Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Артеменков Михаил Николаевич
Должность: Ректор
Дата подписания: 09.11.2026 18:37:51
Уникальный программный ключ:
a63b74166e8f4fc17bd1e0462725f79e393bfdf9

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Смоленский государственный университет»

Кафедра прикладной математики и информатики

«Утверждаю»
Проректор по учебно-методической
работе
_____ Ю.А. Устименко
«7» сентября 2021 г.

Рабочая программа дисциплины
Б1.В.02 Параллельное программирование прикладных задач

Направление подготовки **01.04.02 Прикладная математика и информатика**
Направленность (профиль): **Прикладные Интернет-технологии**

Форма обучения – очная
Курс – 2
Семестр – 3
Всего зачетных единиц – 3, часов – 108
Форма отчетности: экзамен – 3 семестр

Программу разработал
кандидат технических наук, доцент В.И. Мунерман

Одобрена на заседании кафедры
«1» сентября 2021 г., протокол № 1

Смоленск
2021

1. Место дисциплины в структуре ОП

Дисциплина «Параллельное программирование прикладных задач» относится к части, формируемой участниками образовательных отношений, основной образовательной программы по направлению подготовки 01.04.02 Прикладная математика и информатика.

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.

Для изучения дисциплины «Параллельное программирование прикладных задач» необходимы знания, умения и навыки, полученные при изучении таких дисциплин, как «Сетевые стандарты и технологии», «Взаимодействие в распределенных программных системах», «Web программирование» / «Технологии разработки Web-сервисов».

В результате изучения данной дисциплины у обучающихся формируются знания, умения, навыки и компетенции, необходимые для прохождение технологической (проектно-технологической) и преддипломной практик, выполнения и защиты выпускной квалификационной работы.

2. Планируемые результаты обучения по дисциплине

Компетенция	Индикаторы достижения
ПК-1. Способен осуществлять поиск, анализ, систематизацию научной информации в области прикладной математики и информатики для реализации научно-исследовательских проектов и решения прикладных задач.	**Знает:** теоретические основы и технологии организации научно- исследовательской деятельности. **Умеет**: осуществлять поиск, анализ, систематизацию научной (в том числе юридической) информации в области прикладной математики и информатики для реализации научно-исследовательских проектов и решения прикладных задач, подготавливать и представлять для обсуждения научно-исследовательские работы. **Владеет**: навыками организации и проведения научно-исследовательской деятельности в ходе выполнения профессиональных функций.
ПК-2. Способен проектировать программное и информационное обеспечение компьютерных сетей, вычислительные модели и модели данных для реализации элементов новых (или известных) программных продуктов.	**Знает:** структуру программного обеспечения, основные требования к его проектированию, современные языки программирования, технологии программирования, методики разработки и анализа блоксхем алгоритмов. **Умеет:** проектировать программное и информационное обеспечение компьютерных сетей, выявлять требования к программным продуктам, создавать блок-схемы алгоритмов функционирования разрабатываемых программных продуктов, оценивать их вычислительную сложность.

	Владеет: методами и приемами формализации и алгоритмизации поставленных проектных задач по созданию программного обеспечения.
ПК-3. Способен разрабатывать программное обеспечение, в том числе драйверы устройств, компиляторы, загрузчики, сборщики, системные утилиты	**Знает:** архитектуру аппаратной платформы, синтаксис, принципы и особенности программирования (в том числе кроссплатформенного, распределенного и параллельного программирования), стандартные библиотеки выбранного языка программирования, стандарты реализации интерфейсов устройств, технологии разработки и отладки программных продуктов, принципы информационного построения сетевого взаимодействия, методики тестирования программного обеспечения. **Умеет:** использовать методы и приемы формализации и алгоритмизации поставленных задач, применять выбранный язык программирования для написания программного кода, осуществлять отладку программного обеспечения. **Владеет:** основными методами и приемами разработки и отладки программных продуктов.

3. Содержание дисциплины

1. **Принципы разработки параллельных алгоритмов и программ.** 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).
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.
3. **Уровни распараллеливания вычислений**. Распараллеливание вычислений на уровне команд, выражений, программных модулей, отдельно выполняемых заданий.
4. **Этапы построения параллельных алгоритмов и программ**. Выявление функциональной независимости отдельных фрагментов алгоритма (параллелизм команд). Геометрическое разделение вычислений (параллелизм данных). Иерархическая декомпозиция обработки данных. Выбор параллельного алгоритма. Реализация алгоритма в виде параллельной программы. Построение исполняемой программы для параллельной вычислительной системы.
5. **Технологические аспекты распараллеливания**. Декомпозиция алгоритма на параллельно исполняемые фрагменты вычислений. Распределение заданий по процессорам и балансировка. Синхронизация и взаимоисключение. Организация взаимодействия.
6. **Параллельные численные алгоритмы**. Организация параллельного исполнения рекурсивных вычислений. Проблема рекурсивной зависимости этапов обработки данных. Способы разбиения матриц (горизонтальная, вертикальная, блочные схемы). Методы вычисления произведения матриц с использованием разных схем разбиения матриц.

4. Тематический план

№ п/п	Разделы и темы	Всего часов	Формы занятий		
			лекции	лабораторные занятия	самостоятельная работа
1	Принципы разработки параллельных алгоритмов и программ.	12	2	2	8
2	Оценка коммуникационной трудоемкости параллельных алгоритмов.	13	2	2	9
3	Уровни распараллеливания вычислений	13	2	2	9
4	Этапы построения параллельных алгоритмов и программ	13	2	2	9
5	Технологические аспекты распараллеливания	13	2	2	9
6	Параллельные численные алгоритмы	17	4	4	9
		27	–	–	27
ИТОГО		108	14	14	80

5. Виды образовательной деятельности

Занятия лекционного типа

1. **Принципы разработки параллельных алгоритмов и программ**. 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).

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.

3. **Уровни распараллеливания вычислений.** Распараллеливание вычислений на уровне команд, выражений, программных модулей, отдельно выполняемых заданий.

4. **Этапы построения параллельных алгоритмов и программ.** Выявление функциональной независимости отдельных фрагментов алгоритма (параллелизм команд). Геометрическое разделение вычислений (параллелизм данных). Иерархическая декомпозиция обработки данных. Выбор параллельного алгоритма. Реализация алгоритма в виде параллельной программы. Построение исполняемой программы для параллельной вычислительной системы.

5. **Технологические аспекты распараллеливания.** Декомпозиция алгоритма на параллельно исполняемые фрагменты вычислений. Распределение заданий по процессорам и балансировка. Синхронизация и взаимоисключение. Организация взаимодействия.

6. **Параллельные численные алгоритмы.** Организация параллельного исполнения рекурсивных вычислений. Проблема рекурсивной зависимости этапов обработки данных.

7. **Способы разбиения матриц.** Способы разбиения матриц (горизонтальная, вертикальная, блочные схемы). Методы вычисления произведения матриц с использованием разных схем разбиения матриц.

Занятия семинарского типа - лабораторные работы

Задания к лабораторным работам размещены в ЭИОС СмолГУ.

Лабораторная работа №1. *Уровни распараллеливания вычислений*

1. Распараллеливание вычислений на уровне команд.
2. Распараллеливание вычислений на уровне выражений.
3. Распараллеливание вычислений на уровне программных модулей.
4. Распараллеливание вычислений на уровне отдельно выполняемых заданий.

Лабораторная работа №2. *Этапы построения параллельных алгоритмов и программ*

1. Выявление функциональной независимости отдельных фрагментов алгоритма (параллелизм команд).
2. Геометрическое разделение вычислений (параллелизм данных).
3. Иерархическая декомпозиция обработки данных.
4. Выбор параллельного алгоритма.
5. Реализация алгоритма в виде параллельной программы.
6. Построение исполняемой программы для параллельной вычислительной системы.

Лабораторная работа №3. *Технологические аспекты распараллеливания. Синхронизация и взаимоисключение. Организация взаимодействия.*

1. Декомпозиция алгоритма на параллельно исполняемые фрагменты вычислений.
2. Распределение заданий по процессорам и балансировка.

Лабораторная работа №4. *Технологические аспекты распараллеливания*

1. Синхронизация и взаимоисключение переменных и процессов.
2. Организация взаимодействия процессов.

Лабораторная работа №5-6. *Параллельные численные алгоритмы*

1. Организация параллельного исполнения рекурсивных вычислений.
2. Проблема рекурсивной зависимости этапов обработки данных.

Лабораторная работа №7. *Параллельные численные алгоритмы*

1. Способы разбиения матриц (горизонтальная, вертикальная, блочные схемы).
2. Методы вычисления произведения матриц с использованием разных схем разбиения матриц.

Самостоятельная работа

Текущая самостоятельная работа студента направлена на углубление и закрепление знаний студентов и развитии практических умений. Она заключается в работе с лекционными материалами, поиске и обзоре литературы и электронных источников, информации по заданным темам курса, опережающей самостоятельной работе, в изучении тем, вынесенных на самостоятельную проработку, подготовке к лабораторным занятиям.

Самостоятельная внеаудиторная работа студентов состоит в:

- проработке лекционного материала, составлении конспекта лекций по темам, вынесенным на самостоятельное изучение;
- выполнении домашних заданий.

Списки задач для самостоятельной работ, образцы решений оформления документов размещены в ЭИОС СмолГУ.

Темы для самостоятельного изучения

1. При изучении всех тем студентам предлагаются на самостоятельную проработку свойства и методы используемых объектов.
2. Уточнение свойств и методов изучаемых объектов на сайте MSDN.

6. Критерии оценивания результатов освоения дисциплины (модуля)

6.1. Оценочные средства и критерии оценивания для текущей аттестации

1. Теоретические вопросы

Теоретические вопросы по основным темам курса обсуждаются на лекциях и проверяются на лабораторных занятиях.

Студент демонстрирует сформированность компетенций на итоговом уровне, обнаруживает систематическое и глубокое знание учебного материала, усвоил основную литературу и знаком с дополнительной литературой, рекомендованной программой, умеет свободно выполнять практические задания, предусмотренные программой, свободно оперирует приобретенными знаниями, умениями, применяет их в ситуациях повышенной сложности.	Отлично
Студент демонстрирует сформированность компетенций на среднем уровне: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.	Хорошо
Студент демонстрирует сформированность компетенций на базовом уровне: в ходе контрольных мероприятий допускаются значительные ошибки, проявляется отсутствие отдельных знаний, умений, навыков по некоторым компетенциям, испытывает значительные затруднения при оперировании знаниями.	Удовлетворительно
Студент демонстрирует сформированность компетенций на уровне ниже базового, проявляет недостаточность знаний, умений, навыков. Проявляется практически полное отсутствие знаний, умений, навыков.	Неудовлетворительно

2. Выполнение лабораторной работы

Образец задачи

Разработать параллельную программу для операции разделения в алгоритме внешней сортировки.

Критерии оценивания выполнения лабораторных работ

Нормы оценивания каждой лабораторной работы:

№п/п	Структурная часть работы	Количество баллов (*)
1	Ответ на теоретические вопросы по теме	1 балл

	лабораторной работы	
2	Демонстрация выполнения конкретного задания, предложенного для самостоятельного решения к лабораторной работе	2 балла

(*) с возможностью градации до 0,25 балла.

Шкала оценивания.

Оценка «зачтено» за лабораторную работу выставляется, если набрано не менее 2 баллов, в противном случае за работу выставляется «не зачтено».

3. Проверочная работа

Образец проверочной работы

1. Выполнить сравнение времени последовательного и параллельного (алгоритм Кэннона) умножения матриц (9 потоков).
2. Сравнить расчетное и фактическое ускорения.

Критерии оценивания проверочной работы

Нормы оценивания работы

№ п/п	Структурная часть контрольной работы	Количество баллов (*)
1	Правильно реализован каждый метод решения	1 балл
2	Анализ результатов	2 балла

(*) Возможна градация в 0,25 балла.

Шкала оценивания работы:

п/п	Оценка	Количество баллов
1	Отлично	4,75-5
2	Хорошо	3,75-4,5
3	Удовлетворительно	3-3,5
4	Неудовлетворительно	менее 3

6.2 Оценочные средства и критерии оценивания для промежуточной аттестации

Вопросы к экзамену

1. Оценка эффективности параллельных вычислений. Показатель эффекта распараллеливания (ускорение). Эффективность использования вычислительной системы.
2. Основные характеристики вычислительной системы, влияющие на величины ускорения и эффективности (архитектура, количество процессоров, топология каналов передачи данных).
3. Алгоритмы маршрутизации.
4. Методы передачи данных.
5. Анализ трудоемкости основных операций передачи данных.
6. Передача данных между двумя процессорами сети.
7. Одиночная и множественная рассылка сообщений.
8. Операция циклического сдвига.
9. Методы логического представления топологии коммуникационной среды.
10. Отображение кольцевой топологии и топологии решетки на гиперкуб.
11. Распараллеливание вычислений на уровне команд, выражений, программных модулей, отдельно выполняемых заданий.
12. Выбор параллельного алгоритма.
13. Реализация алгоритма в виде параллельной программы.

14. Построение исполняемой программы для параллельной вычислительной системы.
15. Параллельное исполнение машинной программы.
16. Выбор оптимального алгоритма для конкретной вычислительной системы,
17. Нахождение наилучшей топологии вычислительной системы для решения определенной задачи, распараллеливание существующего алгоритма.
18. Декомпозиция алгоритма на параллельно исполняемые фрагменты вычислений.
19. Распределение заданий по процессорам и балансировка.
20. Синхронизация и взаимоисключение. Организация взаимодействия.
21. Автоматическая векторизация и распараллеливание.
22. Проблемно-ориентированные компиляторы.
23. Общая характеристика стандарта OpenMP.
24. Создание параллельных областей.
25. Разделение вычислительной нагрузки между потоками.
26. Работа с данными. Синхронизация.
27. Выявление функциональной независимости отдельных фрагментов алгоритма (параллелизм команд).
28. Геометрическое разделение вычислений (параллелизм данных). Иерархическая декомпозиция обработки данных.
29. Проблема рекурсивной зависимости этапов обработки данных. Каскадная схема. Метод Оутса.
30. Способы разбиения матриц (горизонтальная, вертикальная, блочные схемы).
31. Методы вычисления произведения матриц с использованием разных схем разбиения матриц. Обеспечение предельно допустимого параллелизма.
32. Ленточный алгоритм.
33. Алгоритм Фокса.
34. Алгоритм Кэннона.
35. Симметричной горизонтальное распределение данных.
36. Параллельная реализация операции Join.

Образец экзаменационного билета

1. Работа с данными. Синхронизация.
2. Разработать программу, параллельно реализующую операцию Join.

Нормы оценивания ответа

№п/п	Структурная часть билета	Количество баллов (по 10-балльной шкале)
1	Теоретический вопрос	4 балла (2 балла за каждый теоретический вопрос)
2	Математическая модель	1 баллов
3	Реализация решения задачи	2 баллов
4	Бонусные баллы (работа в течение семестра)	3 балла

(*) Возможна градация в 0,25 балла.

Шкала оценивания работы:

Уровень качества	Оценка по 10-балльной шкале	Оценка по 5-балльной шкале
высший	10 – превосходно 9 – отлично 8 – почти отлично	5 – «отлично»
высокий	7 – очень хорошо 6 – хорошо	4 – «хорошо»

достаточный	5 – удовлетворительно 4 – почти удовлетворительно	3 – «удовлетворительно»
низкий	3,2,1 – неудовлетворительно	2 – «неудовлетворительно»

7. Перечень основной и дополнительной учебной литературы

7.1 Основная литература

1. Гордеев С. И. Организация баз данных в 2 ч. Часть 1: учебник для вузов / С. И. Гордеев, В. Н. Волошина. – М.: Издательство Юрайт, 2018. – 311 с. – Режим доступа: https://urait.ru/book/organizaciya-baz-dannyh-v-2-ch-chast-1-457145
2. Гордеев С. И. Организация баз данных в 2 ч. Часть 2: учебник для вузов / С. И. Гордеев, В. Н. Волошина. – М.: Издательство Юрайт, 2018. – 501 с. – Режим доступа: https://urait.ru/book/organizaciya-baz-dannyh-v-2-ch-chast-2-473007
3. Парфенов, Ю. П. Постреляционные хранилища данных: учебное пособие для вузов / Ю. П. Парфенов. – М.: Издательство Юрайт, 2018. – 121 с. – Режим доступа: https://urait.ru/bcode/453758

7.2 Дополнительная литература

1. Бройдо, В. Л. Вычислительные системы, сети и телекоммуникации : учеб. пособие : / В. Л. Бройдо, О. П. Ильина. - 3-е изд. - СПб. : Питер, 2008.
2. Максимов, Н. В. Компьютерные сети: учеб. пособие : / Н. В. Максимов, И. И. Попов. - 3-е изд., перераб. и доп. - М. : Форум, 2008.
3. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы : учеб. пособие / В. Г. Олифер, Н. А. Олифер. - 4-е изд. - СПб. : Питер, 2011.
4. Гергель В.П., Стронгин Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных систем. - Н.Новгород, ННГУ, 2001.
5. Богачев К.Ю. Основы параллельного программирования. - М.: БИНОМ. Лаборатория знаний, 2003.
6. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. - СПб.: БХВ-Петербург, 2002.
7. Немнюгин С., Стесик О. Параллельное программирование для многопроцессорных вычислительных систем - СПб.: БХВ-Петербург, 2002.
8. Гергель В.П. (2007). Теория и практика параллельных вычислений. – М.: Интернет-Университет, БИНОМ. Лаборатория знаний.
9. Голицына, О. Л. Программное обеспечение : учеб. пособие / О. Л. Голицына, Т. Л. Партыка, И. И. Попов. - 3-е изд., перераб. и доп. - М. : Форум, 2010.
10. В.Г. Олифер, Н.А. Олифер. Компьютерные сети. Принципы, технологии, протоколы. С-Пб.: Питер, 2001. –672 с.
11. Робачевский. Операционная система Unix. С-Пб.: BHV-Санкт-Петербург, 1997.
12. ОС Unix: Руководство системного администратора. Киев: BHV, 1997.
13. "ATM: технология высокоскоростных сетей", А.Н. Назаров, М.В. Симонов. "Радио и связь", 1997.
14. М.В. Ратынский "Основы сотовой связи". Москва, <Радио и связь>, 1998.
15. Н.Н. Слепов "Синхронные цифровые сети SDH". Москва, <Эко-трэндз>, 1997.
16. Шиндер Д. Основы компьютерных сетей. Диал-ка, 2002,304с.
17. Олифер Н. Сетевые операционные системы. Учебник. Питер, 2001, 544 с.

7.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Автоматически реализуемые свойства (Руководство по программированию на C#). – http://msdn.microsoft.com/ru-ru/library/bb384054.aspx
2. Oliver Sturm Functional Programming in C#: Classic Programming Techniques for Modern Projects. – Wiley. 2011. – 290 с.

3. Практическое руководство. Создание и использование библиотек DLL на языке C# (Руководство по программированию на C#). – http://msdn.microsoft.com/ru-ru/library/3707x96z(v=vs.90).aspx

4. Guido Gybels "Использование ассемблера в Дельфи". Электронная книга. – http://delphiexpert.ru/view_download.php?id=7

5. Доступ к COM серверам Microsoft Office. – http://www.delphisources.ru/pages/faq/base/access_to_microsoft_com_servers.html

6. Интеграция C# с Microsoft Office. – http://easyprog.ru/index.php?option=com_content&task=view&id=818&Itemid=54

7. Работа с базами данных в Delphi с помощью ADO. – http://www.delphisources.ru/pages/articles/2006_year/art_27_11_06/art_27_11_06.html

8. Zeeshan Amjad Using Unmanaged code and assembler in C#. – http://www.codeproject.com/Articles/1392/Using-Unmanaged-code-and-assembler-in-C

8. Материально-техническое обеспечение

Учебная аудитория для проведения занятий лекционного типа, оснащенная стандартной учебной мебелью, интерактивной доской, мультимедиапроектором, ноутбуком и колонками.

Учебная аудитория для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации - компьютерная аудитория с выходом в Интернет.

Помещение для самостоятельной работы – компьютерный класс с доступом к сети «Интернет» и ЭИОС СмолГУ.

9. Программное обеспечение

Kaspersky Endpoint Security для бизнеса Стандартный АО «Лаборатория Касперского», лицензия 1FB6-161215-133553-1-6231.

Microsoft Open License, лицензия 49463448 в составе: Microsoft Windows Professional 7 Russian; Microsoft Office 2010 Russian.

Microsoft Visual Studio.

Microsoft SQL Server Express