Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Артеменков Михаил Николаевич
Должность: Ректор
Дата подписания: 09.11.2026 18:37:31
Уникальный программный ключ:
a63b74166e8f4fc17bd1e0462725f79e393b6f69

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
_____Устименко Ю.А.
«7» сентября 2021 г.

Рабочая программа дисциплины
ФТД 01. Web-сервисы для параллельной обработки данных

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

Форма обучения: очная
Курс – 1
Семестр – 1
Всего зачетных единиц –3,часов – 108

Форма отчетности: зачет- 1 семестр

Программу разработал:
кандидат физико-математических наук Винокурова А.С.

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

Смоленск
2021

1. Место дисциплины в структуре ОП

Дисциплина ФТД 01. Web-сервисы для параллельной обработки данных относится к факультативным дисциплинам. Изучается на первом курсе в первом семестре. Содержание способствуют развитию компетенций, необходимых для изучения следующих дисциплин: «Взаимодействие в распределенных программных системах», «Параллельное программирование прикладных задач», «Разработка Web-сервисов для мобильных приложений», «Прикладные Интернет-технологии», «Системы бизнес-аналитики» / «Разработка аналитических систем».

2. Планируемые результаты обучения по дисциплине

Компетенция	Индикаторы достижения
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

Введение. Создание потоков. Синхронизация. Атомарные снимки регистров. Введение в высокопроизводительные вычисления.Тенденции развития вычислительных систем, обуславливающие необходимость применения распределенных (параллельных) методов вычислений. Примеры вычислительно емких задач из разных областей науки. Классификация параллельных систем (SIMD, MISD..., SMP, MPP). Современные высокопроизводительные системы: начиная от расширений SSE, чере многоядерность к узлам кластеров.Понятия ускорения, эффективности (закон Амдала). Многопоточность или IPC. Создание/завершение потоков. Механизм запуска потока. Корректное завершение потоков: cancellation points, interrupted exception, примеры кода в glibc. Сравнение различных потоков (POSIX, boost, java). Проброс исключений между потоками.Примитивы синхронизации. Необходимость синхронизации: простые гонки данных. Реализация примитивов синхронизации: алгоритм булочника. Виды мьютексов: рекурсивные/нерекурсивные, read/write, spin, futex. Корректные захват/освобождение примитивов. CAS-операции и атомики. Условные переменные: использование wait/notify, Spurious wakeups. Алгоритмы синхронизации. Грубая. Тонкая. Оптимистичная. Ленивая. Неблокирующая (параллель с ORM). Атомарные снимки регистров. Классификация алгоритмов: lock-free, wait-free. SWMR-регистры. Lock-free snapshot. Wait-free snapshot.

Характерные ошибки, профилирование, пулы потоков и шаблоны. Ошибки параллельного программирования. Основные ошибки многопоточного программирования: Гонки данных (Data Race), Взаимная блокировка (Deadlock). Специфические ошибки: Реакция потока на сигнал, Блокировки при fork многопоточных программ, Проблема ABA, Инверсия приоритетов. Профилирование многопоточных приложений. Средства анализа производительности: Утилита time, Intel Parallel Studio, Valgrind (модули callgrind, cachegrind). Пример поиска узких мест. Java.util.concurrent и пулы потоков Пулы потоков, корректное завершение пула. Контроль задач через Future. Потокобезопасные контейнеры. OpenMP и Intel TBB. ОбзорOpenMP: параллельные секции, области видимости переменных, ограничения. Обзор Intel TBB: алгоритмы, аллокаторы, деревья задач, особенности планирования (work stealing...), flow graphs (параллель с BPEL).

Шаблоны параллельного программирования. Структурные шаблоны: Декомпозиция по задачам, Геометрическая декомпозиция, Recursive Data, Pipeline. Некоторыепрограммныеструктуры: Parallelloops, Boss/Worker. Разное: Doublecheck, LocalSerializer.

Кластеры, сети Петри. Модели памяти, асинхронный ввод/вывод. Кластерные вычисления. Виды кластерных систем: Балансировки нагрузки, Высокой надёжности, Вычислительные. История и назначение стандарта MPI. Обмен сообщениями: С блокировкой, Без блокировки, Отложенные запросы на взаимодействие. Взаимодействие процессов: Группы и коммуникаторы, Операции коллективного взаимодействия процессов, Редукция, Виртуальные топологии. Консенсус. Сети Петри. Консенсус: Консенсусное число RMW-регистров, Универсальность CAS-операций. Верификация || программ (сетиПетри).Транзакционнаяпамять.Идеятransactionalmemory: Softwaretransactionalmemory, Hardwaretransactionalmemory. Преимущества и круг задач. Реализация HTM на линейках кэша. Lock teleportation. Асинхронный ввод/вывод. Блокирующий/неблокирующий. Синхронный (реактор)/асинхронный (проактор). Преимущества асинхронной работы и реализация со стороны операционной системы. Библиотеки асинхронного ввода/вывода. Модель памяти Устройство кэшей процессора. Пример на протоколе MESI. Барьерыпамяти (store/load). Моделипамяти: Sequential consistency.... Acquire/release семантика.

Линеаризуемость, lock-free структуры данных. Линеаризуемость. Понятие линеаризуемости. Lock-free стек Trieber. Пример на очередях. Lock-free очередь Michael & Scott. Точки линеаризации. Flat-Combining. Схема Flat-Combining. Возможные оптимизации за счёт интерференции операций. Сравнение производительности с lock-free очередью Michael & Scott. Древовидные Lock-free структуры данных. User-space RCU. MultiArray. Схемыуправленияпамятью: Taggedpointers, Hazardpointer. OpenCL.АрхитектуравычисленийнаGPU. ПринципыработысостандартомOpenCL.

4. Тематический план

№ п/п	Разделы и темы	Всего часов	Формы занятий	
			лабораторные занятия	самостоятельная работа
1	Введение. Создание потоков. Синхронизация. Атомарные снимки регистров	27	8	19
2	Характерные ошибки, профилирование, пулы потоков и шаблоны	27	8	19
3	Кластеры, сети Петри. Модели памяти, асинхронный ввод/вывод	27	8	19
4	Линеаризуемость, lock-free структуры данных	27	8	19
	Итого	108	32	76

5. Виды образовательной деятельности

Занятия семинарского типа – лабораторные работы

Лабораторные работы№1-4. Форма представления обучающимися - программа на выбранном языке программирования.

Реализовать классический паттерн producer-consumerc небольшими дополнительными условиями. Программа должна состоять из четырех потоков: один - главный, producer, consumeri interruptor. На стандартный ввод программе подается строка - список чисел произвольной длины.

Задача producer-потока - получить на вход список чисел, и по очереди использовать каждое значение из этого списка для обновления переменной разделяемой между потоками. После этого поток должен дождаться реакции consumer-потока, и продолжить обновление значений только после того как поток-consumerпринял это изменение.

Задача consumer-потока отреагировать на каждое изменение переменной datai набирать сумму полученных значений. После того как достигнуто последнее обновление, функция потока должна вернуть результирующую сумму. Также этот поток должен защититься от попыток потока-interruptorего остановить.

Задача потока-interruptorпроста: пока происходит процесс обновления значений, он должен постоянно пытаться остановить поток consumer. Как только поток producerпроизвел последнее обновление, этот поток завершается.

Функция run_threadsдолжна запускать все три потока, дожидаться их выполнения, и возвращать результат потока-consumer.

Для обеспечения межпоточного взаимодействия допускается использование только pthreadAPI.

Лабораторные работы №5-8. Форма представления обучающимися - программа на выбранном языке программирования.

Реализовать интерфейс множества в виде lock-freeструктуры данных. Задание должно состоять из одного файла LockFreeSetImpl.javaсодержащего класс, реализующий интерфейс LockFreeSet, приложенный к задаче. Спецификация реализация описана в документации к интерфейсу.

Лабораторные работы№9-12:

Реализовать с использованием графовой модели вычислений следующую задачу, разбив её на чати и обеспечив максимальную потенциальную утилизацию ресурсов (изображение == матрица МхК):

1.На вход подаются генерированные случайным образом изображения

2.Различные алгоритмы ищут || на изображении:

- Набор максимальных элементов (максимальная яркость)
- Набор минимальных элементов (минимальная яркость)
- Набор элементов, равных заданному в командной строке значению (0-255)

3. По результатам предыдущего шага на избражении выделяются все найденные элементы (например, квадрат с максимальной яркостью вокруг точки)

4. Результат предудущего шага передаётся на:

- Расчёт обратного изображения (инверсия яркости)
- Расчёт средней яркости изображения

5. Результат расчёта средней яркости выводится в файл в произвольном формате, если указан соответствующий флаг запуска приолжения

6. Результат расчёта обратного изображения никуда не идёт.

Лабораторные работы №№13-16:

Необходимо рассчитать двухмерную свёртку 2 заданных матриц по заданной формуле.

Самостоятельная работа

Самостоятельная работа - планируемая учебная работа студентов, выполняемая во внеаудиторное время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Самостоятельная работа, не предусмотренная программой учебной дисциплины, раскрывающей и конкретизирующей ее содержание, осуществляется студентом инициативно, с целью реализации собственных учебных и научных интересов.

Для более эффективного выполнения самостоятельной работы по дисциплине преподаватель рекомендует источники для работы, характеризует наиболее рациональную методику самостоятельной работы, демонстрирует ранее выполненные студентами работы и т.п.

Виды заданий для внеаудиторной самостоятельной работы, их содержание и характер могут иметь вариативный и дифференцированный характер, учитывать индивидуальные особенности студента.

Самостоятельная работа может осуществляться индивидуально или группами студентов onlineи на занятиях в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности.

Контроль результатов внеаудиторной самостоятельной работы осуществляется в пределах времени, отведенного на обязательные учебные занятия по дисциплине на практических занятий.

6. Критерии оценивания результатов освоения дисциплины (модуля)

6.1. Оценочные средства и критерии оценивания для текущей аттестации

1. Критерии оценивания лабораторных работ.

Оценка	Критерии выставления оценки
«Отлично» (5)	Нагрузочное тестирование и проверка автоматизированными средствами показала отсутствие ошибок. Проверка исходного кода ошибок не выявила
«Хорошо» (4)	Нагрузочное тестирование и проверка автоматизированными средствами показала отсутствие ошибок, однако при проверке исходного кода найдены недочеты
«Удовлетворительно» (3)	Нагрузочное тестирование и проверка автоматизированными средствами показали не более двух отрицательных тестов
«Неудовлетворительно» (2)	В ходе тестирования выявлены ошибки в реализации, отрицательный результат у более двух тестов

6.2. Оценочные средства и критерии оценивания для промежуточной аттестации

Примерный перечень вопросов к зачету:

- Примитивы синхронизации
- Алгоритмы синхронизации
- Атомарные снимки регистров
- Ошибки параллельного программирования
- Пулы потоков
- Шаблоны || программирования
- Консенсус
- Транзакционная память
- Асинхронный ввод/вывод
- Модель памяти
- Линеаризуемость
- Flat-Combining

Критерии оценивания и шкала оценки зачета

Оценка	Критерии выставления оценки
«Зачтено» (5)	Выполнены все лабораторные работы. Дан развернутый ответ на поставленные вопросы. Материал изложен последовательно. Имеются логичные и аргументированные выводы.
«Зачтено» (4)	Выполнено более 75% лабораторных работ. Дан развернутый ответ на поставленные вопросы. Материал изложен в целом последовательно. Имеются логичные и аргументированные выводы.
«Зачтено» (3)	Выполнено более 50% лабораторных работ. Ответ на вопросы не является полным. Материал изложен непоследовательно. Выводы не

	аргументированы.
«Не зачтено» (2)	Выполнено менее 50% работ. Ответ на вопросы является неверным. Материал изложен непоследовательно. Отсутствуют выводы.

7. Перечень основной и дополнительной учебной литературы

7.1. Основная литература

1. Малявко А. А. Параллельное программирование на основе технологий openmp, cuda, opencl, mpi: учебное пособие для вузов / А. А. Малявко. – Москва: Издательство Юрайт, 2021. – 135 с. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL:https://urait.ru/bcode/467800 .
2. Бабичев С. Л. Распределенные системы: учебное пособие для вузов / С. Л. Бабичев, К. А. Коньков. – Москва: Издательство Юрайт, 2020. – 507 с. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: https://urait.ru/bcode/457005 .

7.2. Дополнительная литература

1. Богачёв К.Ю. Основы параллельного программирования. - М.: БИНОМ. Лаборатория знаний, 2003. - 342 с.
2. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. - СПб.: БХВ-Петербург, 2002. - 608 с.
3. Воеводин Вл.В., Жуматий С.А. Вычислительное дело и кластерные системы. - М.: Изд-во МГУ, 2007. - 150 с.
4. Камерон Х., Трейси Х. Параллельное и распределенное программирование с использованием C++. : Пер. с англ. - М.: Изд-во "Вильямс", 2004. - 672 с.
5. Корнеев В.Д. Параллельное программирование в MPI. - Москва-Ижевск: Институт компьютерных исследований, 2003. - 304 с.
6. Лацис А. Как построить и использовать суперкомпьютер. - М.: Бестселлер, 2003. – 240 с.
7. Лупин С.А., Посыпкин М.А. Технологии параллельного программирования. Серия: Высшее образование. - М.: Форум, Инфра-М, 2008. - 208 с.
8. Малышкин В.Э. Введение в параллельное программирование мультикомпьютеров. - М.: Новосибирск, 2003. - 268 с.
9. Миллер Р., Боксер Л. Последовательные и параллельные алгоритмы: Общий подход. - М.: БИНОМ. Лаборатория знаний, 2006. - 406 с.
10. Немнюгин С.А., Стесик О.Л. Параллельное программирование для многопроцессорных вычислительных систем - СПб.: БХВ-Петербург, 2002. - 400 с.
11. Эндрюс Г.Р. Основы многопоточного, параллельного и распределенного программирования: Пер. с англ. - М.: Изд-во "Вильямс", 2003. - 512 с.

7.3.Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Электронная энциклопедия Википедия (wikipedia.org).
2. citforum.ru/database - сайт Центра информационных технологий

8. Материально-техническое обеспечение

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

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

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

Microsoft Visual Studio 2019-22 Community.

Microsoft SQL Server Express.