

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
федерального государственного бюджетного образовательного учреждения высшего образования
**«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(МИ ВлГУ)**

Кафедра *ПИИ*

«УТВЕРЖДАЮ»
Заместитель директора по УР
_____ Д.Е. Андрианов
_____ 23.05.2023

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Параллельное и распределенное программирование

Направление подготовки

09.04.04 Программная инженерия

Профиль подготовки

Технологии разработки интеллектуальных систем

Семестр	Трудоем- кость, час./зач. ед.	Лек- ции, час.	Практи- ческие занятия, час.	Лабора- торные работы, час.	Консультация, час.	Конт- роль, час.	Всего (контакт- ная работа), час.	СРС, час.	Форма промежу- точного контроля (экз., зач., зач. с оц.)
3	216 / 6	12		48	3,2	2,35	65,55	114,8	Экз.(35,65)
Итого	216 / 6	12		48	3,2	2,35	65,55	114,8	35,65

Муром, 2023 г.

1. Цель освоения дисциплины

Цель дисциплины - подготовка специалиста высокой квалификации, способного выполнять все задачи по освоению параллельного и распределённого программирования. В результате изучения дисциплины студенты ознакомятся с основными концепциями и принципами разработки высокопроизводительных программных систем и приложений.

Задачи дисциплины:

В результате изучения дисциплины студент должен владеть основными методами разработки высокопроизводительных ИС, уметь переходить от постановки задачи к ее модели, определять класс, к которому относится данная задача и находить эффективные методы ее решения и разработки; применять методы оптимизации при решении задач профессиональной деятельности, проектировать распределенные информационные системы, их компоненты и протоколы их взаимодействия; владеть навыками программной реализации распределенных информационных систем, навыками программной реализации систем с параллельной обработкой данных и высокопроизводительных систем, навыками создания трансляторов и интерпретаторов языков программирования, навыками создания служб сетевых протоколов, навыками создания компонент операционных систем и систем реального времени, навыками создания систем обработки текстов, навыками организации промышленного тестирования создаваемого программного обеспечения.

2. Место дисциплины в структуре ОПОП ВО

Базовыми дисциплинами являются "Современные алгоритмы обработки данных", "Методология программной инженерии".

3. Планируемые результаты обучения по дисциплине

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине, в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции	Результаты обучения по дисциплине	
УК-1 Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	УК-1.2 Применяет методы системного подхода и критического анализа проблемных ситуаций	Знает технологию разработки алгоритмов и программ, методы отладки и решения задач на современных многоядерных и распределенных вычислительных системах (УК-1.2)	
	УК-1.3 Вырабатывает стратегию решения поставленной задачи	Умеет работать с современными системами программирования, включая средства профилирования и отладки параллельных и распределенных программ (УК-1.3)	
ОПК-4 Способен применять на практике новые научные принципы и методы исследований	ОПК-4.2 Выполняет декомпозицию алгоритмов для реализации с использованием высокопроизводительных вычислений	Знает методы программной реализации систем с параллельной обработкой данных и высокопроизводительных систем (ОПК-4.2) Умеет использовать методы программной реализации систем с параллельной обработкой данных и	вопросы к устному опросу, вопросы к контрольной работе

		<p>высокопроизводительных систем (ОПК-4.2)</p> <p>Владет методами программной реализации систем с параллельной обработкой данных и высокопроизводительных систем для решения практических задач (ОПК-4.2)</p>	
--	--	---	--

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 6 зачетных единиц, 216 часов.

4.1. Форма обучения: очная

Уровень базового образования: высшее.

Срок обучения 2г.

4.1.1. Структура дисциплины

№ п/п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником							Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация	Контроль		
1	Задача конструирования параллельной программы	3	2		4					12	устный опрос
2	Взаимодействующие процессы	3	2		4					24	контрольная работа
3	Сети Петри	3	2		4					12	устный опрос
4	Дедлоки	3	2		4					9	контрольная работа
5	Семафоры	3			4					13	контрольная работа
6	Мониторы	3			8					12	устный опрос
7	Синтез параллельных программ	3			8					10	контрольная работа
8	Параллельное программирование в системах MPI и OpenMP	3	4		12					22,8	контрольная работа
Всего за семестр		216	12		48		+	3,2	2,35	114,8	Экз.(35,65)
Итого		216	12		48			3,2	2,35	114,8	35,65

4.1.2. Содержание дисциплины

4.1.2.1. Перечень лекций

Семестр 3

Раздел 1. Задача конструирования параллельной программы

Лекция 1.

Представление алгоритма. Требования к представлению параллельного алгоритма. (пассивная) (2 часа).

Раздел 2. Взаимодействующие процессы

Лекция 2.

Последовательные процессы. Выполнение системы процессов. (пассивная) (2 часа).

Раздел 3. Сети Петри

Лекция 3.

Критические секции. Задача взаимного исключения. Определение дедлока. Борьба с дедлоками. (пассивная) (2 часа).

Раздел 4. Дедлоки

Лекция 4.

Определение монитора. Задача "производитель-потребитель". Задача "читатели-писатели". (пассивная) (2 часа).

Раздел 8. Параллельное программирование в системах MPI и OpenMP

Лекция 5.

Асинхронное программирование. Понятие асинхронной программы. (пассивная) (2 часа).

Лекция 6.

Определение MPI. Параллельная программа разделения множеств. Коммуникационно-замкнутые слои параллельной программы. (пассивная) (2 часа).

4.1.2.2. Перечень практических занятий

Не планируется.

4.1.2.3. Перечень лабораторных работ

Семестр 3

Раздел 1. Задача конструирования параллельной программы

Лабораторная 1.

Умножение матрицы на матрицу. (интерактивная) (4 часа).

Раздел 2. Взаимодействующие процессы

Лабораторная 2.

Параллельные алгоритмы решения систем линейных алгебраических уравнений методом Гаусса. (интерактивная) (4 часа).

Раздел 3. Сети Петри

Лабораторная 3.

Первый алгоритм решения СЛАУ методом Гаусса. (пассивная) (4 часа).

Раздел 4. Дедлоки

Лабораторная 4.

Второй алгоритм решения СЛАУ методом Гаусса. (пассивная) (4 часа).

Раздел 5. Семафоры

Лабораторная 5.

Параллельные алгоритмы решения СЛАУ итерационными методами. (пассивная) (4 часа).

Раздел 6. Мониторы

Лабораторная 6.

Параллельный алгоритм решения СЛАУ методом простой итерации. (пассивная) (4 часа).

Лабораторная 7.

Параллельный алгоритм решения СЛАУ методом сопряженных градиентов. (пассивная) (4 часа).

Раздел 7. Синтез параллельных программ

Лабораторная 8.

Программирование на компьютерах с общей памятью и примеры параллельных программ в OpenMP. (пассивная) (4 часа).

Лабораторная 9.

Умножение матрицы на матрицу. (пассивная) (4 часа).

Раздел 8. Параллельное программирование в системах MPI и OpenMP

Лабораторная 10.

Параллельный алгоритм решения СЛАУ методом Гаусса. (пассивная) (4 часа).

Лабораторная 11.

Параллельный алгоритм решения СЛАУ методом сопряженных градиентов. (пассивная) (4 часа).

Лабораторная 12.

Введение в вычисления общего назначения с использованием GPU. (пассивная) (4 часа).

4.1.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Модели параллельного программирования. Мультипроцессорные системы с общей памятью.
2. Модели параллельного программирования. Системы с распределенной памятью. Ускорители.
3. Обзор современных методов и средств отладки и настройки параллельных программ.
4. Архитектуры с распределенной памятью.
5. Введение в Grid - и Cloud - технологии.
6. Моделирование и анализ параллельных вычислений.
7. Теоретические основы параллельных алгоритмов.
8. Оценка коммуникационной трудоемкости параллельных алгоритмов.
9. Сравнение архитектуры CPU и GPU.
10. Эволюция GPU. Использование нескольких GPU.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

4.1.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР Не планируется.

4.1.2.6. Примерный перечень тем курсовых работ (проектов)

1. 1. Параллельное программирование с использованием интерфейса MPI в одномерной краевой задаче.
2. 2. Быстрое преобразование Фурье и его распараллеливание.
3. 3. Оценка эффективности работы параллельных процессоров при реализации простых вычислительных алгоритмов.
4. 4. Метод SOR при распараллеливании численного решения эллиптических задач.
5. 5. Применение стандарта MPI в методе Монте-Карло.
6. 6. Введение в стандарт MPI-2: основные направления модификаций.
7. 7. Организация параллельных процессов для динамических параллельных вычислений.
8. 8. Распараллеливание вычислений минимальных полиномиальных сплайнов.

4.2 Форма обучения: заочная

Уровень базового образования: высшее.

Срок обучения 2г 6м.

Семестр	Трудоем- кость, час./ зач. ед.	Лек- ции, час.	Практи- ческие занятия, час.	Лабора- торные работы, час.	Консуль- тация, час.	Конт- роль, час.	Всего (контак- тная работа), час.	СРС, час.	Форма промежуточного контроля (экз., зач., зач. с оп.)
4	216 / 6	8		8	4	2,35	22,35	185	Экз.(8,65)
Итого	216 / 6	8		8	4	2,35	22,35	185	8,65

4.2.1. Структура дисциплины

№ п\п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником							Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация	Контроль		
1	Задача конструирования параллельной программы	4	2		4					12	устный опрос
2	Взаимодействующие процессы	4	2		4					16	контрольная работа
3	Сети Петри	4	2							16	устный опрос
4	Дедлоки	4	2							17	контрольная работа
5	Семафоры	4								16	контрольная работа
6	Мониторы	4								20	устный опрос
7	Синтез параллельных программ	4								26	контрольная работа
8	Параллельное программирование в системах MPI и OpenMP	4								62	контрольная работа
Всего за семестр		216	8		8		+	4	2,35	185	Экз.(8,65)
Итого		216	8		8			4	2,35	185	8,65

4.2.2. Содержание дисциплины

4.2.2.1. Перечень лекций

Семестр 4

Раздел 1. Задача конструирования параллельной программы

Лекция 1.

Представление алгоритма. Требования к представлению параллельного алгоритма. (пассивная) (2 часа).

Раздел 2. Взаимодействующие процессы

Лекция 2.

Последовательные процессы. Выполнение системы процессов. (пассивная) (2 часа).

Раздел 3. Сети Петри

Лекция 3.

Критические секции. Задача взаимного исключения. Определение дедлока. Борьба с дедлоками. (пассивная) (2 часа).

Раздел 4. Дедлоки

Лекция 4.

Определение монитора. Задача "производитель-потребитель". Задача "читатели-писатели". (пассивная) (2 часа).

4.2.2.2. Перечень практических занятий

Не планируется.

4.2.2.3. Перечень лабораторных работ

Семестр 4

Раздел 1. Задача конструирования параллельной программы

Лабораторная 1.

Умножение матрицы на матрицу. (интерактивная) (4 часа).

Раздел 2. Взаимодействующие процессы

Лабораторная 2.

Параллельные алгоритмы решения систем линейных алгебраических уравнений методом Гаусса. (интерактивная) (4 часа).

4.2.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Модели параллельного программирования. Мультипроцессорные системы с общей памятью.
2. Модели параллельного программирования. Системы с распределенной памятью. Ускорители.
3. Обзор современных методов и средств отладки и настройки параллельных программ.
4. Архитектуры с распределенной памятью.
5. Введение в Grid - и Cloud - технологии.
6. Моделирование и анализ параллельных вычислений.
7. Теоретические основы параллельных алгоритмов.
8. Оценка коммуникационной трудоемкости параллельных алгоритмов.
9. Сравнение архитектуры CPU и GPU.
10. Эволюция GPU. Использование нескольких GPU.
11. Асинхронное программирование. Понятие асинхронной программы. (пассивная).
12. Определение MPI. Параллельная программа разделения множеств. Коммуникационно-замкнутые слои параллельной программы. (пассивная).
13. Первый алгоритм решения СЛАУ методом Гаусса. (пассивная).
14. Второй алгоритм решения СЛАУ методом Гаусса. (пассивная).
15. Параллельные алгоритмы решения СЛАУ итерационными методами. (пассивная).

16. Параллельный алгоритм решения СЛАУ методом простой итерации. (пассивная).
 17. Параллельный алгоритм решения СЛАУ методом сопряженных градиентов. (пассивная).
 18. Программирование на компьютерах с общей памятью и примеры параллельных программ в OpenMP. (пассивная).
 19. Умножение матрицы на матрицу. (пассивная).
 20. Параллельный алгоритм решения СЛАУ методом Гаусса. (пассивная).
 21. Параллельный алгоритм решения СЛАУ методом сопряженных градиентов. (пассивная).
 22. Введение в вычисления общего назначения с использованием GPU. (пассивная).
- Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

4.2.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

Не планируется.

4.2.2.6. Примерный перечень тем курсовых работ (проектов)

1. 1. Параллельное программирование с использованием интерфейса MPI в одномерной краевой задаче.
2. 2. Быстрое преобразование Фурье и его распараллеливание.
3. 3. Оценка эффективности работы параллельных процессоров при реализации простых вычислительных алгоритмов.
4. 4. Метод SOR при распараллеливании численного решения эллиптических задач.
5. 5. Применение стандарта MPI в методе Монте-Карло.
6. 6. Введение в стандарт MPI-2: основные направления модификаций.
7. 7. Организация параллельных процессов для динамических параллельных вычислений.
8. 8. Распараллеливание вычислений минимальных полиномиальных сплайнов.

5. Образовательные технологии

В процессе изучения дисциплины применяется контактная технология преподавания (за исключением самостоятельно изучаемых студентами вопросов). При проведении лабораторных работ применяется имитационный или симуляционный подход. Шаги решения задач студентам демонстрируются при помощи мультимедийной техники. В дальнейшем студенты самостоятельно решают аналогичные задания.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

Фонды оценочных материалов (средств) приведены в приложении.

7. Учебно-методическое и информационное обеспечение дисциплины.

7.1. Основная учебно-методическая литература по дисциплине

1. Биллиг, В. А. Параллельные вычисления и многопоточное программирование : учебник / В. А. Биллиг. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 310 с. — ISBN 978-5-4497-0936-3. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/102044.html> (дата обращения: 22.08.2021). — Режим доступа: для авторизир. пользователей - <https://www.iprbookshop.ru/102044.html>
2. Гололобов, С. В. Параллельные алгоритмы вычислительной алгебры : учебно-методическое пособие / С. В. Гололобов, А. А. Калинин. — Новосибирск : Новосибирский государственный университет, 2019. — 76 с. — ISBN 978-5-4437-0958-1. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL:

<https://www.iprbookshop.ru/93820.html> (дата обращения: 22.08.2021). — Режим доступа: для авторизир. пользователей - <https://www.iprbookshop.ru/93820.html>

3. Туральчук, К. А. Параллельное программирование с помощью языка C# / К. А. Туральчук. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019. — 189 с. — ISBN 978-5-4486-0506-2. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/79714.html> (дата обращения: 22.08.2021). — Режим доступа: для авторизир. пользователей - <https://www.iprbookshop.ru/79714.html>

4. Алексеев, А. А. Основы параллельного программирования с использованием Visual Studio 2010 : учебное пособие / А. А. Алексеев. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 330 с. — ISBN 978-5-4497-0341-5. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/89456.html> (дата обращения: 22.08.2021). — Режим доступа: для авторизир. пользователей - <https://www.iprbookshop.ru/89456.html>

7.2. Дополнительная учебно-методическая литература по дисциплине

1. Гергель, В. П. Теория и практика параллельных вычислений : учебное пособие / В. П. Гергель. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 500 с. — ISBN 978-5-4497-0389-7. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/89478.html> (дата обращения: 22.08.2021). — Режим доступа: для авторизир. пользователей - <https://www.iprbookshop.ru/89478.html>

7.3. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

В образовательном процессе используются информационные технологии, реализованные на основе информационно-образовательного портала института (www.mivlgu.ru/iop), и инфокоммуникационной сети института:

- предоставление учебно-методических материалов в электронном виде;
- взаимодействие участников образовательного процесса через локальную сеть института и Интернет;
- предоставление сведений о результатах учебной деятельности в электронном личном кабинете обучающегося.

Информационные справочные системы:

Электронная библиотека ВлГУ (<http://dspace.www1.vlsu.ru>);

Электронная библиотечная системы "IPRBooks" (<http://www.iprbookshop.ru/>).

Программное обеспечение:

LibreOffice (Mozilla Public License v2.0)

Microsoft Windows 10 Professional (Программа Microsoft Azure Dev Tools for Teaching (Order Number: IM126433))

Microsoft Visual Studio (Программа Microsoft Azure Dev Tools for Teaching (Order Number: IM126433))

7.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

iprbookshop.ru

dspace.www1.vlsu.ru);

mivlgu.ru/iop

8. Материально-техническое обеспечение дисциплины

Лаборатория программного обеспечения и сопровождения компьютерных систем

Сервер «Ай Тек» на базе 2 процессоров Intel Xeon; 12 шт. компьютеров Intel Core i5-10400 2,9 GHz/ 8Gb DDR-4/ SSD-480 Gb/ Hiper 21,5'; интерактивная доска SMART Board 480 со встроенным проектором V25; маршрутизатор Gigabit Switch TEG-S16S. Маркерная доска. Доступ к сети Интернет.

9. Методические указания по освоению дисциплины

Для успешного освоения теоретического материала обучающийся: знакомится со списком рекомендуемой основной и дополнительной литературы; уточняет у преподавателя, каким дополнительным пособиям следует отдать предпочтение; ведет конспект лекций и прорабатывает лекционный материал, пользуясь как конспектом, так и учебными пособиями.

До выполнения лабораторных работ обучающийся изучает соответствующий раздел теории. Перед занятием студент знакомится с описанием заданий для выполнения работы, внимательно изучает содержание и порядок проведения лабораторной работы. Лабораторная работа проводится в компьютерном классе. Обучающиеся выполняют индивидуальную задачу компьютерного моделирования в соответствии с заданием на лабораторную работу. Полученные результаты исследований сводятся в отчет и защищаются по традиционной методике в классе на следующем лабораторном занятии. Необходимый теоретический материал, индивидуальное задание, шаги выполнения лабораторной работы и требование к отчету приведены в методических указаниях, размещенных на информационно-образовательном портале института.

Самостоятельная работа оказывает важное влияние на формирование личности будущего специалиста, она планируется обучающимся самостоятельно. Каждый обучающийся самостоятельно определяет режим своей работы и меру труда, затрачиваемого на овладение учебным содержанием дисциплины. Он выполняет внеаудиторную работу и изучение разделов, выносимых на самостоятельную работу, по личному индивидуальному плану, в зависимости от его подготовки, времени и других условий.

Курсовая работа выполняется в соответствии с методическими указаниями на курсовую работу. Обучающийся выбирает одну из указанных в перечне тем курсовых работ, исходя из своих интересов, наличия соответствующих литературных и иных источников. В ходе выполнения курсовой работы преподаватель проводит консультации обучающегося. На заключительном этапе обучающийся оформляет пояснительную записку к курсовой работе и выполняет ее защиту в присутствии комиссии из преподавателей кафедры.

Форма заключительного контроля при промежуточной аттестации – экзамен. Для проведения промежуточной аттестации по дисциплине разработаны фонд оценочных средств и балльно-рейтинговая система оценки учебной деятельности студентов. Оценка по дисциплине выставляется в информационной системе и носит интегрированный характер, учитывающий результаты оценивания участия студентов в аудиторных занятиях, качества и своевременности выполнения заданий в ходе изучения дисциплины и промежуточной аттестации.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению
09.04.04 Программная инженерия и профилю подготовки *Технологии разработки интеллектуальных систем*
Рабочую программу составил *к.т.н., доцент Колтаков А.А.*_____

Программа рассмотрена и одобрена на заседании кафедры *ПИИ*

протокол № 13 от 05.05.2023 года.

Заведующий кафедрой *ПИИ* _____ *Жизняков А.Л.*

(Подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии факультета

протокол № 9 от 19.05.2023 года.

Председатель комиссии ФИТР _____ *Рыжкова М.Н.*

(Подпись)

(Ф.И.О.)

Фонд оценочных материалов (средств) по дисциплине
Параллельное и распределенное программирование

1. Оценочные материалы для проведения текущего контроля успеваемости по дисциплине

Семестр 1

Рейтинг-контроль 1:

Перечень вопросов:

1. Параллельные вычисления. Общий смысл.
2. Мультипроцессоры и мультикомпьютеры
3. Обзор программной нотации
4. Программирование с разделяемыми переменными. Процессы и синхронизация.
5. Программирование с разделяемыми переменными. Блокировки и барьеры.
6. Программирование с разделяемыми переменными. Семафоры. Мониторы.
7. Особенности современных параллельных архитектур. Виды параллельности. Модели параллельного программирования. Характеристики параллельной программы: ускорение, масштабируемость, эффективность.
8. Мультипроцессорные системы с общей памятью. Общая архитектура и подходы к параллельному программированию. Проблемы, возникающие при организации доступа к общим ресурсам.
9. Механизмы управления доступом к критическим ресурсам: активное ожидание, семафоры, мониторы.
10. Библиотека Pthreads: управление потоками, управление мютексами, управление условными переменными.
11. Библиотека OpenMP: модель программирования, синтаксис, понятие параллельной области и режима ее выполнения.
12. Директивы OpenMP. Параллельные циклы for/do. Параметры директив.
13. Директивы OpenMP. Параллельные секции, директива single. Параметры директив.
14. Системные переменные и системные подпрограммы времени выполнения библиотеки OpenMP.
15. Распараллеливание циклов. Зависимости по данным.
16. Синхронизационные конструкции OpenMP: critical, atomic, barrier, master, ordered, flush.
17. Ошибки, возникающие при программировании с общей памятью: состязание и взаимоблокировка.

Задачи:

Задание 1.

Напишите программу, в которой создается k нитей, и каждая нить выводит на экран свой номер и общее количество нитей в параллельной области в формате: I am <Номер нити> thread from <Количество нитей> threads!

Задание 2.

Изучите конструкции для управления работой с данными shared и private. Напишите программу, в которой создается k нитей, и каждая нить выводит на экран свой номер через переменную rank следующим образом:

```
rank = omp_get_thread_num();  
printf("I am %d thread.\n", rank);
```

Экспериментами определите, общей или частной должна быть переменная rank.

Задание 3.

Напишите программу, в которой две нити параллельно вычисляют сумму чисел от 1 до N. Распределите работу по нитям с помощью оператора if языка C. Для сложения результатов вычисления нитей воспользуйтесь OpenMP-параметром reduction.

Рейтинг-контроль 2:

Перечень вопросов:

1. Распределенное программирование. Передача сообщений.
2. Распределенное программирование. Удаленный вызов процедур и рандеву.
3. Распределенное программирование. Модели взаимодействия процессов.
4. Распределенное программирование. Реализация языковых механизмов
5. Синхронное параллельное программирование. Научные вычисления.
6. Синхронное параллельное программирование. Языки, компиляторы, библиотеки и инструментальные средства
7. Кластерные высокопроизводительные вычислительные системы: требования к архитектуре. Коммуникационное оборудование и аппаратно-программные платформы.
8. Модель вычислительной системы с распределенной памятью – LogGP.
9. Программирование в терминах обмена сообщениями. Основные понятия библиотеки MPI: инициализация и выход, понятие коммуникатора, сообщение.
10. Блокирующие коммуникации точка-точка: функции отправки и приема.
11. Неблокирующие коммуникации точка-точка: функции отправки и приема, подпрограммы ожидания/проверки.
12. Типы данных MPI (базовые и производные). Соответствие типов данных MPI и типов данных языков C и Fortran. Конструирование производных типов: непрерывно размещенный, векторный, структурный.
13. Коллективные операции распределения данных в MPI: MPI_Bcast, MPI_Scatter, MPI_Gather, MPI_Allgather, MPI_Alltoall. Операции глобальной редукции: minloc и maxloc; определенные пользователем.
14. Односторонние коммуникации.
15. Основные конструкции технологии Cuda.
16. Сравнительный анализ стандартов OpenCL и OpenACC.
17. Средства отладки и настройки параллельных программ.

Задачи:

Задание 1.

Изучите OpenMP-директиву параллельного выполнения цикла for. Напишите программу, в которой k нитей параллельно вычисляют сумму чисел от 1 до N. Распределите работу по нитям с помощью OpenMP-директивы for.

Задание 2.

Изучите параметр schedule директивы for. Модифицируйте программу «Сумма чисел» из задания 4 таким образом, чтобы дополнительно выводилось на экран сообщение о том, какая нить, какую итерацию цикла выполняет:

[<Номер нити>]: calculation of the iteration number <Номер итерации>.

Задайте k = 4, N = 10.

Задание 3.

Напишите OpenMP-программу, которая вычисляет число

Общее распределение баллов текущего контроля по видам учебных работ для студентов

Рейтинг-контроль 1	Устный опрос 20 вопросов, 2 практических задания	20 баллов
Рейтинг-контроль 2	Контрольная работа 5 вопросов, 2 практических	20 баллов

	задания	
Рейтинг-контроль 3	Контрольная работа 5 вопросов, 2 практических задания	20 баллов
Посещение занятий студентом		10 баллов
Дополнительные баллы (бонусы)		20 баллов
Выполнение семестрового плана самостоятельной работы		20 баллов

2. Промежуточная аттестация по дисциплине

Перечень вопросов к экзамену / зачету / зачету с оценкой.

Перечень практических задач / заданий к экзамену / зачету / зачету с оценкой (при наличии)

ОПК-4

1. Напишите OpenMP-программу, которая вычисляет произведение двух квадратных матриц
2. Общие и частные переменные в OpenMP: программа «Скрытая ошибка»
3. Общие и частные переменные в OpenMP: параметр reduction
4. Распараллеливание циклов в OpenMP: программа «Сумма чисел»
5. Распараллеливание циклов в OpenMP: параметр schedule
6. Распараллеливание циклов в OpenMP: программа "Число

Методические материалы, характеризующие процедуры оценивания

На основе перечня вопросов формируются индивидуальные задания для студентов: 6 вопросов из блока 1, 4 вопроса из блока 2. Результатом итоговой контрольной работы является балл, рассчитанный на основе количества правильных ответов. С учетом индивидуального семестрового рейтинга студента формируется итоговый балл по курсу.

Максимальная сумма баллов, набираемая студентом по дисциплине равна 100.

Оценка в баллах	Оценка по шкале	Обоснование	Уровень сформированности компетенций
Более 80	«Отлично»	Содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	Высокий уровень
66-80	«Хорошо»	Содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом	Продвинутый уровень

		сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками	
50-65	«Удовлетворительно»	Содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки	<i>Пороговый уровень</i>
Менее 50	«Неудовлетворительно»	Содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки	<i>Компетенции не сформированы</i>

3. Задания в тестовой форме по дисциплине

Примеры заданий:

ОПК-4

1. Максимальная пиковая производительность наиболее мощных современных параллельных вычислительных систем измеряется:

- в единицах EFLOPs
- в десятках PFLOPs
- в единицах PFLOPs
- в сотнях TFLOPs?

2. Производительность компьютера, достигнутая при выполнении некоторой программы, выражена в TFLOPs. Это значение говорит о:

- среднем количестве операций над вещественными данными, представленными в форме с фиксированной запятой, выполненных за секунду в процессе обработки данной программы;
- общем числе команд, выполненных за время работы программы;
- средней скорости выполнения данным компьютером арифметических операций над вещественными числами, представленными в форме с плавающей запятой;
- средней скорости выполнения данным компьютером арифметических операций над вещественными числами, представленными в форме с плавающей запятой, достигнутой при выполнении данной программы;
- высокой реальной производительности данного компьютера.

3. Умножение двух квадратных плотных вещественных матриц компьютер выполнил за 5 сек с производительностью 50 GFLOPs. Какого размера были матрицы:

- 500*500
- 1000*1000
- 2000*2000
- 5000*5000

- 7000*7000
- верного ответа нет

4. Отметьте правильные утверждения на тему машинного представления чисел в современных ЭВМ:

- все существовавшие до сих пор ЭВМ используют в качестве базовой двоичную систему счисления;
- машинное эpsilon в основном определяется длиной мантиссы в представлении вещественных чисел;
- мантисса числа в двоичном представлении - та же, что и мантисса его десятичного логарифма;
- машинные ноль и эpsilon не могут быть получены с помощью фортран-программы, их следует найти в документации к компьютеру;
- машинное сложение коммутативно; машинное сложение ассоциативно; машинное умножение коммутативно;
- машинное умножение ассоциативно?

5. Архитектура компьютеров. Отметьте правильные утверждения: в SMP-компьютерах все процессоры равноправны; архитектуры NUMA и ccNUMA позволяют сохранить единое адресное пространство для параллельной программы; кэш-память явилась причиной возникновения архитектуры NUMA; поиск команд, которые можно выполнять параллельно, в суперскалярных процессорах происходит во время работы программы; параллелизм в классических VLIW-компьютерах выделяется компилятором.

6. Отметьте правильные утверждения про компьютеры: классификация Флинна содержит 3 типа компьютеров; классификация Флинна содержит 4 типа компьютеров; классификация Флинна содержит 6 типов компьютеров; одним из признаков векторно-конвейерного компьютера является многопроцессорность; одним из признаков векторно-конвейерного компьютера является наличие хотя бы одного конвейера; одним из признаков векторно-конвейерного компьютера является наличие векторных регистров; концепция неограниченного параллелизма при развитии компьютерной техники в отдалённом будущем может стать реальностью.

7. Какие из технологических этапов присущи только парадигме параллельного программирования: построение математической модели, декомпозиция, аранжировка, написание программы?

8. Чем декомпозиция по данным отличается от декомпозиции по вычислениям?

9. В чем заключаются достоинства и недостатки статического и динамического способов назначения задач виртуальным исполнителям?

ПК-11

1. Каковы основные цели этапа назначения: сокращение загрузки исполнителей, балансировка загрузки исполнителей, сокращение обменов данными между исполнителями, равномерный обмен данными между исполнителями, сокращение накладных расходов на назначение?

2. В конвейерном устройстве есть 7 ступеней, срабатывающих за одну единицу времени каждая. За сколько единиц времени это устройство обработает 7 пар аргументов: 1, 3, 7, 8, 13, 14, верного ответа нет?

3. Есть два конвейерных ФУ: сложение (4 ступени) и умножение (6 ступеней), все ступени срабатывают за один такт. За сколько тактов будет выполнена векторная операция $A_i = B_i + C_i * s$, $i=1,2,...,60$, с использованием данных устройств в режиме с зацеплением ФУ: 60, 69, 70, 128, 130, 240, 600?

4. Есть два конвейерных ФУ: сложение (4 ступени) и умножение (6 ступеней), все ступени срабатывают за один такт. За сколько тактов будет выполнена векторная операция $A_i = B_i + C_i * s$, $i=1,2,...,60$, с использованием данных устройств без зацепления ФУ: 60, 69, 70, 128, 130, 240, 600?

5 Действительные числа в машинном представлении: всегда хранятся точно; всегда хранятся с ненулевой ошибкой округления; хранились на всех существовавших вычислительных системах в двоичном представлении; имеют относительную ошибку округления не более машинного нуля; имеют абсолютную ошибку округления не более машинного эпсилон; иногда хранятся точно?

6. Конвейерное ФУ деления состоит из пяти ступеней, срабатывающих за 2, 5, 3, 1 и 1 такт соответственно. Чему равно наименьшее число тактов, за которое можно обработать 40 пар аргументов на данном устройстве: 1, 5, 12, 40, 207, 212, 480, верного ответа нет?

Полный перечень тестовых заданий с указанием правильных ответов, размещен в банке вопросов на информационно-образовательном портале института по ссылке <https://www.mivlgu.ru/iop/question/bank/managecategories/category.php?courseid=3043>

Оценка рассчитывается как процент правильно выполненных тестовых заданий из их общего числа.