

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель освоения дисциплины - формирование у студентов представления о создании программного обеспечения различных параллельных и распределенных вычислительных систем.

Задачи:

- 1) Изучение общих вопросов распараллеливания вычислительных алгоритмов.
- 2) Изучить инструментарий распараллеливания на системах с общей памятью.
- 3) Знакомство с возможностями решения вычислительных задач на графических процессорах.
- 4) Познакомиться с особенностями написания программ в распределенных системах.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина **Программное обеспечение параллельных и распределенных вычислительных систем** относится к части формируемой участниками образовательных отношений.

Пререквизиты дисциплины: "Операционные системы", "Программирование", "Сети и телекоммуникации".

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
<i>ПК-2 Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности</i>	Частичное	Знать основные принципы построения параллельных и распределенных вычислительных систем и устройств. Уметь проектировать систему на базе типовых функциональных узлов. Владеть базовыми навыками формирования электрических структурных, функциональных и принципиальных схем, в том числе, с использованием технической и справочной литературы.
<i>ПК-3 Способен разрабатывать компоненты системных программных продуктов</i>	Частичное	Знать возможности современных параллельных и распределенных вычислительных систем и их характеристики Уметь применять средства настройки и управления современными операционными системами Владеть инструментальными средствами создания программных продуктов
<i>ПК-4 Способен организовать выполнение научно-исследовательских работ по закреплённой тематике. Способен организовать проведение работ по выполнению научно-исследовательских и опытно-конструкторских работ.</i>	Частичное	Знать: способы распараллеливания различных алгоритмов. Уметь: планировать задачи по переводу программного кода из последовательного в параллельный и распределять данные между вычислительными узлами. Владеть: программными средствами планирования, создания, тестирования и отладки параллельных и распределенных программ.

4. ОБЪЕМ И СТРУКТУРА ДИСЦИПЛИНЫ

Трудоемкость дисциплины составляет 3 зачетных единицы, 108 часов

№ 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	Архитектуры параллельных вычислительных систем. Терминология. Режимы выполнения.	8	1	2			5	1/50	
2	Параллельные вычисления. Распараллеливание циклов.	8	2	2		4	5	3/50	
3	Этапы разработки параллельного алгоритма.	8	3	2		4	5	3/50	Рейтинг-контроль №1
4	Технология OpenMP. Директивы и функции. Модель данных.	8	4	2		4	5	3/50	
5	OpenMP секции (sections) и задачи (task). Примеры программ.	8	5	2		6	5	4/50	
6	Вычисления на графических ускорителях. CUDA архитектурные особенности	8	6	2			5	1/50	Рейтинг-контроль №2
7	CUDA примеры программирования	8	7	2			5	1/50	
8	Программирование на системах с распределенной памятью. Технология MPI.	8	8	2			5	1/50	
9	Примеры программирования с технологией MPI.	8	9	2			5	1/50	Рейтинг-контроль №3
Всего за 8 семестр:				18		18	4 5	18/50	Экзамен (27)
Наличие в дисциплине КП/КР									Нет
Итого по дисциплине				18		18	4 5	18/50	Экзамен (27)

Содержание лекционных занятий по дисциплине

Раздел 1. Архитектуры параллельных вычислительных систем. Терминология. Режимы выполнения.

Раздел 2. Параллельные вычисления. Распараллеливание циклов.

Раздел 3. Этапы разработки параллельного алгоритма.

Раздел 4. Технология OpenMP. Директивы и функции. Модель данных.

Раздел 5. OpenMP секции (sections) и задачи (task). Примеры программ.

Раздел 6. Вычисления на графических ускорителях. CUDA архитектурные особенности

Раздел 7. CUDA примеры программирования

Раздел 8. Программирование на системах с распределенной памятью. Технология MPI

Раздел 9. Примеры программирования с технологией MPI

Содержание лабораторных занятий по дисциплине

1. Знакомство с технологией OpenMP. Создание базового проекта.
2. Освоение и выполнение базовые функции OpenMP в параллельных и последовательных областях.
3. Распараллеливание с использованием секций.
4. Распараллеливание с использованием задач.

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В преподавании дисциплины «Программное обеспечение параллельных и распределенных вычислительных систем» используются разнообразные образовательные технологии как традиционные, так и с применением активных и интерактивных методов обучения.

Активные и интерактивные методы обучения:

- *Интерактивная лекция (тема №1-9);*

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

Текущий контроль успеваемости

Вопросы рейтинг-контроля № 1

Классифицировать по степени распараллеливания ОКОД, ОКМД, МКОД, МКМД

Раскрыть термины: многозадачный режим; параллельные вычисления; распределенные вычисления.

Что требуется для распараллеливания.

Как произвести полное распараллеливание.

Виды распределения задач, при распараллеливании циклов.

Методы разбиения задач для распараллеливания.

Этапы разработки параллельного алгоритма.

Вопросы рейтинг-контроля № 2

Для чего подходит технология OpenMP.

Инкрементальное распараллеливание.

Директива `#pragma omp parallel`

Директива `#pragma omp for`

Директива `#pragma omp section`

Директива `#pragma omp task`

Преобразования циклов.

Проблемы, возникающие при создании параллельных программ.

Гонки за данными.

Блокировки при распараллеливании.

Вопросы рейтинг-контроля № 3

Архитектуры графических процессоров.

Состав программного обеспечения CUDA.

Структура памяти графического процессора.

Терминология CUDA. Устройство, хост, ядро.

Базовый CUDA API для работы с памятью GPU. `cudaMalloc()`; `cudaFree()`; `cudaMemcpy()`.

Модель передачи сообщений MPI.

Режимы запуска параллельных программ SPMD и MPMD.

Особенности программы MPI. Структура MPI-программы.

Коммуникаторы MPI.

Виды коммуникационных функций "точка-точка" MPI.
Неблокирующие отправка и получение сообщения. Завершение неблокирующих обменов MPI.
Режимы отправки сообщений "точка-точка" MPI.
Разрешение тупиков MPI. Совмещение приема и передачи сообщения MPI.
Коллективные операции MPI. Барьерная синхронизация.
Широковещательная рассылка MPI.
Глобальные операции над данными MPI.
Типы данных в MPI.

Промежуточная аттестация по итогам освоения дисциплины

Вопросы к экзамену

Классифицировать по степени распараллеливания ОКОД, ОКМД, МКОД, МКМД
Раскрыть термины: многозадачный режим; параллельные вычисления; распределенные вычисления. Что требуется для распараллеливания.
Как произвести полное распараллеливание.
Виды распределения задач, при распараллеливании циклов.
Методы разбиения задач для распараллеливания.
Этапы разработки параллельного алгоритма.
Для чего подходит технология OpenMP. Инкрементальное распараллеливание.
Директива #pragma omp parallel
Директива #pragma omp for
Директива #pragma omp section
Директива #pragma omp task
Преобразования циклов.
Проблемы, возникающие при создании параллельных программ. Гонки за данными. Блокировки при распараллеливании.
Архитектуры графических процессоров. Состав программного обеспечения CUDA. Структура памяти графического процессора.
Терминология CUDA. Устройство, хост, ядро. Базовый CUDA API для работы с памятью GPU. cudaMalloc(); cudaFree(); cudaMemcpy().
Модель передачи сообщений MPI. Режимы запуска параллельных программ SPMD и MPMD. Особенности программы MPI. Структура MPI-программы. Коммуникаторы MPI.
Виды коммуникационных функций "точка-точка" MPI. Неблокирующие отправка и получение сообщения. Завершение неблокирующих обменов MPI. Режимы отправки сообщений "точка-точка" MPI.
Разрешение тупиков MPI. Совмещение приема и передачи сообщения MPI.
Коллективные операции MPI. Барьерная синхронизация. Широковещательная рассылка MPI. Глобальные операции над данными MPI. Типы данных в MPI.

Самостоятельная работа обучающегося

Целью самостоятельной работы являются формирование личности студента, развитие его способности к самообучению и повышению своего профессионального уровня.
Самостоятельная работа заключается в изучении содержания тем курса по учебникам и дополнительной литературе, подготовке к лабораторным занятиям, к экзамену.

При освоении дисциплины используются следующие сочетания видов учебной работы с методами и формами активизации познавательной деятельности бакалавров для достижения запланированных результатов обучения и формирования компетенций:

- самостоятельное изучение теоретического материала дисциплины с использованием Internet-ресурсов, информационных баз, методических разработок, специальной учебной и научной литературы;
- самостоятельная работа студента, направленная на углубление и закрепление знаний, а также развитие практических умений заключается в работе бакалавров с материалом, поиск и анализ литературы и электронных источников информации по заданной проблеме, выполнении домашних заданий, переводе материалов из тематических информационных ресурсов с иностранных языков, изучении тем, вынесенных на самостоятельную проработку, изучении теоретического материала, подготовке к контрольным мероприятиям и экзамену.

- Творческая проблемно-ориентированная самостоятельная работа направлена на развитие интеллектуальных умений, комплекса компетенций, повышение творческого потенциала студентов заключается в поиске, анализе, структурировании и презентации информации, анализе научных публикаций по определенной теме.

Темы самостоятельной работы студентов

Архитектуры параллельных вычислительных систем. Терминология. Режимы выполнения.
 Параллельные вычисления. Распараллеливание циклов.
 Этапы разработки параллельного алгоритма.
 Технология OpenMP. Директивы и функции. Модель данных.
 OpenMP секции (sections) и задачи (task).
 Вычисления на графических ускорителях.
 Архитектурные особенности CUDA.
 Программирование CUDA.
 Программирование на системах с распределенной памятью.
 Программирование с технологией MPI.

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 ВО	Наличие в электронной библиотеке ВлГУ
1	2	3	4
Основная литература			
1. Малявко, А. А. Параллельное программирование на основе технологий OpenMP, MPI, CUDA : учеб. пособие / Малявко А. А. - Новосибирск : Изд-во НГТУ, - 116 с. - ISBN 978-5-7782-2614-2. - Текст : электронный	2015		https://www.studentlibrary.ru/book/ISBN9785778226142.html
2. Арыков, С. Б. Параллельное программирование над общей памятью. OpenMP : учебное пособие / С. Б. Арыков, М. А. Городничев, Г. А. Шукин. - Новосибирск : НГТУ, - 95 с. - ISBN 978-5-7782-3796-4. - Текст : электронный	2019		https://www.studentlibrary.ru/book/ISBN9785778237964.html
3. Тумаков, Д. Н. Технология программирования CUDA: учеб. пособие / Д. Н. Тумаков, Д. Е. Чикрин, А. А. Егорчев, С. В. Голоусов - Казань : Изд-во Казан. ун-та, - 112 с. - ISBN 978-5-00019-913-8. - Текст : электронный	2017		https://www.studentlibrary.ru/book/ISBN9785000199138.html
Дополнительная литература			
1. Биллиг, В. А. Параллельные вычисления и многопоточное	2016		https://www.studentlibrary.ru/book/intuit_242.html

программирование / Биллиг В. А. - Москва : Национальный Открытый Университет "ИНТУИТ", - Текст : электронный			
2. Антонов, А. С. Параллельное программирование с использованием технологии MPI / Антонов А. С. - Москва : Национальный Открытый Университет "ИНТУИТ", - Текст : электронный	2016		https://www.studentlibrary.ru/book/intuit_240.html
3. Тоуманен, Б. Программирование GPU при помощи Python и CUDA / Тоуманен Б. , пер. с англ. А. В. Борескова. - Москва : ДМК Пресс, - 254 с. - ISBN 978-5-97060-821-0. - Текст : электронный	2020		https://www.studentlibrary.ru/book/ISBN9785970608210.html

7.2. Периодические издания

Журналы (<https://elibrary.ru/>):

1. Вестник компьютерных и информационных технологий
2. Вычислительные технологии
3. Известия вузов: электроника
4. Радиотехнические и телекоммуникационные системы

7.3. Интернет-ресурсы

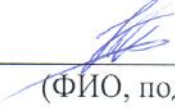
1. Электронная библиотека www.citforum.ru
2. Электронная энциклопедия wikipedia.org
3. Спецификации OpenMP <https://www.openmp.org/specifications/>

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для реализации данной дисциплины имеются специальные помещения для проведения занятий лекционного типа, занятий лабораторного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы. Лабораторные работы проводятся в аудиториях 401-2, 416-2 и 412-2.

Рабочую программу составил доцент кафедры ВТиСУ Куликов И.В.

Рецензент
(представитель работодателя)  Генеральный директор ООО «Диagramма»
Протягов И.В.

Программа рассмотрена и одобрена на заседании кафедры ВТ и СУ
Протокол № 6 от 25.02.2021 года
Заведующий кафедрой ВТ и СУ  Ланцов В.Н.
(ФИО, подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии
направления 09.03.01
Протокол № 2 от 25.02.2021 года
Председатель комиссии  Ланцов В.Н.
(ФИО, подпись)