**Программа учебной дисциплины
«Параллельные и распределенные вычисления»**

Утверждена

Академическим советом ООП

Протокол № от «\_\_»\_\_\_\_\_20\_\_ г.

|  |  |
| --- | --- |
| Автор  | Сухорослов О.В. |
| Число кредитов  |  |
| Контактная работа (час.)  | 60 |
| Самостоятельная работа (час.)  | 130 |
| Курс  | Параллельные и распределенные вычисления |
| Формат изучения дисциплины | Без использования онлайн курса |

1. **ЦЕЛЬ, РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ И ПРЕРЕКВИЗИТЫ**

Целями освоения дисциплины «Параллельные и распределенные вычисления» являются овладение основными концепциями параллельных и распределенных вычислений, классами высокопроизводительных систем, принципами реализации параллельных алгоритмов и используемыми моделями программирования, а также получение навыков практического использования соответствующих технологий и систем при решении прикладных задач..

В результате освоения дисциплины студент должен:

**знать:**

- принципы разработки параллельных алгоритмов, основные модели параллельного программирования и распределенных вычислений, их области применения и ограничения.

**уметь:**

- обосновать выбор той или иной системы, подхода или технологии для решения поставленной задачи;

- реализовывать программы для различных классов высокопроизводительных систем с применением современных технологий параллельных и распределенных вычислений, оценивать эффективность и выполнять оптимизацию полученных реализаций;

- применять полученные знания для решения прикладных задач.

**владеть:**

- общей информацией о проблематике параллельных и распределенных вычислений;

- методами высокопроизводительных вычислений, технологиями параллельных и распределенных вычислений;

- принципами разработки параллельных алгоритмов, моделями параллельного программирования и распределенных вычислений.

Изучение дисциплины «Параллельные и распределенные вычисления» базируется на следующих дисциплинах:

- Алгоритмы и структуры данных;

- Основы и методология программирования;

- Алгоритмы и структуры данных — 2;

- Архитектура компьютера и операционные системы.

Для освоения учебной дисциплины студенты должны владеть следующими знаниями и компетенциями:

* знать основные алгоритмы и структуры данных, принципы их проектирования и анализа;
* уметь создавать программы, решающие задачи по заданному алгоритму, на языках С++ и Python;
* иметь навыки формализации и решения практических задач по программированию, использования технологий программирования и интегрированных сред;
* знать архитектуру компьютеров, принципы функционирования операционных систем и компьютерных сетей;
* уметь логически анализировать информацию и пользоваться документацией.

Основные положения дисциплины должны быть использованы в дальнейшем при изучении следующих дисциплин:

1. Методы и системы обработки больших данных,
2. Машинное обучение на больших данных,
3. Научно-исследовательский семинар "Распределённые системы",
4. при выполнении курсовых и выпускных квалификационных работ.
5. **СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

**Тема 1. Concurrency и многопоточное программирование**

Понятие одновременности (concurrency). Области применения и проблематика. Способы реализации одновременных систем, процессы и потоки, программный инструментарий. Основы многопоточного программирования на языке C++. Типичные ошибки многопоточного программирования. Состояние гонки (race condition). Безопасность (safety). Взаимное исключение. Взаимная блокировка (dealock). Живучесть (liveness). Модель памяти и низкоуровневые примитивы синхронизации. Условная синхронизация. Реализация схемы “Producer-Consumer”. Потокобезопасные структуры данных. Асинхронные вычисления. Конструкции future и promise. Пул потоков. Преимущества и недостатки явных блокировок (locks). Альтернативные подходы к реализации одновременных программ.

**Тема 2. Основы параллельных вычислений**

Связь между concurrency и параллелизмом. Области применения и проблематика параллельных вычислений. Современные параллельные вычислительные системы. Классификация Флинна. Системы с общей разделяемой памятью. Симметричные мультипроцессоры (SMP). Неоднородный доступ к памяти (NUMA). Системы с распределенной памятью. Массивно-параллельные системы (MPP). Вычислительные кластеры. Распределенные вычислительные системы. Гибридные системы. Ускорители и сопроцессоры.

Теоретические основы параллельных вычислений. Анализ внутренней структуры алгоритма и выявление параллелизма. Показатели качества параллельного алгоритма. Законы Амдала и Густафсона-Барсиса. Принципы разработки параллельных алгоритмов. Методология PCAM. Декомпозиция на подзадачи. Типовые структуры параллельных алгоритмов (параллелизм по заданиям, разделяй и властвуй, геометрическая декомпозиция, рекурсивные данные, конвейерная обработка, координация на основе событий). Учёт взаимодействий между подзадачами. Масштабирование подзадач. Распределение подзадач между процессорами. Современные технологии параллельного программирования. Типовые модели параллельного программирования и шаблоны.

**Тема 3. Параллельное программирование на системах с общей памятью**

Основы параллельного программирования на системах с общей памятью на примере технологии OpenMP. Развитие стандарта OpenMP, существующие реализации. Модель выполнения “ветвление-слияние”. Основные директивы и функции OpenMP для языка C++. Атрибуты видимости данных. Ложное разделение данных. Редукция. Оптимизация циклов. Распределение итераций цикла между потоками. Рекурсивный параллелизм, директива task. Привязка потоков к ядрам процессора. Особенности стандарта OpenMP 4.0. Отладка и оптимизация OpenMP-программ.

**Тема 4. Параллельные вычисления на графических процессорах**

Введение в параллельное программирование на графических процессорах (GPU). Особенности архитектуры GPU, отличия от CPU, область применения. Основные этапы проведения вычислений на GPU. Модель вычислений, понятие kernel, выполнение kernel на GPU. Примеры технологий для параллельных вычислений на GPU. Знакомство с технологией CUDA C. Модель программирования, понятия грида и блока потоков, выбор оптимальных параметров запуска. Иерархия памяти GPU, оптимизация работы с памятью. Расхождение потоков, способы устранения. Примеры шаблонов вычислений на GPU: гистограмма, свёртка, редукция, префиксная сумма.

**Тема 5. Параллельное программирование на системах с распределенной памятью**

Основы параллельного программирования на системах с распределенной памятью на примере технологии MPI. Развитие стандарта MPI, существующие реализации. Модель программирования SPMD. Коммуникатор. Основные функции MPI для языка C и их аналоги для языка C++ из библиотеки Boost.MPI. Запуск MPI-программ на кластере. Передача сообщений между парами процессов (точка-точка). Блокирующие, неблокирующие и совмещенные функции обмена сообщениями. Использование неблокирующих функций для перекрытия вычислений и обмена данными. Коллективные взаимодействия процессов (обмен данными, коллективные вычисления, синхронизация). Редукция, встроенные и пользовательские операции. Виртуальные топологии. Пересылка разнотипных данных, пользовательские типы данных и упаковка данных. Особенности стандартов MPI-2 и MPI-3. Отладка и оптимизация MPI-программ.

**Тема 6. Распределенные системы и технологии распределенных вычислений**

Области применения, характерные особенности и виды распределенных систем. Проблемы построения распределенных систем. Архитектурные элементы распределенных систем. Парадигмы взаимодействия распределенных процессов. Клиент-серверные и peer-to-peer системы. Централизованные и децентрализованные системы. Отображение на физические узлы, шардинг, репликация, кэширование, мобильный код. Технологии распределенных вычислений, гриды, добровольные вычисления, облачные вычислительные системы.

Сетевые протоколы UDP и TCP. Реализация распределенных приложений с использованием сетевых сокетов. Технологии распределенного программирования, промежуточное программное обеспечение (middleware). Удалённый вызов процедур, распределенные объекты, веб-сервисы, рассылка сообщений. Реализация распределенных приложений на языке Python с использованием библиотеки Pyro.

Теоретические основы распределенных вычислений, типовые задачи и примеры распределенных алгоритмов. Порядок событий, логические и векторные часы. Построение согласованного глобального состояния (snapshot). Взаимное исключение. Выборы. Консенсус и родственные задачи. Обнаружение отказов, обеспечение отказоустойчивости. Репликация данных и согласованность. Теорема CAP. Практические реализации, система Apache Zookeeper, алгоритм Raft.

**Тема 7. Распределенная обработка больших данных**

Феномен Big Data, проблемы хранения и обработки больших объемов данных. Модель программирования MapReduce. Функции map, reduce, partition, combine и compare. Область применения и примеры задач. Принципы распределенной реализации MapReduce на кластерных системах. Платформа Apache Hadoop, архитектура Hadoop-кластера, основные компоненты платформы. Приёмы и стратегии, используемые при реализации MapReduce-программ. Ограничения модели MapReduce, расширения и альтернативные подходы. Система Apache Spark, модель программирования, сравнение с MapReduce.

**Тема 8. Облачные вычисления**

Концепция облачных вычислений. Характерные черты облачных систем. Модели и примеры облачных сервисов. Примеры использования облачных инфраструктур и сервисов. Работа с облачной инфраструктурой.

1. **ОЦЕНИВАНИЕ**

Результирующая оценка за дисциплину рассчитывается следующим образом:

Орезульт = 0,8·Онакопл + 0,2·Оэкзамен

Накопленная оценка Онакопл формируется на основе результатов текущего контроля (баллов за домашние задания, дополнительных баллов за работу на семинарах). Формула расчёта накопленной оценки объявляется студентам в начале курса.

Способ округления результирующей оценки - арифметический (до ближайшего целого числа).

1. **ПРИМЕРЫ ОЦЕНОЧНЫХ СРЕДСТВ**

# **Примеры задач для домашних заданий**

 1. Задача об обедающих философах.

 2. Параллельная реализация метода k-средних на OpenMP.

 3. Реализация умножения матриц на GPU.

 4. Численное решение уравнения теплопроводности с использованием MPI.

 5. Реализация отказоустойчивой распределенной системы обработки заданий.

**Примерный перечень вопросов к экзамену по дисциплине**

1. Понятие одновременности (concurrency). Области применения и проблематика. Способы реализации одновременных систем, процессы и потоки, программный инструментарий. Основные примитивы многопоточного программирования на C++.

2. Типичные ошибки многопоточного программирования и способы их устранения. Состояние гонки (race condition). Безопасность (safety). Взаимное исключение. Взаимная блокировка (deadlock), необходимые условия возникновения. Живучесть (liveness).

3. Условная синхронизация, std::condition\_variable. Потокобезопасная реализация схемы “Producer-Consumer”. Асинхронные вычисления, std::async. Примитивы future и promise.

4. Атомарные переменные (std::atomic). Атомарные read-modify-write инструкции процессоров. Низкоуровневые эффекты и примитивы синхронизации. Операции с неявными барьерами доступа к памяти.

5. Преимущества и недостатки явных блокировок (locks). Альтернативные подходы к реализации одновременных программ. Акторы (actors). Программная транзакционная память (STM). Программирование потоков данных (dataflow).

6. Параллельные вычислительные системы. Instruction-level parallelism (ILP). Thread-level parallelism (TLP). Классификация Флинна. Системы с общей памятью. Системы с распределенной памятью. Гибридные системы.

7. Теоретические основы параллельных вычислений. Анализ внутренней структуры алгоритма и выявление параллелизма. Модель параллельного алгоритма. Показатели качества параллельного алгоритма. Законы Амдала и Густафсона-Барсиса.

8. Современные технологии параллельного программирования, разновидности и примеры. Типовые модели программирования и шаблоны. Loop Parallelism. Single Program Multiple Data. Partitioned Global Address Space. Master/Worker. Рекурсивный Fork/Join.

9. Типовые структуры параллельных алгоритмов. Параллелизм по заданиям, разделяй и властвуй, геометрическая декомпозиция, рекурсивные данные, конвейерная обработка, координация на основе событий.

10. Принципы разработки параллельных алгоритмов. Методология PCAM. Декомпозиция на подзадачи. Учет взаимодействий между подзадачами. Масштабирование подзадач. Распределение подзадач между процессорами.

11. Технология OpenMP. Область применения, история развития, реализации. Модель выполнения “ветвление-слияние”. Основные директивы и функции OpenMP для языка

C++. Опции видимости данных. Ложное разделение данных (false sharing).

12. Распределение итераций цикла между потоками в OpenMP, используемые стратегии планирования. Директивы sections и single. Поддержка рекурсивного параллелизма в OpenMP, директивы task и taskwait. Привязка потоков к ядрам процессора.

13. Параллельные вычисления на графических процессорах (GPU). Особенности архитектуры GPU, отличия от CPU. Область применения, идеальные и плохие задачи для вычислений на GPU. Понятие гетерогенных вычислений. Основные этапы проведения вычислений на GPU. Модель вычислений, понятие kernel, выполнение kernel на GPU. Разновидности и примеры технологий для параллельных вычислений на GPU.

14. Технология CUDA C. Модель программирования, определение параллельной функции (kernel). Модель выполнения программы, организация вычислений на GPU, понятия грида и блока потоков, выбор оптимальных параметров запуска. Иерархия памяти GPU, глобальная память, регистры, локальная память, общая (shared) память. Оптимизация работы с памятью. Расхождение потоков, способы устранения.

15. Шаблоны вычислений на GPU. Гистограмма. Свёртка (convolution). Редукция. Префиксная сумма (scan). Параллельные алгоритмы и особенности реализации на GPU.

16. Технология MPI. Область применения, история развития, реализации. Модель программирования. Коммуникатор. Структура MPI-программы. Базовые функции MPI для C и их аналоги для C++ из библиотеки Boost.MPI. Типы коммуникационных операций в MPI, примеры функций. Блокирующие и неблокирующие функции.

17. Передача сообщений между парами процессов в MPI. Структура сообщения. Блокирующие, неблокирующие и совмещённые функции обмена сообщениями. Предотвращение взаимной блокировки при обмене сообщениями. Использование неблокирующих функций для перекрытия вычислений и обменов данными.

18. Типы коммуникационных операций в MPI. Коллективные операции. Синхронизация процессов. Обмен данными между процессами. Коллективные вычисления. Альтернативная классификация коллективных операций. Особенности стандартов MPI-2 и MPI-3.

19. Области применения, характерные особенности и виды распределенных систем. Проблемы построения распределенных систем. Парадигмы взаимодействия распределенных процессов. Клиент-серверные и peer-to-peer системы. Централизованные и децентрализованные системы. Отображение на физические узлы, шардинг, репликация, кэширование, мобильный код.

20. Сетевые протоколы UDP и TCP. Реализация распределенных приложений с использованием сетевых сокетов. Технологии распределенного программирования, промежуточное программное обеспечение (middleware). Удалённый вызов процедур, распределенные объекты, веб-сервисы, рассылка сообщений.

21. Отказы в распределенных системах, разновидности, обнаружение и обработка. Способы обеспечения отказоустойчивости. Репликация данных, возможные стратегии. Синхронная и асинхронная репликация. Репликация и согласованность данных, возможные проблемы и гарантии согласованности. Разрешение конфликтов при активной репликации. Размеры кворумов при активной репликации без мастеров.

22. Феномен Big Data, проблемы хранения и обработки больших объёмов данных. Модель программирования MapReduce. Функции map, reduce, partition, combine и compare. Область применения и примеры задач. Принципы параллельной реализации вычислений. Существующие реализации MapReduce.

23. Принципы распределенной реализации MapReduce на кластерных системах. Архитектура Google MapReduce, оптимизации выполнения программ, обработка отказов. Платформа Apache Hadoop, архитектура Hadoop-кластера, основные компоненты платформы.

24. Приёмы и стратегии, используемые при реализации MapReduce-программ. Локальная агрегация промежуточных данных. Стратегии Pairs и Stripes. Приёмы order inversion и value-to-key conversion. Преимущества и недостатки MapReduce. Альтернативные модели вычислений. Система Apache Spark, модель программирования, сравнение с MapReduce.

25. Концепция облачных вычислений. Варианты использования облаков, примеры облачных инфраструктур и сервисов. Отличительные характеристики облаков, модели обслуживания, развёртывания и оплаты. Хранение данных в облаке. Проведение вычислений в облаке. Преимущества и недостатки облаков.

1. **РЕСУРСЫ**
	1. **Основная литература**
	2. Peter Pacheco. An Introduction to Parallel Programming. Morgan Kaufmann, 2011.
	3. Foster I. Designing and Building Parallel Programs: Concepts and Tools for Software Engineering. Reading, MA: Addison-Wesley, 1995. <http://bit.ly/2vfOkO6>
	4. David B. Kirk, Wen-mei W. Hwu. Programming Massively Parallel Processors: A Hands-on Approach. Morgan Kaufmann, 2013 (2nd edition).
	5. Martin Kleppmann. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. O'Reilly Media, 2017.
	6. Jimmy Lin, Chris Dyer. Data-Intensive Text Processing with MapReduce, 2010. http://lintool.github.io/MapReduceAlgorithms/
	7. **Дополнительная литература**
2. Anthony Williams. C++ Concurrency in Action: Practical Multithreading. Manning, 2012. (Перевод на русский: Энтони Уильямс. Параллельное программирование на C++ в действии: Практика разработки многопоточных программ. ДМК Пресс, 2012.)
3. G. Barlas. Multicore and GPU Programming: An Integrated Approach. Morgan Kaufmann, 2014.
4. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2004 г.
5. Гергель В.П. Теория и практика параллельных вычислений: учебное пособие. - М.: ИНТУИТ; БИНОМ. Лаборатория знаний, 2007.
6. Michael McCool, James Reinders, Arch Robison. Structured Parallel Programming: Patterns for Efficient Computation. Morgan Kaufmann, 2012.
7. А.В. Боресков и др. Параллельные вычисления на GPU. Архитектура и программная модель CUDА: Учеб. пособие. - М.: Издательство Московского университета, 2012.
8. J. Cheng, M. Grossman, T. McKercher. Professional CUDA C Programming. Wrox, 2014.
9. W. Gropp, E. Lusk, A. Skjellum. Using MPI: Portable Parallel Programming with the Message-Passing Interface. MIT Press, 2014 (3rd edition)
10. George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair. Distributed Systems: Concepts and Design. Addison Wesley, 2011 (5th edition). <http://www.cdk5.net/>
11. Andrew S. Tanenbaum, Maarten van Steen. Distributed Systems: Principles and Paradigms. Pearson, 2007 (2nd edition). (Перевод на русский 1-го издания: Э. Таненбаум, М. ван Стеен. Распределенные системы. Принципы и парадигмы. - СПб.: Питер, 2003.)
12. Tom White. Hadoop: The Definitive Guide: Storage and Analysis at Internet Scale. O'Reilly Media, 2012 (3rd edition). (Перевод на русский: Том Уайт. Hadoop. Подробное руководство. - СПб.: Питер, 2013.) – 3rd edition, 2012
13. Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia. Learning Spark: Lightning-Fast Big Data Analysis, 2015.
	1. **Программное обеспечение**

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование** | **Условия доступа** |
| 1 | Компилятор GCC | Свободно распространяемое ПО |
| 2 | Интерпретатор языка Python | Свободно распространяемое ПО |
| 3 | NVIDIA CUDA Toolkit | Свободно распространяемое ПО |
| 4 | OpenMPI | Свободно распространяемое ПО |

* 1. **Профессиональные базы данных, информационные справочные системы,
	интернет-ресурсы (электронные образовательные ресурсы)**

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование** | **Условия доступа** |
|  | ***Профессиональные базы данных, информационно-справочные системы*** |
| 1. | Электронно-библиотечная система Юрайт  | URL: https://biblio-online.ru/ |
|  | ***Интернет-ресурсы (электронные образовательные ресурсы)*** |
| 1. | Открытое образование  | URL: https://openedu.ru/ |

* 1. **Материально-техническое обеспечение дисциплины**

Учебные аудитории для лекционных занятий по дисциплине обеспечивают использование и демонстрацию тематических иллюстраций, соответствующих программе дисциплины в составе:

ПЭВМ с доступом в Интернет (операционная система, офисные программы, антивирусные программы);

мультимедийный проектор с дистанционным управлением.

Учебные аудитории для практических занятий по дисциплине оснащены рабочими станциями для каждого студента, а также проектором, экраном и компьютером для проведения презентаций, с возможностью подключения к сети Интернет и доступом к электронной информационно-образовательной среде НИУ ВШЭ.