


МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Чувашский государственный университет имени И. Н. Ульянова»

Факультет информатики и вычислительной техники

Кафедра вычислительной техники

«УТВЕРЖДАЮ»
Проректор по учебной работе


И.Е. Поверинов

31 августа 2017 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ»



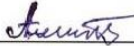

Направление подготовки (специальность) – 10.05.03 «Информационная безопасность автоматизированных систем»

Квалификация (степень) выпускника – Специалист по защите информации

Специализация – «Безопасность открытых информационных систем»

Рабочая программа основана на требованиях Федерального государственного образовательного стандарта высшего образования по специальности 10.05.03 «Информационная безопасность автоматизированных систем», утвержденного приказом Министерства образования и науки 01.12.2016 г. №1509

СОСТАВИТЕЛЬ (СОСТАВИТЕЛИ):

старший преподаватель  А.Г. Алексеев
кандидат технических наук, доцент  А.А. Андреева

ОБСУЖДЕНО:

на заседании кафедры вычислительной техники «30» августа 2017 г., протокол № 1


заведующий кафедрой  А.В. Щипцова


СОГЛАСОВАНО:

Методическая комиссия факультета информатики и вычислительной техники «30» августа 2017 г., протокол № 1

Декан факультета  А.В. Щипцова

Директор научной библиотеки  Н. Д. Никитина

Начальник управления информатизации  И. П. Пивоваров

Начальник учебно-методического управления  В. И. Маколов

Оглавление

1. Цель и задачи обучения по дисциплине	4
2. Место дисциплины в структуре основной образовательной программы (ООП).....	4
3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ООП	4
4. Структура и содержание дисциплины	5
5. Содержание разделов дисциплины.....	6
6. Образовательные технологии	8
7. Формы аттестации и оценочные материалы	9
8. Учебно-методическое и информационное обеспечение дисциплины	11
9. Материально-техническое обеспечение дисциплины	13
10. Средства адаптации преподавания дисциплины к потребностям лиц с ограниченными возможностями	13
11. Методические рекомендации по освоению дисциплины	14

1. Цель и задачи обучения по дисциплине

Цель преподавания дисциплины – формирование у студентов знаний основ параллельного программирования: моделей параллельных систем с разделяемой/распределенной памятью, методов взаимодействия и синхронизации выполнения параллельных задач, особенностей реализации потоков (threads) в операционных системах POSIX и Windows.

Задачами преподавания дисциплины являются:

- знакомство с архитектурой современных многоядерных процессоров;
- изучение особенностей реализации многопоточности в различных ОС и языках программирования;
- знакомство с принципами построения и функционирования современных суперкомпьютеров;
- формирование умения правильно выбирать и использовать инструментальные средства в зависимости от поставленной задачи и архитектуры целевой системы;
- получение практического опыта разработки параллельных программ с использованием языков программирования C/C++, Java и современного инструментария (OpenMP, Intel TBB, MPI, CUDA/OpenCL).

2. Место дисциплины в структуре основной образовательной программы (ООП)

Дисциплина реализуется в рамках дисциплин по выбору вариативной части образовательной программы бакалавра.

Изучение данной дисциплины базируется на следующих курсах:

«Языки программирования» – знать и уметь применять языки программирования высокого уровня.

«Безопасность операционных систем» – иметь представление о диспетчеризации процессов/потоков и разделении доступа к разделяемым ресурсам в современных операционных системах.

«Иностранный язык» (английский) – уметь свободно читать и переводить техническую литературу, осуществлять поиск информации на иностранном языке в сети Internet.

Дисциплина является предшествующей для прохождения практик, государственной итоговой аттестации.

3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ООП

способность к освоению новых образцов программных, технических средств и информационных технологий (ОПК-8);

способность применять знания в области электроники и схемотехники, технологий, методов и языков программирования, технологий связи и передачи данных при разработке программно-аппаратных компонентов защищенных автоматизированных систем в сфере профессиональной деятельности (ПК-10).

В результате обучения по дисциплине, обучающийся должен (ЗУН):

знать:

- основы проектирования параллельных программ (31),
- особенности реализации потоков в ОС POSIX и Windows (32),
- принципы построения и функционирования современных суперкомпьютеров (33),
- основы гетерогенных параллельных вычислений (34).

уметь:

- правильно выбирать инструментальные средства в зависимости от поставленной задачи и архитектуре целевой системы при проектировании параллельной программы (У1),
 - разрабатывать параллельные программы для систем с разделяемой и распределенной памятью (У2),
 - оценивать эффективность параллельных программ (У3).
- владеть навыками:
- разработки параллельных программ для систем с разделяемой памятью (многоядерные процессоры) под ОС POSIX и Windows (Н1),
 - разработки параллельных программ для систем с распределенной памятью (Н2),
 - разработки параллельных программ для гетерогенных систем с использованием технологий CUDA/OPENCL (Н3),
 - отладки и поиска ошибок в параллельных программах (Н4),
 - оценки эффективности полученной параллельной программы (Н5).

4. Структура и содержание дисциплины

Образовательная деятельность по дисциплине проводится:

- в форме контактной работы обучающихся с педагогическими работниками организации и (или) лицами, привлекаемыми организацией к реализации образовательных программ на иных условиях (далее – контактная работа);
- в форме самостоятельной работы.

Контактная работа включает в себя занятия лекционного типа, занятия семинарского типа (практические занятия, лабораторные работы), групповые и (или) индивидуальные консультации, в том числе в электронной информационно-образовательной среде.

Обозначения:

Л – лекции, л/р – лабораторные работы, п/р – практические занятия, КСР – контроль самостоятельной работы, СРС – самостоятельная работа студента, ИФР – интерактивная форма работы, К – контроль.

4.1. Содержание дисциплины

Содержание	Формируемые компетенции	Формируемые ЗУН
Раздел 1. Проектирование параллельных программ	ОПК-8, ПК-10	31, У1, У3, Н4, Н5
1.1. Введение в параллельное программирование.		
1.2. Проектирование параллельных программ		
Раздел 2. Разработка параллельных программ с разделяемой памятью	ОПК-8, ПК-10	32, У2, У3, Н1, Н4, Н5
2.1. OpenMP		
2.2. Intel Threading Building Blocks		
2.3. Поток в ОС Windows		
2.4. POSIX Threads		
2.5. Поток Java		
Раздел 3. Разработка параллельных программ с распределенной памятью	ОПК-8, ПК-10	33, У2, У3, Н2, Н4, Н5
3.1. Message Passing Interface		
Раздел 4. Гетерогенные параллельные вычисления	ОПК-8, ПК-10	34, У2, У3, Н3, Н4, Н5
4.1. Технологии CUDA и OPENCL		
Зачет	ОПК-8, ПК-10	31, 32, 33, 34, У1, У2, У3, Н1, Н2, Н3, Н4, Н5

4.2. Объем дисциплины, виды учебной работы обучающихся по очной форме обучения

Содержание	Всего, час	Контактная работа, час				СРС, час	ИФР, час	К, час
		Л	л/р	п/р	КСР			
Раздел 1. Проектирование параллельных программ								
1.1. Введение в параллельное программирование.	1	1						
1.2. Проектирование параллельных программ	1	1						
Раздел 2. Разработка параллельных программ с разделяемой памятью								
2.1. OpenMP	20	2	4	4		10	8	
2.2. Intel Threading Building Blocks	12	2	2	2		6	4	
2.3. Потoki в ОС Windows	13	2	2	2		7	4	
2.4. POSIX Threads	15	2	2	2		9	4	
2.5. Потoki Java	14	2	2	2		8	4	
Раздел 3. Разработка параллельных программ с распределенной памятью								
3.1. Message Passing Interface	18	2	2	2		12	4	
Раздел 4. Гетерогенные параллельные вычисления								
4.1. Технологии CUDA и OPENCL	12	2	2	2		6	4	
Зачет	2				2			
Итого	108	16	16	16	2	58	32	
з.е.	3							

5. Содержание разделов дисциплины

5.1. Лекции

Раздел 1. Проектирование параллельных программ.

Тема 1.1. Введение в параллельное программирование.

Лекция 1. Цели и задачи параллельного программирования. Ограничения одноядерных компьютеров. Развитие современных суперкомпьютеров. Список 500 самых мощных суперкомпьютеров мира. Тест LINPACK. Таксономия Флинна. Модели памяти в параллельных архитектурах. Архитектуры современных одно- и многоядерных процессоров. NUMA системы. Базовые определения: параллельная задача, коммуникация, синхронизация, гранулярность. Накладные расходы на параллелизм. Типовые модели параллельного программирования. Закон Амдала. Примеры параллельных задач. Сложность параллельного программирования.

Тема 1.2. Проектирование параллельных программ.

Лекция 2. Этапы проектирования параллельной программы. Преобразование последовательной программы в параллельную: декомпозиция по данным, функциональная декомпозиция. Рекурсивная декомпозиция. Коммуникация: “точка-точка” и коллективная. Виды коллективной коммуникации. Механизмы синхронизации выполнения параллельных задач: барьеры, семафоры, взаимное исключение (мьютекс), ждущая блокировка (спинлок), критическая секция, атомарные операции. Ошибки синхронизации: “гонка” и взаимная блокировка (клинч). Зависимость операторов по данным. Балансировка нагрузки. Гранулярность. Операции ввода-вывода в параллельной программе. Ресурсоемкость, масштабируемость и переносимость параллельных программ. Автоматический параллелизм. Базовые модели параллельных алгоритмов.

Раздел 2. Разработка параллельных программ с разделяемой памятью.

Тема 2.1. OpenMP.

Лекция 3. Стандарт OpenMP: версии стандарта, история развития, контролирующая организация. Поддержка стандарта в современных компиляторах. Модель выполнения в OpenMP. Модель памяти в OpenMP. Виды переменных в OpenMP: общие переменные и переменные потока. Пример: простейшая программа. Иллюстрация ситуации “гонка”. Распараллеливание циклов. Пример: умножение матриц. Диспетчеризация потоков в циклах. Пример: поиск простых чисел в указанном диапазоне. Параллелизм на основе задач. Пример: нахождение чисел Фибоначчи.

Лекция 4. Задача коммивояжера: постановка задачи и история возникновения. Источники исходных данных для задачи. Метод “грубой силы”. Последовательная версия программы. Преобразования последовательной версии в параллельную. Синхронизация доступа к общим переменным из разных потоков. Сравнительный анализ производительности при использовании критической секции и ждущей блокировки. Метод ветвей и границ. Сравнительный анализ последовательного метода ветвей и границ с параллельной версией “грубой силы”. Параллельная версия метода ветвей и границ.

Тема 2.2. Intel Threading Building Blocks.

Лекция 5. История возникновения, версии и состав библиотеки. Планировщик, его инициализация и особенности параллелизма в Intel TBB. Параллельные алгоритмы: `parallel_for`, `parallel_reduce`, `parallel_scan`, `parallel_do`, `parallel_sort`, `parallel_pipeline`. Ограничения на применение контейнеров STL в параллельных программах. Потокобезопасные контейнеры Intel TBB. “Ложное” разделение памяти. Работа с памятью в Intel TBB. Блокировки, атомарные операции и хронометр в Intel TBB. Рекурсивный параллелизм, обработка исключений. Совместимость с OpenMP.

Лекция 6. Конвейерная обработка данных, общие принципы. Пример: обработка видео. Знакомство с библиотекой компьютерного зрения OpenCV: открытие и сохранение видеофайла, реализация простого фильтра для кадровой обработки видеопотока. Реализация конвейера в Intel TBB. Преобразование последовательной версии в параллельную. Оценка полученного решения.

Тема 2.3. Потоки в ОС Windows.

Лекция 7. Процессы и потоки в ОС Windows: базовые определения, вытесняющая многозадачность, приоритеты. Создание потоков. Инверсия приоритетов и борьба с ней. Особенности выполнения потоков на NUMA системах. Объекты синхронизации потоков: события, взаимное исключение, семафоры, таймеры ожидания. функции ожидания (одного объекта, нескольких объектов, ожидания по адресу, ожидание с регистрацией).

Лекция 8. Атомарные операции. Критические секции. Барьеры синхронизации. Задача о писателях-читателях. Блокировки чтения-записи. Условные переменные. Легковесные потоки (fibers). Пулы потоков.

Тема 2.4. POSIX Threads.

Лекция 9. Стандарт POSIX. Дисциплины диспетчеризации потоков в POSIX: в порядке очереди, карусельная, спорадический сервер. Состояния потока, граф состояний потока. Типы данных в POSIX Threads. Создание и управление потоками.

Лекция 10. Типы и виды взаимного исключения (mutex). Пример: борьба с инверсией приоритетов в POSIX Threads. Задача об обедающих философах. Однократная инициализация, создание многопоточных библиотек. Локальное хранилище потока (TLS). Специфичные для Linux функции. Расширения POSIX для систем реального времени.

Тема 2.5. Потоки в языке Java.

Лекция 11. Потоки на уровне языка, специфика реализации потоков на виртуальной машине. Атрибуты и состояния потока. Два способа создания потоков в Java. Монитор, модель мониторов в Java.

Лекция 12. Атомарные операции. Потокобезопасные контейнеры. Блокировки. Объекты синхронизации: `CountDownLatch`, `CyclicBarrier`, `Semaphore`, `Phaser`. Создание масштабируемых многопоточных приложений: пулы потоков, исполнители (executors). Fork-Join Framework.

Раздел 3. Разработка параллельных программ с распределенной памятью.

Тема 3.1. Message Passing Interface.

Лекция 13. Распределенные вычисления. Ограничения многоядерных ЭВМ. Кластер. Топология кластера. Стандарт MPI, история версий. Реализации стандарта MPI: Open MPI, MPICH. Инициализация и завершение работы. Идентификация процесса. Режимы коммуникации «точка-точка». Блокирующий и неблокирующий обмен сообщениями. Соответствие между типами данных для языка C/C++. Пример: простейшая программа. Особенности запуска программы при выполнении в кластере.

Лекция 14. Виды коллективного взаимодействия. Группы процессов и коммутаторы. Виртуальные топологии: декартова и граф. Пример: умножение матриц на топологии «двухмерная сетка». Пример: умножение матриц на топологии «трехмерная сетка». Оценка времени выполнения участков кода. Односторонняя коммуникация. Создание многопоточных MPI приложений.

Раздел 4. Гетерогенные параллельные вычисления.

Тема 4.1. Технологии CUDA и OpenCL.

Лекция 15. Гетерогенные параллельные вычисления. История вычислений с использованием видеокарт. Технологии CUDA и OpenCL. Поколения устройств с поддержкой CUDA. Архитектура потокового мультипроцессора серии GeForce 8. Программная модель CUDA: устройство, вычислительный массив, блок, поток, warp. Встроенные переменные. Типы памяти, скорость доступа к различным типам памяти. Модификаторы переменных. Модель выполнения. Построение программ с использованием CUDA C и MSVS. Модификаторы функций.

Лекция 16. Этапы вычисления. Выделение и копирование памяти. Пример: сложение векторов. Эффективность операций с памятью. Пример: умножение матриц. Характеристики устройства. Ограничения по памяти. Определение параметров для достижения максимальной производительности – CUDA Occupancy Calculator. Дивергенция выполнения. Пример: фильтр обработки изображения. Библиотеки на основе CUDA. Соответствие терминов в CUDA и OpenCL.

5.2. Практические занятия и лабораторные работы

№	Тема	Практические занятия, количество часов	Лабораторные работы, количество часов
1	Накладные расходы на параллелизм	2	2
2	Алгоритмы параллельной сортировки	2	2
3	Конвейерная обработка данных	2	2
4	Потоки в ОС Windows	2	2
5	POSIX Threads	2	2
6	Параллельные алгоритмы на графах	2	2
7	Message Passing Interface	2	2
8	Технология CUDA	2	2
	Всего	16	16

6. Образовательные технологии

В соответствии со структурой образовательного процесса по дисциплине применяются следующие технологии:

- диагностики;
- целеполагания;
- управления процессом освоения учебной информации;

- применения знаний на практике, поиска новой учебной информации;
- организации совместной и самостоятельной деятельности обучающихся (учебно-познавательной, научно-исследовательской, частично-поисковой, репродуктивной, творческой и пр.);
- контроля качества и оценивания результатов образовательной деятельности (технология оценивания качества знаний, рейтинговая технология оценки знаний и др.)

В соответствии с требованиями ФГОС ВО для реализации компетентного подхода при обучении дисциплине предусмотрено широкое использование в учебном процессе активных и интерактивных методов проведения занятий:

При обучении дисциплине применяются следующие формы занятий:

- лекции, направленные на получение новых и углубление научно-теоретических знаний, в том числе вводная лекция, информационная лекция, обзорная лекция, лекция-консультация, проблемная лекция, лекции-дискуссии, лекции-беседы и др.;
- практические занятия, проводимые под руководством преподавателя в учебной аудитории, направленные на углубление и овладение определенными методами самостоятельной работы, могут включать коллективное обсуждение материала, дискуссии, решение и разбор конкретных практических ситуаций, компьютерные симуляции, тренинги и др.;
- лабораторные занятия, проводимые под руководством преподавателя в учебной лаборатории с использованием компьютеров и учебного оборудования, направленные на закрепление и получение новых умений и навыков, применение знаний и умений, полученных на теоретических занятиях, при решении практических задач и др.

Все занятия обеспечены мультимедийными средствами (SMART доски, проекторы, экраны) для повышения качества восприятия изучаемого материала. В образовательном процессе широко используются информационно-коммуникационные технологии.

Самостоятельная работа студентов – это планируемая работа студентов, выполняемая по заданию при методическом руководстве преподавателя, но без его непосредственного участия. Формы самостоятельной работы студентов определяются содержанием учебной дисциплины, степенью подготовленности студентов. Они могут иметь учебный или учебно-исследовательский характер: подготовка к лабораторным работам и др.

Формами контроля самостоятельной работы выступают проверка письменных отчетов по результатам выполненных заданий и лабораторных работ. Результаты самостоятельной работы учитываются при оценке знаний на зачете.

7. Формы аттестации и оценочные материалы

Текущий контроль предназначен для проверки хода и качества усвоения учебного материала, стимулирования учебной работы обучающихся и совершенствования методики проведения занятий. Он проводится в ходе всех видов занятий в форме, избранной преподавателем.

Промежуточная аттестация имеет целью определить степень достижения учебных целей по учебной дисциплине и проводится в форме зачета. Зачет принимается преподавателем, читающим лекции по данной учебной дисциплине в соответствии с перечнем основных вопросов, выносимых для контроля знаний обучающихся:

7.1. Вопросы к зачету:

1. Цели и задачи параллельного программирования. Ограничения одноядерных компьютеров.
2. Классификация систем по наличию в них параллелизма - таксономия Флинна. Примеры.
3. Модели памяти в параллельных архитектурах: разделяемая и распределенная память. Достоинства и недостатки каждой модели. Гибридная модель.

4. Архитектуры современных одно- и многоядерных процессоров. NUMA системы.
5. Закон Амдала. Накладные расходы на параллелизм.
6. Преобразование последовательной программы в параллельную: декомпозиция по данным, функциональная декомпозиция. Рекурсивная декомпозиция.
7. Коммуникация: “точка-точка” и коллективная. Виды коллективной коммуникации.
8. Механизмы синхронизации выполнения параллельных задач: барьеры, семафоры, взаимное исключение (мьютекс), ждущая блокировка (спинлок), критическая секция, атомарные операции.
9. Ошибки синхронизации: “гонка” и взаимная блокировка (клинч). Зависимость операторов по данным. Балансировка нагрузки. Гранулярность.
10. Ресурсоемкость, масштабируемость и переносимость параллельных программ.
11. Модель выполнения в OpenMP. Модель памяти в OpenMP. Виды переменных в OpenMP: общие переменные и переменные потока.
12. Диспетчеризация потоков в циклах: static, dynamic, guided.
13. Параллельные алгоритмы в Intel TBB: parallel_for, parallel_reduce, parallel_scan, parallel_do, parallel_sort, parallel_pipeline.
14. Ограничения на применение контейнеров STL в параллельных программах. Потокбезопасные контейнеры Intel TBB.
15. “Ложное” разделение памяти. Работа с памятью в Intel TBB.
16. Блокировки, атомарные операции и хронометр в Intel TBB.
17. Рекурсивный параллелизм и обработка исключений в Intel TBB.
18. Процессы и потоки в ОС Windows: базовые определения, вытесняющая многозадачность, приоритеты. Создание потоков.
19. Объекты синхронизации потоков в ОС Windows: события, взаимное исключение, семафоры, таймеры ожидания. функции ожидания (одного объекта, нескольких объектов, ожидания по адресу, ожидание с регистрацией).
20. Дополнительные средства синхронизации в ОС Windows: атомарные операции, критические секции, барьеры синхронизации, блокировки чтения-записи. Условные переменные.
21. Стандарт POSIX. Дисциплины диспетчеризации потоков в POSIX: в порядке очереди, карусельная, спорадический сервер. Состояния потока, граф состояний потока.
22. Создание и управление потоками в POSIX Threads. Типы и виды взаимного исключения. Борьба с инверсией приоритетов.
23. Однократная инициализация, создание многопоточных библиотек. Локальное хранилище потока (TLS). Специфичные для Linux функции. Расширения POSIX для систем реального времени.
24. Потоки на уровне языка (Java), специфика реализации потоков на виртуальной машине. Атрибуты и состояния потока. Два способа создания потоков в Java. Монитор, модель мониторов в Java.
25. Объекты синхронизации в Java: CountdownLatch, CyclicBarrier, Semaphore, Phaser. Атомарные операции.
26. Создание масштабируемых многопоточных приложений на языке Java: пулы потоков, исполнители (executors). Fork-Join Framework.
27. Распределенные вычисления. Ограничения многоядерных ЭВМ. Кластер. Топология кластера.
28. Инициализация и завершение работы в MPI. Идентификация процесса. Режимы коммуникации «точка-точка». Блокирующий и неблокирующий обмен сообщениями.
29. Виды коллективного взаимодействия в MPI. Группы процессов и коммутаторы. Виртуальные топологии: декартова и граф.

30. Программная модель CUDA: устройство, вычислительный массив, блок, поток, warp. Встроенные переменные.
31. Типы памяти в технологии CUDA, скорость доступа к различным типам памяти. Модификаторы переменных. Модель выполнения. Построение программ с использованием CUDA C и MSVS. Модификаторы функций.
32. Этапы вычисления в CUDA. Выделение и копирование памяти. Эффективность операций с памятью. Ограничения по памяти.
33. Определение параметров CUDA программы для достижения максимальной производительности. Дивергенция выполнения.
34. Классическая задача параллельного программирования: производитель-потребитель.
35. Классическая задача параллельного программирования: задача о читателях и писателях.
36. Классическая задача параллельного программирования: задача об обедающих философах.

Зачет проводится по окончании занятий по дисциплине до начала экзаменационной сессии в период недели контроля самостоятельной работы.

Билет для проведения промежуточной аттестации в форме зачета включают вопросы для проверки сформированности знаний, умений и навыков.

Оценка «зачтено» проставляется студенту, выполнившему и защитившему в полном объеме лабораторные работы в течение семестра, чей уровень знаний, умений и навыков соответствует уровню оценок «отлично», «хорошо» или «удовлетворительно» (п.7.2). Оценка «не зачтено» проставляется студенту, не выполнившему и (или) не защитившему в полном объеме практические задания и лабораторные работы в течение семестра, либо чей уровень знаний, умений и навыков соответствует уровню оценки «неудовлетворительно».

8. Учебно-методическое и информационное обеспечение дисциплины

Электронный каталог и электронные информационные ресурсы, предоставляемые научной библиотекой ФГБОУ ВО «Чувашский государственный университет имени И.Н. Ульянова» доступны по ссылке <http://library.chuvsu.ru/>

8.1. Рекомендуемая основная литература (ежегодное обновление перечня и условия доступа представлены в Приложениях к рабочей программе)

№ п/п	Наименование
1.	Алексеев А. Г. Параллельное программирование: учебное пособие [для 4 курса направления "Информатика и вычислительная техника"] / Алексеев А. Г., Йовенко А. Р., [отв. ред. А. А. Павлов]; Чуваш. гос. ун-т им. И. Н. Ульянова - Чебоксары: Изд-во Чуваш. ун-та, 2015. - 195с.: табл.. ISBN 978-5-7677-2162-7.
2.	Федотов И.Е. Модели параллельного программирования [Электронный ресурс] / И.Е. Федотов. — Электрон. текстовые данные. — М. : СОЛОН-ПРЕСС, 2012. — 384 с. Режим доступа: http://www.iprbookshop.ru/20877.html
3.	Туральчук К.А. Параллельное программирование с помощью языка С# [Электронный ресурс] / К.А. Туральчук. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 189 с. Режим доступа: http://www.iprbookshop.ru/39560.html
4.	Абрамян М.Э. Практикум по параллельному программированию с использованием электронного задачника Programming Taskbook for MPI [Электронный ресурс] / М.Э. Абрамян. — Электрон. текстовые данные. — Ростов-на-Дону: Южный федеральный университет, 2010. — 172 с. Режим доступа: http://www.iprbookshop.ru/47085.html

8.2. Рекомендуемая дополнительная литература (ежегодное обновление и условия доступа перечня представлены в Приложениях к рабочей программе)

№ п/п	Наименование
1.	Левин М.П. Параллельное программирование с использованием OpenMP [Электронный ресурс] / М.П. Левин. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 133 с. Режим доступа: http://www.iprbookshop.ru/52216.html
2.	Гергель В.П. Теория и практика параллельных вычислений [Электронный ресурс] / В.П. Гергель. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 480 с. Режим доступа: http://www.iprbookshop.ru/57385.html
3.	Иванов И.П. Программные средства обработки результатов расчетов в инженерных пакетах Ansys CFX и Abaqus для высокопроизводительных вычислительных установок [Электронный ресурс] : учебное пособие / И.П. Иванов, А.М. Чеповский. — Электрон. текстовые данные. — М. : Московский государственный технический университет имени Н.Э. Баумана, 2009. — 192 с. Режим доступа: http://www.iprbookshop.ru/31179.html
4.	Берчун Ю.В. Язык описания электронной аппаратуры VHDL [Электронный ресурс] : учебное пособие / Ю.В. Берчун. — Электрон. текстовые данные. — М. : Московский государственный технический университет имени Н.Э. Баумана, 2010. — 64 с. Режим доступа: http://www.iprbookshop.ru/31335.html
5.	Савельев В.А. Распараллеливание программ [Электронный ресурс] : учебник / В.А. Савельев, Б.Я. Штейнберг. — Электрон. текстовые данные. — Ростов-на-Дону: Южный федеральный университет, 2008. — 192 с. Режим доступа: http://www.iprbookshop.ru/47109.html
6.	Intel Parallel Programming Professional (Introduction) [Электронный ресурс] / В.П. Гергель [и др.]. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 568 с. Режим доступа: http://www.iprbookshop.ru/52170.html

8.3. Программное обеспечение, профессиональные базы данных, информационно-справочные системы.

Программное обеспечение, профессиональные базы данных, информационно-справочные системы, предоставляемые управлением информатизации ФГБОУ ВО «Чувашский государственный университет имени И.Н. Ульянова» доступны по ссылке <http://ui.chuvsu.ru/>*

8.3.1. Программное обеспечение

№ п/п	Наименование	Условия доступа/скачивания
1.	MS Office/ LibreOffice	лицензия университета/ свободное лицензионное соглашение (https://ru.libreoffice.org/)
2.	MS Windows/Linux (Ubuntu)	лицензия университета/ свободное лицензионное соглашение (http://ubuntu.ru/)
3.	Visual Studio Community	http://www.visualstudio.com/ru/vs/community
4.	Комплект разработчика приложений на языке Java (JDK)	http://www.java.com
5.	Интегрированная среда разработки Eclipse	https://www.eclipse.org
6.	Кросс платформенная библиотека шаблонов Intel Threading Building Blocks	https://www.threadingbuildingblocks.org
7.	Кросс платформенная библиотека компьютерного зрения OpenCV	http://opencv.org
8.	Кросс платформенная библиотека MPICH	http://www.mpich.org
9.	Инструментарий разработчика CUDA	https://developer.nvidia.com/cuda-toolkit

	(CUDA Toolkit)	
10.	Комплект разработчика OpenCL приложений (Intel® SDK for OpenCL™ Applications)	https://software.intel.com/en-us/intel-openc1

8.3.2. Базы данных, информационно-справочные системы

№ п/п	Наименование программного обеспечения	Условия доступа/скачивания
1.	Гарант	из внутренней сети университета (договор)
2.	Консультант +	

8.3.3. Рекомендуемые интернет-ресурсы и открытые он-лайн курсы

№ п/п	Наименование интернет ресурса	Режим доступа
1.	Российская Государственная Библиотека	http://www.rsl.ru
2.	Государственная публичная научно-техническая библиотека России	http://www.gpntb.ru
3.	Фундаментальная библиотека Нижегородского государственного университета	http://www.unn.ru/library
4.	Научная библиотека Казанского государственного университета	http://lsl.ksu.ru
5.	Научная электронная библиотека	http://elibrary.ru
6.	Полнотекстовая библиотека учебных и учебно-методических материалов	http://window.edu.ru
7.	Электронно-библиотечная система IPRbooks	http://www.iprbookshop.ru

9. Материально-техническое обеспечение дисциплины

Учебные аудитории для лекционных занятий по дисциплине оснащены автоматизированным рабочим местом (АРМ) преподавателя, обеспечивающим тематические иллюстрации и демонстрации, соответствующие программе дисциплины в составе:

- ПЭВМ с доступом в интернет (операционная система, офисные программы, антивирусные программы);
- мультимедийный проектор с дистанционным управлением;
- мультимедийное звуковое оборудование;
- настенный экран;
- интерактивная доска SMART;
- телевизор SMART.

Учебные аудитории для практических, лабораторных и самостоятельных занятий по дисциплине оснащены АРМ преподавателя и пользовательскими АРМ по числу обучающихся, объединенных локальной сетью («компьютерный» класс), с возможностью подключения к сети Интернет и доступом к электронной информационно-образовательной среде ФГБОУ ВО «Чувашский государственный университет имени И.Н. Ульянова».

10. Средства адаптации преподавания дисциплины к потребностям лиц с ограниченными возможностями

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) могут предлагаться одни из следующих вариантов восприятия информации с учетом их индивидуальных

психофизических особенностей, в том числе с применением электронного обучения и дистанционных технологий:

- для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); в печатной форме на языке Брайля; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации.

- для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации.

- для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.

11. Методические рекомендации по освоению дисциплины

В ходе лекционных занятий студенту рекомендуется вести конспектирование учебного материала. Следует обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации, положительный опыт в ораторском искусстве. При составлении конспекта желательно оставлять в рабочих конспектах поля, на которых в дальнейшем можно делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. В ходе лекционных занятий рекомендуется задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

В ходе подготовки к практическим занятиям и лабораторным работам рекомендуется изучить основную литературу, ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях. Основой для выполнения лабораторной работы являются разработанные кафедрой методические указания. Рекомендуется дорабатывать свой конспект лекций, делая в нем соответствующие записи из литературы, рекомендованной преподавателем и предусмотренной рабочей программой дисциплины. В процессе подготовки студент может дополнить список использованной литературы современными источниками, не представленными в списке рекомендованной литературы, и в дальнейшем использовать собственные подготовленные учебные материалы при написании выпускной квалификационной работы.

Формы организации студентов на лабораторных работах: фронтальная и индивидуальная. При фронтальной форме организации занятий все студенты выполняют одновременно одну и ту же работу. При индивидуальной форме организации занятий каждый студент выполняет индивидуальное задание.

В результате выполнения лабораторной работы запланирована подготовка письменного отчета, то отчет о выполненной работе необходимо оформлять в соответствии с требованиями методических указаний. Качество выполнения лабораторных работ является важной составляющей оценки текущей успеваемости обучающегося.