



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего
образования
«Дальневосточный федеральный университет»
(ДВФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)

«СОГЛАСОВАНО»

Руководитель ОП

Артемяева И.Л.

«УТВЕРЖДАЮ»

И.о. директора департамента

Смагин С.В.

«15» июля 2021 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Параллельное программирование

Направление подготовки 09.03.04 Программная инженерия
(Программная инженерия)
Форма подготовки очная

курс 4 семестр 7
лекции 16 час.
практические занятия 00 час.
лабораторные работы 16 час.
в том числе с использованием МАО лек. 0 / пр. 0 / лаб. 16 час.
всего часов аудиторной нагрузки 32 час.
в том числе с использованием МАО 16 час.
самостоятельная работа 76 час.
в том числе на подготовку к экзамену 45 час.
контрольные работы (количество) не предусмотрены
курсовая работа / курсовой проект не предусмотрены
зачет _ семестр
экзамен 7 семестр

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта по направлению подготовки 09.03.04 Программная инженерия утвержденного приказом Министерства образования и науки РФ от 19 сентября 2017 г. № 920 (с изменениями и дополнениями)

Рабочая программа обсуждена на заседании кафедры прикладной математики, механики, управления и программного обеспечения, протокол № 12.1 от «25» декабря 2019 г.

И.о. директора департамента программной инженерии и искусственного интеллекта Смагин С.В.

Составитель: Чусов А.А., доцент кафедры прикладной математики, механики, управления и программного обеспечения, к.т.н.

Владивосток
2021

Оборотная сторона титульного листа РПД

I. Рабочая программа пересмотрена на заседании кафедры прикладной математики, механики, управления и программного обеспечения:

Протокол от «09» июля 2021 г. № 7.1

Заведующий кафедрой _____ Артемяева И.Л.
(подпись) (И.О. Фамилия)

II. Рабочая программа пересмотрена на заседании департамента программной инженерии и искусственного интеллекта:

Протокол от «17» сентября 2021 г. № 9.1

И.о. директора департамента _____ Смагин С.В.
(подпись) (И.О. Фамилия)

III. Рабочая программа пересмотрена на заседании департамента:

Протокол от «_____» _____ 20__ г. № _____

Директор департамента _____
(подпись) (И.О. Фамилия)

IV. Рабочая программа пересмотрена на заседании департамента:

Протокол от «_____» _____ 20__ г. № _____

Директор департамента _____
(подпись) (И.О. Фамилия)

ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель: раскрыть смысл ключевых понятий из области параллельных вычислений, сформировать представление о современных параллельных вычислительных архитектурах, моделях, методах и технологиях их программирования, привить навыки работы с параллельными вычислительными платформами.

Задачи:

- приобретение студентами базового набора знаний в областях параллельной алгоритмизации и параллельных вычислений;
- первичных навыков работы с современными параллельными вычислительными системами и инструментальными средствами разработки параллельного программного обеспечения.

Планируемые результаты обучения по данной дисциплине (знания, умения, владения), соотнесенные с планируемыми результатами освоения образовательной программы.

Общепрофессиональные компетенции выпускников и индикаторы их достижения:

Наименование категории (группы) общепрофессиональных компетенций	Код и наименование общепрофессиональной компетенции	Код и наименование индикатора достижения общепрофессиональной компетенции
	ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.1. Знает основные направления развития вычислительных машин, комплексов и сетей, критерии применимости традиционных и нетрадиционных параллельных вычислительных архитектур для решения различных проблем предметных областей ОПК-2.2. Умеет создавать и применять вычислительные системы, как универсального, так и специализированного назначения, для решения вычислительных задач предметной области. ОПК-2.3. Владеет методами алгоритмического представления вычислительных проблем в формах, реализуемых на

		параллельных вычислителях традиционных и нетрадиционных архитектур.
	ОПК-7. Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой	<p>ОПК-7.1. Знает концептуальные основы параллельных вычислений, области применимости основных параллельных алгоритмов, оценки эффективности параллельных программ, критерии параллелизуемости последовательных алгоритмов, модели параллельных вычислений.</p> <p>ОПК-7.2. Умеет выполнять моделирование, синтез, анализ, параллельных алгоритмов, оценку их эффективности с использованием фундаментальных понятий информатики и дискретной математики.</p> <p>ОПК-7.3. Имеет навыки синтеза и анализа параллельных алгоритмов решения типовых вычислительных задач с заданными функциональными требованиями к эффективности.</p>

Профессиональные компетенции выпускников и индикаторы их достижения:

Задача профессиональной деятельности	Объект или область знания	Код и наименование профессиональной компетенции	Код и наименование индикатора достижения профессиональной компетенции	Основание (ПС, анализ иных требований, предъявляемых к выпускникам)
Тип задач профессиональной деятельности: научно-исследовательский				
Участие в научно-исследовательских и опытно-конструкторских работах в области программной инженерии. Анализ и выбор	Прикладные и информационные процессы. Информационные технологии и. Программное обеспечен	ПК-4. Готовность к использованию методов и инструментальных средств исследования объектов профессиональной деятельности	ПК-4.1. Знает современные инструментальные средства проектирования, анализа, реализации, отладки и тестирования параллельных программ для параллельных	06.028 Системный программист 06.022 Системный аналитик

<p>программно-технологических платформ, сервисов и информационных ресурсов программной инженерии; подготовка обзоров, аннотаций, составление рефератов и докладов, публикаций и библиографии и по научно-исследовательской работе в области программной инженерии</p>	<p>ие</p>		<p>вычислителей основных современных архитектур. ПК-4.2. Умеет применять современные универсальные и специализированные инструментальные средства моделирования, анализа и разработки программ, формулировать и оценивать эффективность различных параллельных реализаций. ПК-4.3. Владеет навыками работы с основными инструментальными средствами проектирования, реализации и разработки программного обеспечения для основных параллельных вычислительных архитектур.</p>	
<p>Тип задач профессиональной деятельности: проектный</p>				
<p>Формирование требований к информатизации и автоматизации и прикладных процессов, формализация предметной области проекта; технико-экономическое</p>	<p>Прикладные и информационные процессы. Информационные технологии. Программное обеспечение</p>	<p>ПК-6. Владение навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения</p>	<p>ПК-6.1. Знает методы моделирования и анализа параллельных алгоритмов и программ при решении задач предметной области. ПК-6.2. Умеет создавать и применять модели информационных</p>	<p>06.028 Системный программист 06.004 Специалист по тестированию в области информационных технологий 06.001 Программист</p>

<p>е обоснование проектных решений и составление технического задания на разработку программного продукта; проектирование программно-аппаратных средств в соответствии с техническим заданием; применение современных инструментальных средств при разработке программного обеспечения; документирование компонентов информационной системы на стадии жизненного цикла</p>			<p>технологий и проблем предметных областей для решения вычислительных задач в предметных областях. ПК-6.3. Владеет методами синтеза и анализа моделей информационных технологий, а также применения этих моделей для решения вычислительных задач в предметных областях.</p>	
--	--	--	---	--

Тип задач профессиональной деятельности: производственно-технологический

<p>Проведение работ по инсталляции программного обеспечения автоматизированных систем и загрузки баз данных; настройка параметров ИС и</p>	<p>Программное обеспечение</p>	<p>ПК-9. Владение навыками использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления</p>	<p>ПК-9.1. Знает основные инструменты языков программирования и операционных систем для реализации параллельных вычислений, логические сущности, предоставляемые параллельными</p>	<p>06.028 Системный программист 06.022 Системный аналитик 06.004 Специалист по тестированию в области информационных технологий 06.001 Программист</p>
--	--------------------------------	--	--	--

<p>тестирование результатов настройки; ведение технической документации; техническое сопровождение ИС в процессе эксплуатации ; применение Web технологий при реализации удаленного доступа в системах клиент – сервер и распределенных вычислений</p>		<p>базами данных</p>	<p>платформами для параллельного программирования, критерии их применимости, их вклад в функциональную эффективность. ПК-9.2. Умеет выбирать адекватные решаемой вычислительной задаче параллельные методы и средства языков программирования, операционных систем и платформ; формально обосновывать свой выбор. ПК-9.3. Имеет навыки адекватного применения сущностей интерфейсов реализации параллельных вычислений, на основе функциональных требований к эффективности, а также требований инкапсуляции и связанности интерфейсов параллельной реализации.</p>	
--	--	----------------------	---	--

Для формирования вышеуказанных компетенций в рамках дисциплины «Параллельное программирование» применяются неимитационные методы активного/интерактивного обучения: выполнение проектов с использованием компьютерных технологий и специализированного программного обеспечения.

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Трудоемкость (16 час)

Тема 1. Обоснование и теоретические основы параллельных вычислений, алгоритмов и программ (1 час)

1. Актуальность параллельных вычислений. Закон Мура и гипотеза Минского.

2. Теоретическое обоснование параллельных вычислений. Параллелизуемость и масштабируемость параллельных алгоритмов. Закон Амдала. Параллельные формы алгоритмов.

Тема 2. Архитектуры и интерфейсы платформ для параллельных вычислений (2 часа)

1. Таксономия Флинна.

2. Параллельные системы с общей памятью. Симметричные мультипроцессоры. Кэш-память и когерентность кэша.

3. Векторизация вычислений для SIMD-расширений процессоров.

4. Механизмы автоматической оптимизации выполнения последовательных вычислений центральным процессором и их влияние на параллельную программу; обзор классических и современных языковых средств запрета элементов автоматической оптимизации последовательных вычислений.

5. Системы с разделенной памятью. Архитектуры вычислительных систем с разделенной памятью. Особенности дизайна параллельных вычислений с использованием мультимониторных систем.

6. Использование специализированных процессоров для решения задач, примеры реализации для цифровой обработки одномерных сигналов. Графические процессоры для вычислений общего назначения. Реализация GPGPU графическими процессорами AMD (ATI) и Nvidia. Архитектура и язык Nvidia CUDA. Пример переноса C-программы, реализующей сложение векторов, на графические процессоры Nvidia CUDA, реализация с помощью OpenCL для графических процессоров AMD (ATI).

Тема 3. Обзор языковых средств реализации параллельных вычислений (2 часа)

1. Языковое расширение и библиотека OpenMP для языков C и C++.

2. Средства реализации совместных и параллельных вычислений Windows и Posix.

3. Интерфейс реализации совместных и параллельных вычислений с помощью стандартной библиотеки C11.

4. Потоки C++11 и параллельные алгоритмы C++17.

5. Примеры реализации параллельных алгоритмов поиска минимума и аккумулярования элементов вектора.

Тема 4. Средства управления параллельными вычислениями (4 часа)

1. Интерфейс и реализация взаимодействия единиц параллельного выполнения в системах с общей и разделенной памятью. Управление вычислениями с помощью передачи сообщений в системах с общей и разделенной памятью.

2. Проблемы управления и контроля доступа к разделяемым ресурсам. Состязательность и гонки. Ложное разделение памяти. Синхронизация доступа к разделяемым ресурсам.

3. Примитивы синхронизации параллельного выполнения в системах с общей памятью. Мьютексы. Мониторы: переменные условия, события и семафоры. Барьерная синхронизация.

4. Проектирование и реализация комплексных протоколов синхронизации доступа к разделяемым ресурсам. Реализация событий и семафоров на основе мьютексов и переменных условий. Реализация интерфейса передачи сообщений на основе многопоточной очереди сообщений.

5. Примитивы синхронизации, предоставляемые библиотеками языков C11 и C++11, платформами Windows и Linux (Pthreads). Масштабируемость алгоритмов, использующих синхронизацию. Облегченные мьютексы и спин-счетчики.

6. Атомарный неблокирующий доступ к элементарным ресурсам разделяемой памяти. Методы задания политик переупорядочивания языковых операций и инструкций в C++11. Видимость изменений в памяти. Примеры реализаций алгоритмов вставки/извлечения из очереди без использования мьютексной синхронизации.

7. Параллелизм, основанный на задачах. Интерфейс, балансировка нагрузки, пулы потоков, планирование выполнения. Динамический и статический параллелизм выполнения. Примеры реализации параллелизма, основанного на задачах, (расчет чисел последовательности Фибоначчи) с помощью конструкций OpenMP и C++11. Реализуемость для систем с общей памятью.

8. Особенности организации параллельных вычислений в системах с распределенной памятью посредством передачи сообщений.

9. Топологии вычислительных сетей и коммуникационная сложность.

Тема 5. Логическая модель и программная архитектура графических процессоров Nvidia CUDA (2 часа)

1. Логическая модель вычислителей CUDA. Поточковые мультипроцессоры CUDA. Блоки и потоки CUDA.

2. Арифметико-логические устройства и блоки трансцендентных операций CUDA. Параллелизм данных. Типы данных и параллелизм потокового мультипроцессора CUDA. Ветвления, условные переходы, рекурсия и прерывания на устройствах CUDA.

3. Память CUDA: общая память устройства, разделяемая память и кэш, текстурная память.

4. Компиляция CUDA кода. Взаимодействие компилятора nvcc, LLVM и распространенных компиляторов C++ Windows и Linux.

5. Обзор основных возможностей устройств CUDA различных Compute Capability.

6. Понятие группы потоков CUDA Warp. Оценка занятости мультипроцессоров CUDA при выполнении вычислений и доступе к памяти. Перекрывающийся доступ к разделяемой и общей памяти.

Тема 6. Реализация параллельных вычислений с помощью CUDA (3 часа)

1. CUDA расширение языка C. Понятие ядра (kernel) CUDA. Функции устройства. Поддержка элементов языков C и C++ и ограничения среды выполнения CUDA.

2. Передача данных между хостом и устройством CUDA.

3. Использование разделяемой (shared) памяти для снижения длительности ввода-вывода данных. Влияние адресации и выравнивания на сериализацию доступа к разделяемой памяти: банки разделяемой памяти и конфликты доступа к банкам.

4. Интерфейсы для синхронизации вычислений на CUDA. Атомарные операции CUDA.

5. Снижение затрат на доступ устройством CUDA к памяти хоста. Блокировка страниц памяти хоста. Взаимное отображение памяти устройства и памяти хоста; автоматизированная передача данных между устройством и хостом.

6. Асинхронное и одновременное выполнение ядер (kernel) CUDA. Потоки CUDA Stream. Перекрывающееся выполнение ядер CUDA и операций ввода-вывода. Реализация гетерогенного параллелизма с помощью центрального мультипроцессора и процессоров CUDA.

7. Программное взаимодействие с несколькими устройствами CUDA.

8. Высокоуровневые библиотеки CUDA. Nvidia Thrust. PyCUDA. Поддержка CUDA в среде Matlab.

II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА

Лабораторные работы (16 час.)

Лабораторная работа № 1. Реализация параллельного численного интегрирования потоками выполнения центральных процессоров с помощью OpenMP, потоков C++11 и средств операционной системы (2 час.)

Реализовать последовательный и параллельный алгоритм численного интегрирования функции с записью локальных результатов интегрирования разделяемый массив и суммированием его элементов в основном потоке выполнения. Измерить показатели времени выполнения интегрирования от количества используемых рабочих потоков.

Лабораторная работа № 2. Доступ к разделяемой памяти: кеширование, синхронизация посредством взаимного исключения, состязательность и гонки (2 час.)

Модифицировать алгоритм численного интегрирования функции так, чтобы элементы массива результатов, получаемых рабочими потоками, были выравнены по восьмибайтовой границе. Модифицировать алгоритм так, чтобы частные результаты интегрирования накапливались в разделяемой переменной, защищаемой мьютексом. Оценить масштабируемость обеих реализаций.

Лабораторная работа № 3. Параллелизм выполнения, реализуемый задачами (2 час.)

С помощью мьютекса и монитора реализовать блокирующую очередь задач, агрегирующий ее пул рабочих потоков, предоставляющий интерфейс для добавления задач и ожидания их выполнения, а также класс задач, рекурсивно реализующих вычисление заданного элемента последовательности Фибоначчи. Оценить и объяснить масштабируемость реализации.

Лабораторная работа № 4. Параллельные алгоритмы инициализации массива и вычисления среднего с помощью редукции на CUDA (2 час.)

В рамках одного ядра (kernel) CUDA реализовать алгоритм параллельной инициализации массива случайными числами с помощью линейного конгруэнтного генератора и выбранного источника энтропии, и

затем, выполнив барьерную синхронизацию с помощью `__syncthreads`, осуществить вычисление среднего арифметического сгенерированных элементов через их редукцию.

Лабораторная работа № 5. Использование разделяемой памяти блока CUDA и параллельная фильтрация одномерного сигнала, представленного массивом целых (2 час.)

Представить и реализовать программой для устройства CUDA алгоритм, принимающий на вход вектор числовых значений и выполняющий инициализацию выходного буфера, так чтобы каждый его элемент был средним значением элементов входного вектора, которые принадлежат окну с заданным радиусом, расположенным над соответствующим элементом входного вектора.

Модифицировать реализацию так, чтобы использовалась разделяемая память блока устройства CUDA, выполнив барьерную синхронизацию. Оценить ускорение вычислений для различных радиусов окна.

Лабораторная работа № 6. Параллельное быстрое преобразование Фурье на CUDA (2 час.)

Реализовать БПФ одномерного сигнала, представленного входным вектором целочисленных значений. Для реализации использовать единственное ядро CUDA и синхронизацию с помощью барьера.

Лабораторная работа № 7. Гетерогенный параллелизм, реализуемый потоками центрального процессора и потоками CUDA. Асинхронный ввод-вывод и одновременность выполнения различных реализаций ядер CUDA. (4 час.)

Реализовать параллельную генерацию синусоидального сигнала рабочими потоками центрального процессора параллельно с потоками CUDA. Каждый рабочий поток центрального процессора должен до выполнения собственных вычислений вызывать ядро CUDA с одним блоком и максимальным числом потоков в блоке. Передача данных между устройством и виртуальной памятью и вызов ядра CUDA должны выполняться в собственном потоке (stream) CUDA. Оценить эффект от привязки страниц виртуальной памяти, в которых выделен выходной буфер, к CUDA с помощью `cudaHostRegister`.

III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Параллельное программирование» представлено в разделе VIII и включает в себя:

план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;

характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;

требования к представлению и оформлению результатов самостоятельной работы;

критерии оценки выполнения самостоятельной работы.

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/ п	Контролируемые разделы / темы дисциплины	Коды и этапы формирования компетенций		Оценочные средства	
				текущий контроль	промежуточная аттестация
1	Обоснование параллельных вычислений	ОПК-2 ОПК-7 ПК-4 ПК-6 ПК-9	знает	Устный опрос (УО-1); тесты (ПР-1)	Экзамен, вопросы 1, 5, 11, 15, 21.
			умеет	Устный опрос (УО-1); контрольная работа (ПР-2)	Экзамен, вопросы 1, 5, 11, 15, 21.
			владеет	Контрольная работа (ПР-2); тесты (ПР-1)	Экзамен, вопросы 1, 5, 11, 15, 21.
2	Архитектура параллельных систем	ОПК-2 ОПК-7 ПК-4 ПК-6 ПК-9	знает	Устный опрос (УО-1); тесты (ПР-1)	Экзамен, вопросы 3, 12, 16, 17, 19, 25, 27, 29.
			умеет	Устный опрос (УО-1); контрольная работа (ПР-2)	Экзамен, вопросы 3, 12, 16, 17, 19, 25, 27, 29.
			владеет	Контрольная работа (ПР-2); тесты (ПР-1)	Экзамен, вопросы 3, 12, 16, 17, 19, 25, 27, 29.
3	Модели	ОПК-2	знает	Устный опрос	Экзамен,

	параллельных вычислений	ОПК-7 ПК-4 ПК-6 ПК-9		(УО-1); тесты (ПР-1)	вопросы 5, 7, 9, 10, 11, 13, 21, 23
			умеет	Устный опрос (УО-1); контрольная работа (ПР-2)	Экзамен, вопросы 5, 7, 9, 10, 11, 13, 21, 23
			владеет	Контрольная работа (ПР-2); тесты (ПР-1)	Экзамен, вопросы 5, 7, 9, 10, 11, 13, 21, 23
3	Реализация параллельных алгоритмов	ОПК-2 ОПК-7 ПК-4 ПК-6 ПК-9	знает	Устный опрос (УО-1); тесты (ПР-1)	Экзамен, вопросы 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32
			умеет	Устный опрос (УО-1); контрольная работа (ПР-2)	Экзамен, вопросы 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32
			владеет	Контрольная работа (ПР-2); тесты (ПР-1)	Экзамен, вопросы 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32

Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в разделе IX.

V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература

(электронные и печатные издания)

1. Соснин В.В. Введение в параллельные вычисления [Электронный ресурс] / В.В. Соснин, П.В. Балакшин. — Электрон. текстовые данные. —

СПб. : Университет ИТМО, 2015. — 54 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/68646.html>

2. Параллельные вычисления на GPU. Архитектура и программная модель CUDA [Электронный ресурс] : учебное пособие / А.В. Боресков [и др.]. — Электрон. текстовые данные. — М. : Московский государственный университет имени М.В. Ломоносова, 2015. — 336 с. — 978-5-19-011058-6. — Режим доступа: <http://www.iprbookshop.ru/54647.html>

3. Элементы имитационного моделирования процессов функционирования информационно-вычислительных систем [Электронный ресурс] : практикум по дисциплине Архитектура вычислительных систем / . — Электрон. текстовые данные. — М. : Московский технический университет связи и информатики, 2014. — 8 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/63376.html>

4. Мищенко В.К. Архитектура высокопроизводительных вычислительных систем [Электронный ресурс] : учебное пособие / В.К. Мищенко. — Электрон. текстовые данные. — Новосибирск: Новосибирский государственный технический университет, 2013. — 40 с. — 978-5-7782-2365-3. — Режим доступа: <http://www.iprbookshop.ru/44898.html>

5. Зиангирова Л.Ф. Вычислительные системы, сети и телекоммуникации [Электронный ресурс] : учебно-методическое пособие / Л.Ф. Зиангирова. — Электрон. текстовые данные. — Саратов: Вузовское образование, 2015. — 150 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/31942.html>.

6. Ёранссон, А. Эффективное использование потоков в операционной системе Android [Электронный ресурс] / А. Ёранссон ; пер. с англ. Снастина А.В. — Электрон. дан. — Москва : ДМК Пресс, 2015. — 304 с. — Режим доступа: <https://e.lanbook.com/book/93268>. — Загл. с экрана.

Дополнительная литература

1. Николаев Е.И. Параллельные вычисления: учебное пособие / Николаев Е.И. — Ставрополь: Северо-Кавказский федеральный университет, 2016. — 185 с. — ISBN 2227-8397. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <http://www.iprbookshop.ru/66086.html>

2. Параллельные вычисления на GPU. Архитектура и программная модель CUDA: учебное пособие / А.В. Боресков [и др.]. — Москва: Московский государственный университет имени М.В. Ломоносова, 2015. — 336 с. — ISBN 978-5-19-011058-6. — Текст: электронный // Электронно-

библиотечная система IPR BOOKS: [сайт]. — URL:
<http://www.iprbookshop.ru/54647.html>

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Programming Guide: CUDA Toolkit Documentation.
<https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html>
2. OpenMP 4.0 API C/C++ Syntax Quick Reference Card.
<https://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf>
3. Cppreference.com. <https://en.cppreference.com/w/>

Перечень информационных технологий и программного обеспечения

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используется следующее программное обеспечение:

1. Среды разработки ПО Microsoft Visual Studio Community 2015 и Microsoft Visual Studio Community 2017.
2. Компиляторы gcc, g++ версии не ниже 6.2.0, а также отладчик gdb.
3. Пакет Nvidia CUDA Toolkit версии не ниже 9.0.
4. Microsoft Office (Access, Excel, PowerPoint, Word и т. д).
5. Open Office.
6. Skype.

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используются следующие информационно-справочные системы:

1. Научная электронная библиотека eLIBRARY.
2. Электронно-библиотечная система издательства «Лань».
3. Электронная библиотека "Консультант студента".
4. Электронно-библиотечная система IPRbooks.
5. Информационная система "ЕДИНОЕ ОКНО доступа к образовательным ресурсам".
6. Доступ к электронному заказу книг в библиотеке ДВФУ, доступ к нормативным документам ДВФУ, расписанию, рассылке писем.

Лабораторные занятия проводятся в специализированном компьютерном классе.

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Для достижения поставленных целей преподавания дисциплины реализуются следующие средства, способы и организационные мероприятия:

изучение теоретического материала дисциплины на лекциях с использованием компьютерных технологий;

самостоятельное изучение теоретического материала дисциплины с использованием Internet-ресурсов, методических разработок, специальной учебной и научной литературы;

закрепление теоретического материала при проведении лабораторных работ, выполнение проблемно-ориентированных, творческих заданий.

подготовка материалов для выступления на семинарах по темам курса, участие в дискуссиях.

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Лекции проводятся с использованием проектора и внутренней системы портала ДВФУ. Лабораторные занятия проходят в аудиториях, оборудованных компьютерами типа Lenovo C360G-i34164G500UDK с лицензионными программами Microsoft Office 2013 и аудиовизуальными средствами проектор Panasonic DLPProjectorPT-D2110XE, плазма LG FLATRON M4716CCBAM4716CJ. Для выполнения самостоятельной работы студенты в жилых корпусах ДВФУ обеспечены Wi-Fi. Лабораторные занятия проводятся в компьютерном классе.

№	Наименование	Кол-во
1	Библиотечный фонд ДВФУ	
2	Учебные классы ДВФУ С общим количеством: - посадочных мест - рабочих мест (компьютер+монитор) - проекторов, экранов	1 10 10 3
3	Рабочие места с выходом в интернет	10
4	Вычислительные системы коллективного пользования ДВФУ Из них: Вычислительных кластеров с архитектурой x86	3 1

	Вычислительных кластеров с архитектурой x86_64	1
	Вычислительных кластеров с архитектурой x86_64+GPU	1

VIII. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

План-график выполнения самостоятельной работы по дисциплине

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1	2 неделя обучения	Составление и анализ параллельного алгоритма подсчета элементов с заданным свойством в векторе.	2 часа	Проект
2.	4 неделя обучения	Составление и реализация параллельных алгоритмов длинной арифметики.	2	Проект
3.	4 неделя обучения	Быстрая сортировка с использованием параллельных вычислений.	2	Проект
4.	4 неделя обучения	Инструментальные средства параллельных вычислений в C11 и C++17.	2	Собеседование
5.	6 неделя обучения	Инструменты автоматизированного анализа параллельного кода программ.	2	Собеседование
6.	8 неделя обучения	Оптимизация доступа к	4	Проект

		разделяемой памяти блока CUDA на примере задачи умножения матриц.		
7.	8 неделя обучения	Методы ограничения переупорядочивания инструкций процессора и языковых конструкций при компиляции.	4	Собеседование
8.	9 неделя обучения	Распределение вычислений и инструменты MPI.	2	Проект
9.	9 неделя обучения	Реализация параллельного сложения векторов с помощью OpenCL.	2	Проект

Самостоятельные работы проводятся на рабочих местах с доступом к ресурсам Internet и в домашних условиях. Порядок выполнения самостоятельной работы соответствует программе курса и контролируется в ходе аудиторных занятий. Самостоятельная работа подкрепляется учебно-методическим и информационным обеспечением, включающим рекомендованные учебники и учебно-методические пособия.

Самостоятельная работа считается выполненной, в отчете по проделанной работе представлено письменные пояснения к полученным выводам и, если требуется, код программной реализации, компилируемый и выполняющий задачу корректно.

Проводится проверка правильности выполнения заданий по самостоятельной работе. Задание зачтено, если нет ошибок. По текущим ошибкам даются пояснения.

IX. ФОНДЫ ОЦЕНОЧНЫХ СРЕДСТВ

№ п/ п	Контролируемые разделы / темы дисциплины	Коды и этапы формирования компетенций		Оценочные средства	
				текущий контроль	промежуточная аттестация
1	Обоснование параллельных вычислений	ОПК-2 ОПК-7 ПК-4 ПК-6 ПК-9	знает	Устный опрос (УО-1); тесты (ПР-1)	Экзамен, вопросы 1, 5, 11, 15, 21.
			умеет	Устный опрос (УО-1); контрольная работа (ПР-2)	Экзамен, вопросы 1, 5, 11, 15, 21.
			владеет	Контрольная работа (ПР-2); тесты (ПР-1)	Экзамен, вопросы 1, 5, 11, 15, 21.
2	Архитектура параллельных систем	ОПК-2 ОПК-7 ПК-4 ПК-6 ПК-9	знает	Устный опрос (УО-1); тесты (ПР-1)	Экзамен, вопросы 3, 12, 16, 17, 19, 25, 27, 29.
			умеет	Устный опрос (УО-1); контрольная работа (ПР-2)	Экзамен, вопросы 3, 12, 16, 17, 19, 25, 27, 29.
			владеет	Контрольная работа (ПР-2); тесты (ПР-1)	Экзамен, вопросы 3, 12, 16, 17, 19, 25, 27, 29.
3	Модели параллельных вычислений	ОПК-2 ОПК-7 ПК-4 ПК-6 ПК-9	знает	Устный опрос (УО-1); тесты (ПР-1)	Экзамен, вопросы 5, 7, 9, 10, 11, 13, 21, 23
			умеет	Устный опрос (УО-1); контрольная работа (ПР-2)	Экзамен, вопросы 5, 7, 9, 10, 11, 13, 21, 23
			владеет	Контрольная работа (ПР-2); тесты (ПР-1)	Экзамен, вопросы 5, 7, 9, 10, 11, 13, 21, 23
3	Реализация параллельных алгоритмов	ОПК-2 ОПК-7 ПК-4	знает	Устный опрос (УО-1); тесты (ПР-1)	Экзамен, вопросы 2, 4, 6, 8, 10, 12, 14, 16,

		ПК-6 ПК-9			18, 20, 22, 24, 26, 28, 30, 32
			умеет	Устный опрос (УО-1); контрольная работа (ПР-2)	Экзамен, вопросы 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32
			владеет	Контрольная работа (ПР-2); тесты (ПР-1)	Экзамен, вопросы 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32

Шкала оценивания уровня сформированности компетенций

Код и формулировка компетенции	Этапы формирования компетенции		критерии	показатели
ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	знает (пороговый уровень)	основные направления развития вычислительных машин, комплексов и сетей, критерии применимости традиционных и нетрадиционных параллельных вычислительных архитектур для решения различных проблем предметных областей.	Знание основных архитектур и интерфейсов параллельных вычислительных платформ и связанных с этим ограничений на реализацию.	Способность выбрать один из набора методов параллельных вычислений для произвольной предметной области на основе оценки их реализуемости и масштабируемости; способность спрогнозировать эффективность возможности реализации на наборе возможных параллельных вычислительных архитектур.
	умеет (продвинутой)	создавать и применять вычислительные системы, как универсального, так и специализированного	Умение оценить целесообразность применения параллельных вычислений, выбрать наиболее подходящую для	Способность адекватного анализа проблемы и выбора соответствующей архитектуры параллельной

		назначения, для решения вычислительных задач предметной области.	решения вычислительной проблемы архитектуру вычислительной системы, оценить предоставляемые ей инструменты.	вычислительной платформы для решения выбранных вычислительных задач.
	владеет (высокий)	методами алгоритмического представления вычислительных проблем в формах, реализуемых на параллельных вычислителях традиционных и нетрадиционных архитектур.	Владение навыками работы с программными средствами и системами, реализующими параллельные вычисления.	Способность реализовать заданный метод параллельных вычислений при помощи программного и аппаратного средства, а также получить результат на основе выборки реальных данных.
ПК-4. Готовность к использованию методов и инструментальных средств исследования объектов профессиональной деятельности	знает (пороговый уровень)	современные инструментальные средства проектирования, анализа, реализации, отладки и тестирования параллельных программ для параллельных вычислителей основных современных архитектур.	Знание методов анализа, тестирования и отладки параллельных алгоритмов и программ.	Способность осуществить и обосновать выбор адекватных инструментов для реализации параллельных вычислений для выбранного набора параллелизуемых задач.
	умеет (продвинутый)	применять современные универсальные и специализированные инструментальные средства моделирования, анализа и разработки программ, формулировать	Умение применять методы анализа проблемы для проектирования, реализации и поддержки параллельных программ, выбирать и использовать адекватные инструменты	Способность осуществить и обосновать выбор адекватных методов параллельных вычислений для выбранного класса параллелизуемых задач.

		и оценивать эффективность различных параллельных реализаций.	для моделирования, реализации, тестирования и отладки параллельных алгоритмов и программ.	
	владеет (высокий)	навыками работы с основными инструментальными средствами проектирования, реализации и разработки программного обеспечения для основных параллельных вычислительных архитектур.	Владение навыками работы со средствами анализа, моделирования, тестирования и отладки параллельных алгоритмов и программ.	Способность применить навыки владения средствами анализа и моделирования параллельных алгоритмов и программ для решения типичных вычислительных задач предметной области.
ОПК-7. Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой	знает (пороговый уровень)	концептуальные основы параллельных вычислений, области применимости основных параллельных алгоритмов, оценки эффективности параллельных программ, критерии параллелизуемости последовательных алгоритмов, модели параллельных вычислений	Знание сущности и положения информатики, задающие концептуальную основу параллельной алгоритмизации и разработки параллельного программного обеспечения, существующие методы синтеза и анализа параллельных алгоритмов, оценки их эффективности и методов описания.	Знание положения теории графов и ее применение для построения и исследования параллельных форм алгоритмов, оценки их эффективности; положения теории сложности и ее применение для оценки параллельных алгоритмов
	умеет (продвинутой)	выполнять моделирование, синтез, анализ, параллельных алгоритмов, оценку их эффективности	Умение составить модель проблемы, представить параллельные формы	Способность применить адекватные средства представления модели параллельных

		с использованием фундаментальных понятий информатики и дискретной математики.	реализующих алгоритмов и оценить их эффективность; применить адекватные методы реализации параллельных вычислений.	вычислений для задач выбранных классов.
	владеет (высокий)	Имеет навыки синтеза и анализа параллельных алгоритмов решения типовых вычислительных задач с заданными функциональными требованиями к эффективности	Владение навыками синтеза и анализа параллельных алгоритмов решения типовых вычислительных задач с заданными функциональными требованиями к эффективности	Способность применить навыки выбора модели и реализации параллельных алгоритмов для вычислительных задач выбранных классов.
ПК-6. Владение навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения	знает (пороговый уровень)	методы моделирования и анализа параллельных алгоритмов и программ при решении задач предметной области.	Знание методов моделирования и анализа параллельных алгоритмов и программ при решении задач предметной области.	Способность продемонстрировать знание методов моделирования и представления параллельных вычислений для задач различных классов.
	умеет (продвинутой)	создавать и применять модели информационных технологий и проблем предметных областей для решения вычислительных задач в предметных областях.	Умение составить модель проблемы, представить параллельные формы реализующих алгоритмов и оценить их эффективность; применить адекватные методы реализации параллельных вычислений.	Способность применить адекватные средства представления модели параллельных вычислений для задач выбранных классов.

	владеет (высокий)	методами синтеза и анализа моделей информационных технологий, а также применения этих моделей для решения вычислительных задач в предметных областях	Владение навыками синтеза и анализа моделей информационных технологий, а также применения этих моделей для решения вычислительных задач в предметных областях.	Способность применить навыки выбора модели и реализации параллельных алгоритмов для вычислительных задач выбранных классов.
ПК-9. Владение навыками использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных	знает (пороговый уровень)	.	Знание основных инструментов программирования и операционных систем для реализации параллельных вычислений, логические сущности, предоставляемые параллельными платформами для параллельного программирования, критерии их применимости, их вклад в функциональную эффективность	Способность продемонстрировать знания основных инструментов программирования и операционных систем для реализации параллельных вычислений, логические сущности, предоставляемые параллельными платформами для параллельного программирования, критерии их применимости, их вклад в функциональную эффективность
	умеет (продвинутой)	выбирать адекватные решаемой вычислительной задаче параллельные методы и средства языков программирования,	Умение выбирать адекватные решаемой вычислительной задаче параллельные методы и средства языков программирования	Способность выбирать адекватные решаемой вычислительной задаче параллельные методы и средства языков программирования

		операционных систем и платформ; формально обосновывать свой выбор.	ия, операционных систем и платформ; формально обосновывать свой выбор.	ия, операционных систем и платформ; формально обосновывать свой выбор
	владеет (высокий)	навыками адекватного применения сущностей интерфейсов реализации параллельных вычислений, на основе функциональных требований к эффективности, а также требований инкапсуляции и связанности интерфейсов параллельной реализации.	Владение навыками адекватного применения сущностей интерфейсов реализации параллельных вычислений, на основе функциональных требований к эффективности, а также требований инкапсуляции и связанности интерфейсов параллельной реализации.	Способность применить навыки адекватного применения сущностей интерфейсов реализации параллельных вычислений, на основе функциональных требований к эффективности, а также требований инкапсуляции и связанности интерфейсов параллельной реализации

Методические рекомендации, определяющие процедуры оценивания результатов освоения дисциплины

Текущая аттестация студентов. Текущая аттестация студентов по дисциплине «Параллельное программирование» проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

Текущая аттестация по дисциплине «Параллельное программирование» проводится в форме контрольных мероприятий (собеседования) по оцениванию фактических результатов обучения студентов и осуществляется ведущим преподавателем.

Объектами оценивания выступают:

- учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость всех видов занятий по аттестуемой дисциплине);

- степень усвоения теоретических знаний;

- уровень овладения практическими умениями и навыками по всем видам учебной работы;

- результаты самостоятельной работы.

Промежуточная аттестация студентов. Промежуточная аттестация студентов по дисциплине «Параллельное программирование» проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

По дисциплине «Параллельное программирование» предусмотрен экзамен (устный опрос в форме ответов на вопросы экзаменационных билетов).

Оценочные средства для текущей аттестации

Проводится проверка правильности выполнения заданий по самостоятельной работе. Задание зачтено, если нет ошибок. По текущим ошибкам даются пояснения.

Вопросы для собеседования

по дисциплине «Параллельное программирование»

1. Инструментальные средства параллельных вычислений в C11 и C++17.
2. Инструменты автоматизированного анализа параллельного кода программ.
3. Методы ограничения переупорядочивания инструкций процессора и языковых конструкций при компиляции.

Критерии оценки устного ответа

✓ 100-86 баллов - если ответ показывает глубокое и систематическое знание всего программного материала и структуры конкретного вопроса, а также основного содержания и новаций лекционного курса по сравнению с учебной литературой. Студент демонстрирует отчетливое и свободное владение концептуально-понятийным аппаратом, научным языком и терминологией соответствующей научной области. Знание основной литературы и знакомство с дополнительно рекомендованной литературой. Логически корректное и убедительное изложение ответа.

✓ 85-76 - баллов - знание узловых проблем программы и основного содержания лекционного курса; умение пользоваться концептуально-понятийным аппаратом в процессе анализа основных проблем в рамках

данной темы; знание важнейших работ из списка рекомендованной литературы. В целом логически корректное, но не всегда точное и аргументированное изложение ответа.

✓ 75-61 - балл – фрагментарные, поверхностные знания важнейших разделов программы и содержания лекционного курса; затруднения с использованием научно-понятийного аппарата и терминологии учебной дисциплины; неполное знакомство с рекомендованной литературой; частичные затруднения с выполнением предусмотренных программой заданий; стремление логически определенно и последовательно изложить ответ.

✓ 60-50 баллов – незнание, либо отрывочное представление о данной проблеме в рамках учебно-программного материала; неумение использовать понятийный аппарат; отсутствие логической связи в ответе.

Оценочные средства для промежуточной аттестации

Вопросы к экзамену

1. Классификация архитектур вычислительных систем. (Классификация Флинна).

2. Реализовать алгоритм

$$w_ave(x_1, x_2) = \frac{1}{x_2 - x_1} \int_{x_1}^{x_2} f(x)w(x)dx$$

расчета средневзвешенного значения функции $f(x)$ на интервале $x_1 \leq x \leq x_2$ параллельно, если задана весовая функция $w(x)$, также определенная на интервале $x_1 \leq x \leq x_2$. Вход и выход обеих функций – числа с плавающей точкой двойной точности (double).

Оценить эффективность параллельного выполнения такого вычисления T потоками выполнения.

Пусть функция $f(x)$ имеет побочные эффекты, которые могут приводить к гонкам. Какие изменения будет необходимо внести в реализацию, и как это повлияет на эффективность решения задачи?

3. Основные архитектуры параллельных компьютеров.

4. Пусть задана квадратная матрица A значений с N столбцами, N строками и с рангом N . Реализовать ядро CUDA (и вспомогательные функции устройства), вычисляющее обратную матрицу A^{-1} . При условии, что все элементы матрицы A могут быть умещены в собственной разделяемой памяти одного блока CUDA, а число $N^2 \leq T$, где T – максимальное число

потоков CUDA на блок, реализовать алгоритм с использованием разделяемой памяти CUDA.

Интерфейс:

```
void Inverse(/*in*/ const double* A, size_t N, /*out*/ double* AInv);
```

5. Модели параллельного программирования. Основные свойства параллельных алгоритмов.

6. Пусть имеется оцифрованный звуковой сигнал, заданный набором уровней исходного сигнала на дискретном наборе временных отчетов. Такое цифровое представление звукового сигнала можно рассматривать как вектор уровней сигнала.

Пусть уровень звука момент времени определяется 32-х битовым числом с плавающей точкой. С помощью CUDA реализовать фильтр, осуществляющий вейвлет-преобразование Хаара $h(V, N)$, так, чтобы для вектора $V = \{v_i\}_{0 \leq i < N}$ из $N = 2^k$ элементов

$$h(V, N) = \begin{cases} \{v_0\}, N \equiv 1 \\ h\left(\bigcup_{i=0}^{\frac{N}{2}-1} \left\{\frac{v_{2i} + v_{2i+1}}{2}\right\}, \frac{N}{2}\right) \cup \bigcup_{i=0}^{\frac{N}{2}-1} \left\{\frac{v_{2i} - v_{2i+1}}{2}\right\}, N > 1 \end{cases}$$

Интерфейс:

```
void h(const float* input, size_t N, float* output);
```

7. Модель задача-канал. Основные свойства модели "задача/канал".

8. Пусть имеется некоторый канал, через который многопоточному серверу передаются запросы $M = \{m_i\}$ на последовательные вычисления, объем которых независим от запроса и равен t . Опишите возможную логическую организацию вычислений на сервере с T логическими процессорами. Опишите узкие места в предложенной организации и способ их устранения.

Если возможно атомарное получение количества потоков выполнения, единовременно выполняющих ожидание прихода через канал сообщения, то как повлияет на реализацию и оперативность сервера введение дополнительного параллельного и независимого канала?

9. Модель передачи сообщений.

10. С помощью переменных условий реализовать (с помощью Posix или C11 или C++11 threads) событие, под которым понимается механизм синхронизации со следующим интерфейсом:

Создание экземпляра: без параметров.

Ожидание события с блокировкой ожидающего потока:

```
void wait_event(event);
```

Перевод события в сигнальное состояние:

```
void set_event(event);
```

Сброс события из сигнального состояния:

```
void reset_event(event);
```

Событие, не находящееся в сигнальном состоянии должно блокировать поток, вызывающий метод `wait_event` над ним.

Когда событие переводится в сигнальное состояние, все потоки, ожидающие событие, разблокируются. Потоки, вызывающие `wait_event` над событием, находящимся в сигнальном состоянии, не блокируются.

Событие остается в сигнальном состоянии до тех пор, пока не будет вызван метод `reset_event` над ним.

Только что созданное событие находится в несигнальном состоянии.

11. Модель параллелизма данных.

12. С помощью переменных условий реализовать (с помощью Posix или C11 или C++11 threads) семафор, под которым понимается механизм синхронизации, который ассоциирован с некоторым целочисленным счетчиком потоков. Завладение семафором означает декремент положительного счетчика или, если счетчик равен нулю, блокировку вызывающего потока выполнения до тех пор, пока счетчик не окажется ненулевым. Инкремент счетчика потоков, агрегируемого семафором, связан с операцией освобождения семафора. Семафор создается с заданным целым числом N – максимальным числом потоков, которые одновременно могут владеть экземпляром семафора.

Требуется реализовать следующий интерфейс.

Создание семафора: на вход процедуры создания семафора подается беззнаковое целое ненулевое N .

Завладение семафором:

```
void acquire_semaphore(semaphore);
```

Освобождение семафора:

```
void release_semaphore(semaphore);
```

13. Модель разделяемой памяти.

14. С помощью CUDA реализовать подсчет в случайном большом векторе V из N 32-х битовых целых элементов, которые делятся на два, три и пять.

Интерфейс:

```
void count_multiplies(const int * V, size_t N, unsigned* m2, unsigned * m3, unsigned * m5);
```

15. Совместное и параллельное выполнение программ (Concurrency vs Parallelism).

16. В языке C и в C-подобных (напр., php, perl) языках существует функция `rand()` без параметров, возвращающая псевдослучайное целое число на основе некоторого глобального состояния, которое переносит в себе энтропию (которой, с помощью функции `srand` генератор был проинициализирован) от генерации к генерации. И во многих случаях доступ к этому глобальному состоянию из множества потоков параллельного выполнения не является потокобезопасным. Сформулируйте возможные подходы к реализации параллельного (псевдо-) случайного генератора последовательностей чисел, записываемых в выходной вектор. Укажите на достоинства и недостатки выбранных подходов.

17. Реализация совместных вычислений средствами операционных систем Unix, Windows. Потоки Posix.

18. С помощью CUDA и центрального процессора реализовать гетерогенные параллельные вычисления для генерации в выходной вектор значений (плавающая точка, одинарная точность) синусоиды для заданных диапазоном (x_0, x_{N-1}) и (большим) шагом N значений абсциссы.

Интерфейс:

```
void generate(float* buffer, size_t N, float x0, float xN_1);
```

19. Языковые инструменты для параллельных и совместных вычислений: OpenMP, потоки C11 и C++11.

20. Составить параллельный алгоритм подсчета частот байтовых символов в векторе байт `input` длины N . Результатом должна являться таблица частот – вектор `weights` из 256-ти целых типа `unsigned`.

Реализовать алгоритм для CUDA.

Интерфейс:

```
void gather_weights(const uint8_t* input, size_t N, unsigned* weights);
```

21. Детерминизм параллельных алгоритмов и программ.

22. Пусть двумя векторами $X = \{x_i\}_{0 \leq i < N}$ и $Y = \{y_i\}_{0 \leq i < N}$ заданы два длинных целых беззнаковых числа. Показать параллельную реализацию умножения «в столбик» потоками центрального процессора. Произведение должно быть записано в предоставленный буфер $Z = \{z_i\}_{0 \leq i < 2N}$.

Интерфейс:

```
void parallel_mul(const unsigned* X, const unsigned* Y, size_t N, unsigned* Z);
```

23. Параллелизм, основанный на задачах.

24. Какова временная вычислительная сложность подсчета суммы всех значений в случайном векторе чисел с помощью редукции, если имеется неограниченное количество параллельных логических процессоров?

25. Инструменты синхронизации при реализации совместных вычислений: мьютексы и мониторы.

26. Пусть задан некоторый двусвязный список $list = \{(L, V_0, R)_i\}_{0 \leq i < N}$, каждый i -ый узел которого хранит адреса соседних узлов, левого – L_i и правого – R_i , а также ассоциированное с ним значение $V_{0,i}$ с плавающей точкой одинарной точности (`float`). Для крайнего левого элемента указатель L_0 равен `NULL`. Аналогично, для крайнего правого элемента списка указатель R_{N-1} равен `NULL`.

Ассоциировать каждый список с задачей и, используя SPMD-параллелизм, а также синхронизацию типа «барьер», реализовать алгоритм, который принимает на вход головной (первый) узел списка, а также значение t и изменяет значения $V_{0,i}$ на $V_{t,i}$ согласно формуле:

$$V_{t,i} = \frac{V_{t-1,i-1} + 2V_{t-1,i} + V_{t-1,i+1}}{4}$$

для $0 < i < N - 1$ и $t > 0$.

Реализовать алгоритм на CUDA с интерфейсом:

```
struct ListNode
{
    struct ListNode* L;
    struct ListNode* R;
```

```
float V;
};
void fin_diff(struct ListNode* list, unsigned t);
```

27. Вычисления на графических процессорах. SIMT-параллелизм.

28. Пусть имеется некоторое дерево, которое задается корнем. Элементы дерева никак не упорядочены. Записать алгоритм параллельного поиска элемента с заданным значением в таком дереве, считая, что все элементы уникальны по ассоциированным с ними значениям. Реализовать алгоритм, используя понятие задачи и динамический параллелизм на центральном процессоре.

29. Ускорители вычислений NVIDIA CUDA. Программно-аппаратная архитектура CUDA.

30. С помощью потоков CUDA и, отдельно, потоков CPU реализовать быструю сортировку вектора $V = \{v_i\}_{0 \leq i < N}$. Учитывать ограничения на доступ к памяти за пределами буфера $v_0 \dots v_{N-1}$.

Интерфейс:

```
void parallel_qsort(const float* V, size_t N, float* V_sorted);
```

Буферы V и V_sorted разные и не пересекаются.

31. Гетерогенные вычисления с использованием центральных и графических процессоров.

32. Пусть на двумерном Евклидовом пространстве задан вектор точек $P = \{p_i\}_{0 \leq i < N}$, $N > 2$. Создать параллельный алгоритм поиска пары ближайших друг к другу точек, используя стратегию «разделяй и властвуй». Реализовать алгоритм на центральном процессоре, используя понятие задачи и динамический параллелизм.

```
struct point_t
```

```
{
```

```
    float x, y;
```

```
};
```

```
void closest_points(const point_t* P, size_t N, point_t* p1, point_t* p2);
```

Критерии выставления оценки студенту на экзамене

Баллы (рейтингов ой оценки)	Оценка экзамена (стандартная)	Требования к сформированным компетенциям
86-100	«отлично»	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.

76-85	«хорошо»	Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения.
61-75	«удовлетворительно»	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ.
0-60	«неудовлетворительно»	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.