 180	36	—	36	81	экзамен (27 ч.)

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель освоения дисциплины: изучение математических моделей и методов параллельного программирования для многопроцессорных вычислительных систем, а также практическое освоение приемов и техник разработки параллельных программ.

Задачи:

- изучение основных понятий и принципов параллельных вычислений
- освоение современных технологий разработки параллельных программ
- овладение инструментарием разработки параллельных программ

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина «Параллельное программирование и основы суперкомпьютерных технологий» относится к дисциплинам обязательной части блока Б1 Дисциплины (модули) учебного плана.

Изучение данной дисциплины проходит в пятом семестре и опирается на результатах изучения дисциплин «Архитектура компьютеров», «Компьютерные сети», «Основы программирования», «Объектно-ориентированное программирование», «Структуры и алгоритмы компьютерной обработки данных», «Иностранный язык». 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. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП

Код формируемых компетенций	Уровень освоения компетенции	Планируемые результаты обучения по дисциплине характеризующие этапы формирования компетенций (показатели освоения компетенции)
1	2	3
ОПК-2	частичное освоение	Знать: • основные положения и концепции в области программирования; • архитектура языков программирования; • основная терминология в области программного обеспечения; • знаком с содержанием Единого реестра российских программ; Уметь: • умеет осуществлять обоснованный выбор компьютерных/суперкомпьютерных методов и необходимого программного обеспечения при решении задач профессиональной деятельности; Владеть • навыками применения данных методов и программного обеспечения при решении конкретных задач;
ОПК-3	частичное освоение	Знать: • методы теории алгоритмов; • методы системного и прикладного программирования; • принципы и методологии тестирования программного обеспечения; • принципы математического моделирования; • типовые (универсальные) математические (включая информационные и имитационные) модели, формулы, теоремы и методы, используемые в широком наборе областей применения прикладной математики; Уметь: • умеет соотносить знания в области программирования;

		• умеет определять и составлять информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем; • умеет осуществлять обоснованный выбор адекватных поставленной задаче базовых математических моделей; • умеет модифицировать базовые и (или) разрабатывать оригинальные математические модели в соответствии со спецификой поставленной задачи моделирования; Владеть • навыками разработки программного обеспечения; • навыками выполнения математического моделирования от анализа постановки задачи до анализа результатов;
ПК-1	частичное освоение	Знать: • возможности современных и перспективных средств разработки программных продуктов, технических средств; • методологии разработки программного обеспечения и технологии программирования; • методологии и технологии проектирования и использования баз данных; • методы и средства проектирования программного обеспечения и баз данных; • языки формализации функциональных спецификаций; • принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; • типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения; • основные концепции и атрибуты качества программного обеспечения; • механизмы авторизации и аутентификации; • стили написания кода; • методики разработки требований к системе и шаблоны документов требований к системе; • методы классического системного анализа; • методы проведения интервью с поставщиками требований; • стандарты оформления технических заданий; Уметь: • проводить анализ исполнения требований; • вырабатывать варианты реализации требований; • проводить оценку и обоснование рекомендуемых решений; • выбирать средства и варианты реализации программного обеспечения; • использовать существующие типовые решения и шаблоны проектирования программного обеспечения; • применять методы и средства проектирования программного обеспечения, структур данных, баз

		данных, программных интерфейсов; • выбирать методики разработки требований к системе и шаблоны документов требований к системе; • определять источники информации для требований к системе; • формулировать цели, исходя из анализа проблем, потребностей и возможностей; • оценивать соответствие требованиям существующих систем и их аналогов; • выполнять тестирование системы с целью проверки её реализации на соответствие требованиям; • формулировать и оформлять запросы на изменение требований; Владеть: • навыки оценки возможностей, времени и трудоемкости реализации требований к программному обеспечению; • навыки согласования требований к программному обеспечению с заинтересованными сторонами; • навыки оценки и согласования сроков выполнения поставленных задач; • навыки разработки и согласования технических спецификаций на программное обеспечение; • навыки формирования и предоставления отчетности в соответствии с установленными регламентами; • навыки проектирования структур данных, баз данных, программных интерфейсов; • навыки выбора стиля написания кода; • навыки планирования работ по разработке требований к системе; • навыки анализа проблемной ситуации; • навыки участия в совещаниях рабочих групп; • навыки оформления требований заинтересованных лиц в документе бизнес-требований; • навыки согласования целей создания системы с заинтересованными лицами; • навыки оформления технического задания на систему; • навыки представления концепции, технического задания и изменений в них заинтересованным лицам;
ПК-2	частичное освоение	Знать: • методы анализа и тестирования требований к программному средству; • методы анализа, проектирования и разработки программного обеспечения; Уметь: • тестировать требования к программному средству; • оформлять документацию по тестированию; • анализировать требования на соответствие принятым стандартам и методам проектирования; • использовать современные CASE-средства; Владеть

		• навыками выявления противоречий в требованиях одинакового и различных уровней к программному средству; • навыками проверки осуществимости функционирования и сопровождения программного средства; • навыками определения возможности введения изменений и дополнений требований к программному средству;
--	--	--

4. ОБЪЕМ И СТРУКТУРА ДИСЦИПЛИНЫ

Трудоемкость дисциплины составляет 5 зачетных единиц, 180 ч.

№ п/п	Наименование тем и/или разделов/тем 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, форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные работы	СРС		
1	Основы параллельных вычислений. Многопоточное программирование.	5	1-6	12	—	8	22	8 / 40	рейтинг-контроль №1
2	Стандарты и технологии разработки параллельных программ	5	7-12	12	—	20	37	20 / 63	рейтинг-контроль №2
3	Проблемные вопросы параллельного программирования	5	13-18	12	—	8	22	8 / 40	рейтинг-контроль №3
Всего за 5 семестр:		5		36	—	36	81	36 / 50	экзамен (27)
Наличие в дисциплине КП/КР		—	—	—	—	—	—	—	—
Итого по дисциплине		5		36	—	36	81	36 / 50	экзамен (27)

Содержание лекционных занятий по дисциплине

Раздел 1. Основы параллельных вычислений. Многопоточное программирование.

- 1) Производительность вычислительных систем. Единицы измерения производительности. Способы оценки.
- 2) Классификация вычислительных систем
- 3) Методы увеличения производительности вычислительных систем. Векторная обработка. Конвейерная обработка. Параллельная обработка.
- 4) Методы разработки параллельных программ. Распараллеливание данных. Распараллеливание команд.
- 5) Характеристики параллельных алгоритмов. Степень параллелизма. Ускорение. Эффективность.
- 6) Закон Амдала
- 7) Многозадачность. Процессы и потоки.
- 8) Реализация многозадачности в операционных системах, алгоритмы планирования.
- 9) Разработка многопоточных программ. POSIX/Windows Threads
- 10) Многопоточность в современных языках программирования. Объектно-ориентированный подход.
- 11) Пулы потоков
- 12) Локальное хранилище потока (Local Thread Storage)

Раздел 2. Стандарты и технологии разработки параллельных программ

- 13) Стандарт OpenMP. Назначение, поддержка в компиляторах. Переменные окружения, функции, директивы.

- 14) Разработка параллельных программ на основе OpenMP
- 15) Введение в высокопроизводительные и распределённые вычисления. Стандарт MPI
- 16) Разработка параллельных программ на основе MPI. Базовые функции, точечный обмен между процессами
- 17) Коллективный обмен между процессами.

Раздел 3. Проблемные вопросы параллельного программирования

- 18) Проблемы взаимодействия задач в параллельных программах. Гонки и тупики.
- 19) Объекты синхронизации взаимодействующих задач на основе блокировок
- 20) Алгоритмы неблокирующей синхронизации, атомарные операции
- 21) Параллельные структуры данных
- 22) Асинхронное программирование
- 23) Особенности параллельных программ для графических процессоров (GPU)

Содержание практических/лабораторных занятий по дисциплине

Лабораторная работа №1 Разработка многопоточных программ.

Функции WinAPI для создания потоков. Класс `std::thread`. Типовая модель параллельной программы. Ожидание завершения рабочих потоков. Измерение характеристик последовательного и параллельного алгоритма.

Лабораторная работа №2 Технология OpenMP

Изучение функций и директив OpenMP. Модель выполнения OpenMP-программы. Решение задач линейной алгебры средствами OpenMP.

Лабораторная работа №3 Стандарт MPI

Библиотечные реализации MPI. Типовая структура MPI-программы. Запуск MPI-программы. Утилита `mpirun/mpiexec`. Применение функции MPI для распределённых вычислений.

Лабораторная работа №4 Синхронизация и балансировка параллельных задач

Разработка многопоточной программы на платформе .Net. Использование общих данных в нескольких потоках. Динамическое распределение вычислений между потоками. Очереди ожидания потоков.

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В преподавании дисциплины используются разнообразные образовательные технологии как традиционные, так и с применением активных и интерактивных методов обучения.

Активные и интерактивные методы обучения:

- Проблемные лекции (раздел 3)
- Групповая дискуссия (все лабораторные занятия);
- Разбор конкретных ситуаций (лабораторная работа 1,4);
- Анализ ситуаций (лабораторная работа 4);
- Уровневая дифференциация (все лабораторные занятия, контрольные мероприятия).

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

Текущий контроль успеваемости студентов

Примерный перечень вопросов к рейтинг-контролю №1

- 1) Оценки производительности вычислительных систем (время, IPS, FLOPS).
- 2) Классификация вычислительных систем (по Флинну и по организации оперативной памяти). Особенности каждого класса систем.
- 3) Подходы, используемые для увеличения производительности вычислительных систем.
- 4) Свойства параллельных алгоритмов: степень параллелизма, зернистость, ускорение, эффективность. Закон Амдаля и его практический смысл.
- 5) Средняя степень параллелизма алгоритма сдвигания, скалярного умножения векторов, умножения матрицы на вектор. Примеры алгоритмов, обладающих идеальной степенью параллелизма.
- 6) Подходы к распараллеливанию алгоритмов: на основе параллелизма данных и параллелизма задач. Преимущества и недостатки, способы реализации.

- 7) Многозадачность. Формы реализации многозадачности: процессы и потоки, их особенности. Свойства задач.
- 8) Планирование и диспетчеризация. Алгоритмы планирования задач ОС.
- 9) Разработка параллельных программ на основе многопоточности. Стандарт POSIX Threads (pthreads). Пример.
- 10) Многопоточность в современных ЯП (на примере C#): объектно-ориентированный подход, сравнение моделей программирования на основе классов `System.Threading.Thread` и `System.Threading.Task`.
- 11) Пулы потоков: назначение, принцип организации. Ограничения при использовании пулов потоков в параллельных программах.
- 12) Локальное хранилище потока. Назначение и пример использования.

Примерный перечень вопросов к рейтинг-контролю №2

- 1) Назначение стандарта OpenMP. Модель выполнения OpenMP-программы.
- 2) Разработка параллельных программ на основе стандарта OpenMP: основные переменные окружения, функции и директивы. Пример.
- 3) Назначение стандарта MPI. Модель выполнения MPI-программы.
- 4) Основные функции MPI, прием/передача данных между отдельными процессами. Блокирующие и неблокирующие операции. Пример.
- 5) Коллективные функции MPI: широковещательная рассылка, разделение и сбор данных, агрегирование данных. Пример.

Примерный перечень вопросов к рейтинг-контролю №3

- 1) Синхронизация задач. Понятие «критической секции» программы. Взаимное исключение задач.
- 2) Гонки при взаимодействии параллельных задач: определение, пример, способы решения.
- 3) Тупиковые ситуации: определение, пример, способы обнаружения и предотвращения.
- 4) Объекты синхронизации: общий принцип работы, типы объектов и их особенности.
- 5) Неблокирующая синхронизация: назначение, преимущества и недостатки, способы реализации.
- 6) Параллельные структуры данных: разновидности, примеры таких структур в языках программирования, преимущества и недостатки.
- 7) Асинхронная модель программирования: назначение, отличие от параллельной модели, варианты реализации в языках программирования.

Промежуточная аттестация по итогам освоения дисциплины (экзамен).

Примерный перечень вопросов к экзамену

- 1) Оценки производительности вычислительных систем (время, IPS, FLOPS).
- 2) Классификация вычислительных систем (по Флинну и по организации оперативной памяти). Особенности каждого класса систем.
- 3) Подходы, используемые для увеличения производительности вычислительных систем.
- 4) Свойства параллельных алгоритмов: степень параллелизма, зернистость, ускорение, эффективность. Закон Амдаля и его практический смысл.
- 5) Средняя степень параллелизма алгоритма сдвигания, скалярного умножения векторов, умножения матрицы на вектор. Примеры алгоритмов, обладающих идеальной степенью параллелизма.
- 6) Подходы к распараллеливанию алгоритмов: на основе параллелизма данных и параллелизма задач. Преимущества и недостатки, способы реализации.
- 7) Многозадачность. Формы реализации многозадачности: процессы и потоки, их особенности. Свойства задач.
- 8) Планирование и диспетчеризация. Алгоритмы планирования задач ОС.
- 9) Разработка параллельных программ на основе многопоточности. Стандарт POSIX Threads (pthreads). Пример.
- 10) Многопоточность в современных ЯП (на примере C#): объектно-ориентированный подход, сравнение моделей программирования на основе классов `System.Threading.Thread` и `System.Threading.Task`.

- 11) Пулы потоков: назначение, принцип организации. Ограничения при использовании пулов потоков в параллельных программах.
- 12) Локальное хранилище потока. Назначение и пример использования.
- 13) Назначение стандарта OpenMP. Модель выполнения OpenMP-программы.
- 14) Разработка параллельных программ на основе стандарта OpenMP: основные переменные окружения, функции и директивы. Пример.
- 15) Назначение стандарта MPI. Модель выполнения MPI-программы.
- 16) Основные функции MPI, прием/передача данных между отдельными процессами. Блокирующие и неблокирующие операции. Пример.
- 17) Коллективные функции MPI: широковещательная рассылка, разделение и сбор данных, агрегирование данных. Пример.
- 18) Синхронизация задач. Понятие «критической секции» программы. Взаимное исключение задач.
- 19) Гонки при взаимодействии параллельных задач: определение, пример, способы решения.
- 20) Тупиковые ситуации: определение, пример, способы обнаружения и предотвращения.
- 21) Объекты синхронизации: общий принцип работы, типы объектов и их особенности.
- 22) Неблокирующая синхронизация: назначение, преимущества и недостатки, способы реализации.
- 23) Параллельные структуры данных: разновидности, примеры таких структур в языках программирования, преимущества и недостатки.
- 24) Асинхронная модель программирования: назначение, отличие от параллельной модели, варианты реализации в языках программирования.

Самостоятельная работа студентов по дисциплине включает в себя следующие виды деятельности:

- 1) проработку учебного материала по конспектам, учебной и научной литературе;
- 2) подготовку к практическим и лабораторным занятиям;
- 3) подготовку по всем видам контрольных мероприятий, в том числе к текущему контролю знаний и промежуточной аттестации.

Темы для самостоятельной работы студентов

- 1) Особенности аппаратной реализации многопроцессорных вычислительных систем и супер-ЭВМ
- 2) Характеристики топологий сети передачи данных. Алгоритмы маршрутизации. Методы передачи данных.
- 3) Методы логического представления топологии коммуникационной среды. Отображение кольцевой топологии и топологии решетки на гиперкуб.
- 4) Модель параллельных вычислений в виде сети Петри
- 5) Модель параллельных вычислений в виде графа «процесс-ресурс»
- 6) Параллельные алгоритмы сортировки
- 7) Параллельные численные алгоритмы линейной алгебры
- 8) Параллельные численные алгоритмы решения дифференциальных уравнений в частных производных
- 9) Параллельные численные алгоритмы многомерной многоэкстремальной оптимизации
- 10) Отладка параллельных приложений.

Фонд оценочных средств для проведения аттестации уровня сформированности компетенций обучающихся по дисциплине оформляется отдельным документом.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

7.1. Книгообеспеченность

Наименование литературы: автор, название, вид издания, издательство	Год издания	КНИГООБЕСПЕЧЕННОСТЬ	
		Количество экземпляров изданий в библиотеке ВлГУ в соответствии с ФГОС	Наличие в электронной библиотеке ВлГУ

		ВО		
1	2	3		4
Основная литература*				
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.	2012	30		—
Основы параллельного программирования [Электронный ресурс] / Богачёв К.Ю. - М. : БИНОМ, 2013. - http://www.studentlibrary.ru/book/ISBN9785996309399.html	2013	—		http://www.studentlibrary.ru/book/ISBN9785996309399.html
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»	2012	—		http://www.iprbookshop.ru/20877
Параллельное программирование на C++ в действии. Практика разработки многопоточных программ [Электронный ресурс] / Энтони Уильямс ; Пер. с англ. Слинкин А.А. - М. : ДМК Пресс, 2012. - http://www.studentlibrary.ru/book/ISBN9785940744481.html	2012	—		http://www.studentlibrary.ru/book/ISBN9785940744481.html
Дополнительная литература				
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	2013	—		http://www.iprbookshop.ru/45044
Модели распределенных вычислений [Электронный ресурс] / Топорков В.В. - М. : ФИЗМАТЛИТ, 2011. - http://www.studentlibrary.ru/book/ISBN5922104950.html	2011	—		http://www.studentlibrary.ru/book/ISBN5922104950.html
Технология CUDA в примерах: введение в программирование графических процессоров [Электронный ресурс] / Сандерс Дж., Кэндрот Э. - М. : ДМК Пресс, 2011. - http://www.studentlibrary.ru/book/ISBN9785940745044.html	2011	—		http://www.studentlibrary.ru/book/ISBN9785940745044.html

7.2. 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

Для реализации данной дисциплины имеются специальные помещения для проведения занятий лекционного типа, занятий практического/лабораторного типа, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы.

Практические и лабораторные занятия проводятся в аудитории (компьютерном классе) 5116-3 (или аналогичном компьютерном классе в зависимости от сетки расписания).

Перечень используемого лицензионного программного обеспечения:

- 1) MS Visual Studio;
- 2) MS HPC Pack SDK (свободно распространяемое ПО)
- 3) MPICH (свободно распространяемое ПО)
- 4) MS Word;

Рабочую программу составил Голубев А.С.
(ФИО, подпись)

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

Программа рассмотрена и одобрена на заседании кафедры физики и прикладной математики
Протокол № 1 от 31.08.2020 года
Заведующий кафедрой _____ Аракелян С.М.
(ФИО, подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии
направления 02.03.02 Фундаментальная информатика и информационные технологии
Протокол № 1 от 31.08.2020 года
Председатель комиссии _____ Аракелян С.М.
(ФИО, подпись)

ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

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 _____