

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных технологий и прикладной математики

УТВЕРЖДАЮ

Проректор по учебной работе,
качеству образования – первый
проректор

Хагуров Т.А.

подпись

«31» мая 2019 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)
Б1.В.ДВ.01.02 ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ

Направление подготовки/специальность 01.04.02 Прикладная математика и информатика

Направленность (профиль) / специализация Технологии программирования и разработки информационно-коммуникационных систем

Форма обучения очная

Квалификация магистр

Краснодар 2019

Рабочая программа дисциплины «Параллельное программирование» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки / специальности 01.04.02 Прикладная математика и информатика

Программу составил(и):

В.В. Подколзин, доцент, канд. физ.-мат. наук
И.О. Фамилия, должность, ученая степень, ученое звание


подпись

О.В. Гаркуша, доцент, канд. физ.-мат. наук, доцент
И.О. Фамилия, должность, ученая степень, ученое звание


подпись

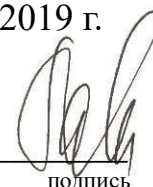
Рабочая программа дисциплины «Параллельное программирование» утверждена на заседании кафедры информационных технологий протокол № 15 от «07» мая 2019 г.



И. о. зав. кафедрой (разработчика) О.В. Гаркуша
фамилия, инициалы


подпись

Рабочая программа обсуждена на заседании кафедры утверждена на заседании кафедры информационных технологий протокол № 15 от «07» мая 2019 г.


подпись

И. о. зав. кафедрой (выпускающей) О.В. Гаркуша
фамилия, инициалы

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики протокол № 1 от «15» мая 2019г.

Председатель УМК факультета Коваленко А.В
фамилия, инициалы


подпись

Рецензенты:

Рубцов Сергей Евгеньевич, кандидат физико-математических наук, доцент кафедры математического моделирования ФГБГОУ «КубГУ»

Бегларян Маргарита Евгеньевна, кандидат физико-математических наук, доцент, заведующий кафедрой СГЕНД СКФ ФГБОУ ВО «Российский государственный университет правосудия»

1. Цели и задачи изучения дисциплины.

1.1 Цель освоения дисциплины.

Целью курса является изучение математических моделей, методов, современных технологий параллельного программирования, приобретение умений и навыков использования на практике средств разработки и сред выполнения параллельных программ для решения трудоемких вычислительных задач.

1.2 Задачи дисциплины.

Основными задачами курса является освоение следующих тем:

- Введение в параллельную обработку данных.
- Принципы построения параллельных вычислительных систем.
- Организация программ как системы процессов.
- Параллельное программирование для систем с общей памятью.
- Система MPI.
- Принципы разработки параллельных алгоритмов и программ.
- Модели функционирования параллельных программ.
- Параллельные алгоритмы решения типовых задач.

1.3 Место дисциплины в структуре образовательной программы.

Дисциплина относится к вариативной части Блока 1 "Дисциплины (модули)" учебного плана.

Дисциплина взаимодействует для формирования компетенций с дисциплинами Информатика, Основы программирования, Теория автоматов и формальных языков, Web-программирование, Основы разработки кросс-платформенных приложений.

Требованием к «входным» знаниям является понимание основ архитектуры ЭВМ и сетевых технологий, владение основами программирования на языках C/C++ и Java.

1.4 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы.

Перечень результатов образования, формируемых дисциплиной:

Знать:

- 1) аппаратные и программные аспекты реализации параллелизма, способы и средства разработки программ;
- 2) формальные модели параллельного программирования, применяемые для моделирования параллельных программ в современных операционных системах и оболочках;
- 3) принципы организации взаимодействия асинхронных процессов в современных операционных системах;.

Уметь:

- 4) использовать современные программные средства, операционные системы и оболочки, сервисные программы для распараллеливания алгоритмов
- 5) реализовывать параллельные программы на локальном компьютере и в сети рабочих станций с применением концептуальных функционального и объектно-ориентированного направлений программирования.

Владеть:

- 6) технологиями создания многопоточных приложений для систем с общей памятью с

- применением концептуальных положений функционального и объектно-ориентированного направлений программирования;
- 7) современными инструментальными средствами параллельных вычислительных и суперкомпьютерных систем.

Изучение данной учебной дисциплины направлено на формирование у обучающихся компетенций

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	Знать	Уметь	Владеть
1	ОПК-7	способностью использовать знания основных концептуальных положений функционального, логического, объектно-ориентированного и визуального направлений программирования, методов, способов и средств разработки программ в рамках этих направлений	1),	5)	6)
2	ПК-5	готовностью к использованию современных системных программных средств: операционных систем, операционных и сетевых оболочек, сервисных программ	2), 3)	4)	7)

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ.

Общая трудоёмкость дисциплины составляет 3 зач.ед. (108 часов), их распределение по видам работ представлено в таблице (для студентов ОФО).

Вид учебной работы		Всего часов	Семестры (часы)			
			8	—		
Контактная работа, в том числе:						
Аудиторные занятия (всего):						
Занятия лекционного типа		16	16	-	-	-
Лабораторные занятия		32	32	-	-	-
Занятия семинарского типа (семинары, практические занятия)		-	-	-	-	-
Иная контактная работа:						
Контроль самостоятельной работы (КСР)		2	2			
Промежуточная аттестация (ИКР)		0,3	0,3			
Самостоятельная работа, в том числе:						
Курсовая работа		-	-	-	-	-
Проработка учебного (теоретического) материала		4	4	-	-	-
Выполнение индивидуальных заданий (подготовка сообщений, презентаций)		16	16	-	-	-
Реферат		-	-	-	-	-
Подготовка к текущему контролю		2	2	-	-	-
Контроль:						
Подготовка к экзамену		35,7	35,7			
Общая трудоемкость	час.	108	108	-	-	-
	в том числе контактная	50,3	50,3			

	работа					
	зач. ед	3	3			

Процедура промежуточной аттестации проходит в форме экзамена.

2.2 Структура дисциплины:

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы дисциплины, изучаемые в 8 семестре (очная форма).

Вид промежуточной аттестации: экзамен.

№	Наименование разделов	Количество часов				
		Всего	Аудиторная работа		Внеаудиторная работа	
			Л	ЛР	СРС	контроль
1.	Принципы построения параллельных вычислительных систем	22	4	6	4	8
2.	Параллельное программирование в системах с общей памятью	26	4	8	6	8
3.	Параллельное программирование на основе MPI	26	4	8	6	8
4.	Введение в методы параллельного программирования	26	4	8	6	8
5.	Обзор изученного материала	5,7		2		3,7
6.	Контроль самостоятельной работы (КСР)	2				
7.	Промежуточная аттестация (ИКР)	0,3				
	Итого по дисциплине:	108	16	32	22	35,7

Примечание: Л – лекции, ПЗ – практические занятия / семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

2.3 Содержание разделов дисциплины:

2.3.1 Занятия лекционного типа.

№	Наименование раздела	Содержание раздела	Форма текущего контроля
1	Принципы построения параллельных вычислительных систем	Ограничения на частоту процессоров и многоядерность, сложные и простые ядра. Пути достижения параллелизма. Архитектура вычислительных систем статическая и динамическая конвейеризация. Векторные вычисления. Систематика Флинна. Примеры суперкомпьютеров и кластеров. Моделирование и анализ параллельных вычислений.	решение задач
2	Параллельное программирование в системах с общей памятью	Процессы и потоки. Потоки Windows, критические секции и семафоры. Потоки Posix, мьютексы. OpenMP.	решение задач

		Потоки Java, синхронизация. Транзакционная память Clojure.	
3	Параллельное программирование на основе MPI	Настройка MPICH2. Структура программы MPI. Функции MPI.	решение задач
4	Введение в методы параллельного программирования	Подсчет числа Pi. Умножение матрицы на вектор. Умножение матриц.	решение задач

№ лекции	Раздел	Тема лекции	Содержание лекции
1.	Принципы построения параллельных вычислительных систем	Архитектура параллельных вычислительных систем	Ограничения на частоту процессоров и многоядерность, сложные и простые ядра. Пути достижения параллелизма: - конвейеры, параллельная работа узлов процессора, прерывания, DMA; - ядра, потоки и процессы; - мультипроцессорность, общая память; - мультимикомпьютеры, разделяемая память. Систематика Флинна, детализация. Статическая и динамическая конвейеризация. Векторные вычисления. Вычисления на GPU. Примеры суперкомпьютеров и кластеров. Обсуждение задач параллельного программирования и параллельных вычислений.
2.	Принципы построения параллельных вычислительных систем	Моделирование и анализ параллельных вычислений	Моделирование и анализ параллельных вычислений: - граф операции-операнды - определение времени выполнения параллельного алгоритма - ускорение, эффективность и стоимость, примеры - закон Амдаля, закон Густавсона-Барсиса. Изоэффективность, пример расчета Пи.
3.	Параллельное программирование в системах с общей памятью	Процессы и потоки, Windows	Процессы и потоки. Состояние потока. Контекст процесса и потока. Дескрипторы. Потоки ядра и пользовательские потоки. Операции над процессами и потоками в Windows.
4.	Параллельное программирование	Процессы Unix и потоки Pthreads,	Операции над процессами и потоками в Unix. Планирование, алгоритмы планирования.

	ание в системах с общей памятью	планирование и синхронизация	Критерии оценки. Синхронизация. Решение задачи взаимного исключения. Критические секции и семафоры Windows. Мьютексы в Linux. Проблемы синхронизации: гонки, тупики, заброшенные замки.
5.	Параллельное программирование в системах с общей памятью	OpenMP	Модель, спецификация и примеры работы библиотеки OpenMP.
6.	Параллельное программирование на основе MPI	MPI	Модель, спецификация и примеры работы с MPI. Операции передачи данных.
7.	Параллельное программирование на основе MPI	MPI	Типы в MPI. Группы, коммутаторы и виртуальные топологии MPI.
8.	Введение в методы параллельного программирования		Умножение матрицы на вектор. Умножение матриц. Оптимизация однопоточного умножения: кеш процессора, memset. Ленточная схема, пример MPI. Умножение матриц: алгоритм Фокса, алгоритм Кэннона. Матричное представление графов. Задача поиска всех кратчайших путей.
9.	Введение в методы параллельного программирования		

2.3.2 Занятия семинарского типа.

Не предусмотрены

2.3.3 Лабораторные занятия.

№ занятия	Раздел	Тема лабораторного занятия	Содержание лабораторного занятия	Форма текущего контроля
1.	Параллельное программирование в системах с общей памятью	Потоки Windows.	Разбор примеров работы с потоками, критическими секциями и семафорами.	решение задач

2.	Параллельное программирование в системах с общей памятью	Потоки Windows задача 1 и 2.	Реализация умножения вектора на число. Реализация скалярного умножения векторов.	решение задач
3.	Параллельное программирование в системах с общей памятью	Потоки Windows задача 3 и 4.	Реализация умножения матрицы на вектор. Реализация умножения матриц.	решение задач
4.	Параллельное программирование в системах с общей памятью	Потоки Pthreads в Linux.	Разбор примеров работы с потоками и мьютексами.	решение задач
5.	Параллельное программирование в системах с общей памятью	Потоки Pthreads задача 1 и 2.	Реализация умножения вектора на число. Реализация скалярного умножения векторов.	решение задач
6.	Параллельное программирование в системах с общей памятью	Потоки Pthreads задача 3 и 4.	Реализация умножения матрицы на вектор. Реализация умножения матриц.	решение задач
7.	Параллельное программирование в системах с общей памятью	Введение в OpenMP.	Разбор примеров.	решение задач
8.	Параллельное программирование в системах с общей памятью	OpenMP задача 1 и 2.	Реализация умножения вектора на число. Реализация скалярного умножения векторов.	решение задач
9.	Параллельное программирование в системах с общей памятью	OpenMP задача 3 и 4.	Реализация умножения матрицы на вектор. Реализация умножения матриц.	решение задач
10.	Параллельное программирование в системах с общей памятью	Потоки Java.	Разбор примеров.	решение задач
11.	Параллельное программирование в системах	Java задача 1 и 2.	Реализация умножения вектора на число. Реализация скалярного умножения	решение задач

	с общей памятью		векторов.	
12.	Параллельное программирование в системах с общей памятью	Java задача 3 и 4.	Реализация умножения матрицы на вектор. Реализация умножения матриц.	решение задач
13.	Параллельное программирование на основе MPI	Введение в MPI.	Настройка MPICH2. Разбор примеров.	решение задач
14.	Параллельное программирование на основе MPI	MPI задача 1 и 2.	Реализация умножения вектора на число. Реализация скалярного умножения векторов.	решение задач
15.	Параллельное программирование на основе MPI	MPI задача 3 и 4.	Реализация умножения матрицы на вектор. Реализация умножения матриц.	решение задач

2.3.4 Примерная тематика курсовых работ (проектов)

Курсовые работы - не предусмотрены

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	Принципы построения параллельных вычислительных систем	Программирование на JAVA [Текст] : учебное пособие / С. Г. Сеница, А. В. Уварова ; М-во образования и науки Рос. Федерации, Кубанский гос. ун-т. - Краснодар : [Кубанский государственный университет], 2016. - 117 с. : ил. - Библиогр.: с. 116. - ISBN 978-5-8209-1215-3 Стандарты оформления исходного кода программ и современные интегрированные среды разработки программного обеспечения: учеб.-метод.пособие. Ю.В. Кольцов, А.В.Уварова, С.Г.Сеница [и др.] – Краснодар: Кубанский гос.ун-т, 2017
2	Параллельное программирование в системах с общей памятью	Программирование на JAVA [Текст] : учебное пособие / С. Г. Сеница, А. В. Уварова ; М-во образования и науки Рос. Федерации, Кубанский гос. ун-т. - Краснодар : [Кубанский государственный университет], 2016. - 117 с. : ил. - Библиогр.: с. 116. - ISBN 978-5-8209-1215-3 Стандарты оформления исходного кода программ и современные интегрированные среды разработки программного обеспечения: учеб.-метод.пособие. Ю.В. Кольцов, А.В.Уварова, С.Г.Сеница [и др.] – Краснодар: Кубанский гос.ун-т, 2017
3	Параллельное	Программирование на JAVA [Текст] : учебное пособие / С. Г.

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
	программирование на основе MPI	Синица, А. В. Уварова ; М-во образования и науки Рос. Федерации, Кубанский гос. ун-т. - Краснодар : [Кубанский государственный университет], 2016. - 117 с. : ил. - Библиогр.: с. 116. - ISBN 978-5-8209-1215-3 Стандарты оформления исходного кода программ и современные интегрированные среды разработки программного обеспечения: учеб.-метод.пособие. Ю.В. Кольцов, А.В.Уварова, С.Г.Синица [и др.] – Краснодар: Кубанский гос.ун-т, 2017
4	Введение в методы параллельного программирования	Программирование на JAVA [Текст] : учебное пособие / С. Г. Синица, А. В. Уварова ; М-во образования и науки Рос. Федерации, Кубанский гос. ун-т. - Краснодар : [Кубанский государственный университет], 2016. - 117 с. : ил. - Библиогр.: с. 116. - ISBN 978-5-8209-1215-3 Стандарты оформления исходного кода программ и современные интегрированные среды разработки программного обеспечения: учеб.-метод.пособие. Ю.В. Кольцов, А.В.Уварова, С.Г.Синица [и др.] – Краснодар: Кубанский гос.ун-т, 2017

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа,

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии

В соответствии с требованиями ФГОС программа дисциплины предусматривает использование в учебном процессе следующих образовательные технологии: чтение лекций с использованием мультимедийных технологий; метод малых групп, разбор практических задач и кейсов.

При обучении используются следующие образовательные технологии:

– Технология коммуникативного обучения – направлена на формирование коммуникативной компетентности студентов, которая является базовой, необходимой для адаптации к современным условиям межкультурной коммуникации.

– Технология разноуровневого (дифференцированного) обучения – предполагает осуществление познавательной деятельности студентов с учётом их индивидуальных способностей, возможностей и интересов, поощряя их реализовывать свой творческий потенциал. Создание и использование диагностических тестов является неотъемлемой частью данной технологии.

– Технология модульного обучения – предусматривает деление содержания дисциплины на достаточно автономные разделы (модули), интегрированные в общий курс.

– Информационно-коммуникационные технологии (ИКТ) - расширяют рамки образовательного процесса, повышая его практическую направленность, способствуют интенсификации самостоятельной работы учащихся и повышению познавательной активности. В рамках ИКТ выделяются 2 вида технологий:

– Технология использования компьютерных программ – позволяет эффективно дополнить процесс обучения языку на всех уровнях.

– Интернет-технологии – предоставляют широкие возможности для поиска информации, разработки научных проектов, ведения научных исследований.

– Технология индивидуализации обучения – помогает реализовывать личностно-ориентированный подход, учитывая индивидуальные особенности и потребности учащихся.

– Проектная технология – ориентирована на моделирование социального взаимодействия учащихся с целью решения задачи, которая определяется в рамках профессиональной подготовки, выделяя ту или иную предметную область.

– Технология обучения в сотрудничестве – реализует идею взаимного обучения, осуществляя как индивидуальную, так и коллективную ответственность за решение учебных задач.

– Игровая технология – позволяет развивать навыки рассмотрения ряда возможных способов решения проблем, активизируя мышление студентов и раскрывая личностный потенциал каждого учащегося.

– Технология развития критического мышления – способствует формированию разносторонней личности, способной критически относиться к информации, умению отбирать информацию для решения поставленной задачи.

Комплексное использование в учебном процессе всех вышеперечисленных технологий стимулируют личностную, интеллектуальную активность, развивают познавательные процессы, способствуют формированию компетенций, которыми должен обладать будущий специалист.

Основные виды интерактивных образовательных технологий включают в себя:

– работа в малых группах (команде) - совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности;

– проектная технология - индивидуальная или коллективная деятельность по отбору, распределению и систематизации материала по определенной теме, в результате которой составляется проект;

– анализ конкретных ситуаций - анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений;

– развитие критического мышления – образовательная деятельность, направленная на развитие у студентов разумного, рефлексивного мышления, способного выдвинуть новые идеи и увидеть новые возможности.

Подход разбора конкретных задач и ситуаций широко используется как преподавателем, так и студентами во время лекций, лабораторных занятий и анализа результатов самостоятельной работы. Это обусловлено тем, что при исследовании и решении каждой конкретной задачи имеется, как правило, несколько методов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

Семестр	Вид занятия	Используемые интерактивные образовательные технологии	количество интерактивных часов
---------	-------------	---	--------------------------------

Семестр	Вид занятия	Используемые интерактивные образовательные технологии	количество интерактивных часов
8	Л, ЛР	Занятия в режимах взаимодействия «преподаватель – студент» и «студент – студент»	12
Итого			12

Темы, задания и вопросы для самостоятельной работы призваны сформировать навыки поиска информации, умения самостоятельно расширять и углублять знания, полученные в ходе лекционных и практических занятий.

Подход разбора конкретных ситуаций широко используется как преподавателем, так и студентами при проведении анализа результатов самостоятельной работы.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

4.1 Фонд оценочных средств для проведения текущего контроля.

1. Задачи

В ходе выполнения лабораторных работ студентам предлагается с помощью пяти технологий параллельного программирования выполнить четыре задания: умножение вектора на число, скалярное произведение векторов, умножение матрицы на число, произведение матриц.

Технологии: потоки Windows, потоки Pthreads, потоки Java, OpenMP, MPI. Результаты времени выполнения на одном и том же объеме вычислений необходимо замерять на одном, двух и четырех потоках, записать в таблицу и проанализировать.

2. Проекты.

Студентам предлагается выполнить проекты в командах 2-3 человека, подготовить и защитить доклад по теме проекта, подготовить тезисы для участия в студенческой конференции ФКТиПМ или статью для публикации в журнале, опубликовать исходный код на GitHub.

Темы проектов:

1. MMX/SSE

С помощью векторных инструкций процессора MMX или SSE выполнить одну из операций:

- скалярное произведение векторов;
- умножение матриц;
- нахождение минимума;

Протестировать на большом объеме данных и сравнить с однопоточной реализацией.

2. OpenCL

С помощью OpenCL выполнить одну из операций:

- скалярное произведение векторов;
- умножение матриц;
- нахождение минимума;

Протестировать на большом объеме данных и сравнить с однопоточной реализацией.

3. Nvidia Cuda

С помощью Nvidia Cuda выполнить одну из операций:

- скалярное произведение векторов;
- умножение матриц;
- нахождение минимума;

Протестировать на большом объеме данных и сравнить с однопоточной реализацией.

4. WebCL

С помощью WebCL выполнить одну из операций:

- скалярное произведение векторов;
- умножение матриц;
- нахождение минимума;

Протестировать на большом объеме данных и сравнить с однопоточной реализацией.

5. Web Workers

С помощью Web Workers выполнить одну из операций:

- скалярное произведение векторов;
- умножение матриц;
- нахождение минимума;

Протестировать на большом объеме данных и сравнить с однопоточной реализацией.

6. Cilk

С помощью Cilk выполнить одну из операций:

- скалярное произведение векторов;
- умножение матриц;
- нахождение минимума;

Протестировать на большом объеме данных и сравнить с однопоточной реализацией.

7. TBV

С помощью TBV выполнить одну из операций:

- скалярное произведение векторов;
- умножение матриц;
- нахождение минимума;

Протестировать на большом объеме данных и сравнить с однопоточной реализацией.

8*. Написать программу, переносящую запущенный процесс с одной виртуальной машины под управлением ОС Linux на другую.

9*. Реализовать пример параллельной обработки данных на платформе Android.

- 10*. Реализовать пример параллельной обработки данных средствами Trix DSP или STM32F4 DSP.
- 11*. Реализовать пользовательские потоки для ОС Linux или Windows.
- 12*. Реализовать многопоточность на микроконтроллере STM32 (ядро Arm Cortex).
- 13*. Задача в каталоге 1.NStars – поиск оптимального маршрута и параметров миссии по исследованию звезд и поиску внеземной жизни.
- 14*. Реализовать программу поиска оптимальной разрезки заданного прямоугольника (лента) или множества заданных прямоугольников (листы) на множество заданных выпуклых многоугольников. При разрезке все заданные выпуклые многоугольники не должны пересекаться. Более оптимальной считается та разрезка, при которой высота использованного прямоугольника меньше (лента) или количество использованных прямоугольников меньше (листы).
- 15*. Задача в каталоге 2.MedClassify – классификация медицинских услуг.
- 16*. Реализовать параллельное обучение и распознавание изображений сверточными нейронными сетями по базеImageNet.
- 17*. Реализовать прозрачно параллельное вычисление findall, label или истинности предиката в SWI Prolog с использованием потоков на CPU или GPU.
- 18*. Реализовать библиотеку для прозрачных параллельных вычислений на GPU в Clojure.
- 19*. Написать приложение, определяющее название компании или имя физ. лица по почтовому адресу.
- 20*. Написать веб-сервис, определяющий пол владельца аккаунта в инстаграмме.
- 21*. Написать приложение для автоматического управления кампаниями в Яндекс Директ для интернет-магазинов.
- 22*. Реализовать обучение и распознавание голосовых команд для мобильных роботов.
- 23*. Написать приложение для поиска схожих по параметрам аккаунтов в соц. сетях с заданным множеством аккаунтов.
- 24*. Реализовать локализацию и определение размеров ближайшего объекта по изображению с двух камер и данным датчиков расстояния в реальном времени.
- 25*. Реализовать чтение дорожной разметки в реальном времени для мобильного робота.
- 26*. Реализовать машинное обучение оптимальной походки на местности для многоагентной системы хексаподов.
- 27*. Реализовать машинное обучение оптимальных параметров езды по линии для многоагентной системы колесных роботов.

28*. Реализовать систему распознавания жестов по данным от перчатки с датчиками изгиба на пальцах.

29*. Реализовать систему распознавания жестов по данным от датчиков смартфона.

30*. Реализовать систему составления оптимальных расписаний для факультета.

31*. Реализовать подсистему планирования отпусков для CRM.

32*. Реализовать систему планирования заказов поставщикам для МойСклад, Класс365 или 1С: Предприятие.

33*. Реализовать сервер для упрощенной версии протокола ModBus по TCP/IP (бинарный и ASCII вариант), UDP, веб-сокеты и клиентов на C/C++, Java, Android и JavaScript для одновременной записи и чтения данных по разным протоколам.

4.2 Фонд оценочных средств для проведения промежуточной аттестации.

Зачетно-экзаменационные материалы для промежуточной аттестации (экзамен)

Перечень компетенций, проверяемых оценочным средством:

ПК-5 готовностью к использованию современных систем программных средств; операционных систем и оболочек, сервисных программ.

ОПК-7. способностью использовать знания основных концептуальных положений функционального, логического, объектно-ориентированного и визуального направлений программирования, методов, способов и средств разработки программ в рамках этих направлений

Вопросы для промежуточной аттестации по итогам освоения дисциплины:

1. Уровни распараллеливания вычислений: независимость узлов, конвейеризация, параллельные и распределенные вычисления. Определение понятий: параллельные вычисления, суперкомпьютер, кластер.
2. Необходимость высокопроизводительных вычислений. Причины ограничения роста тактовой частоты процессоров. Классификация Флинна (SISD, SIMD, MISD, MIMD), понятие мультипроцессоров и мультикомпьютеров.
3. CISC- и RISC-процессоры. Основные черты RISC-архитектуры.
4. Повышение производительности процессоров за счет конвейеризации. Условия оптимального функционирования конвейера.
5. Суперконвейерные и суперскалярные процессоры. Выделение независимо работающих устройств: IU, FPU, MMU, BPU.
6. Повышение производительности процессоров за счет введения кэш-памяти. Кэши: единый, Гарвардский, с прямой записью, с обратной записью.
7. Согласование кэшей в мультипроцессорных системах с общей памятью.
8. Программа, процессор, процесс. Основные составляющие процесса, состояния процесса.
9. Виды ресурсов: аппаратные, программные, активные, пассивные, локальные, разделяемые, постоянные, временные, некритичные,

критичные.

10. Типы взаимодействия процессов: сотрудничающие и конкурирующие процессы. Критические секции, взаимное исключение процессов (задач). Проблемы, возникающие при синхронизации задач.
11. Механизмы взаимодействия процессов. Разделяемая память.
12. Механизмы взаимодействия процессов. Семафоры.
13. Механизмы взаимодействия процессов. Сигналы, события.
14. Потоки (нити, threads). Сравнение с процессами. Ресурсы, приоритеты.
15. Алгоритм Деккера.
16. Создание, уничтожение и ожидание завершения работы потоков Windows.
17. Синхронизация потоков. Критические секции Windows.
18. Создание, уничтожение и ожидание завершения работы потоков POSIX Threads. Барьеры в POSIX Threads.
19. Синхронизация потоков. Мьютексы: виды и операции над ними. Мьютексы в POSIX Threads.
20. Система MPI. Понятие параллельной программы MPI. Общая характеристика. Создание и завершение MPI-программы. Получение ранга процесса и количества процессов. Структура MPI-программы. Пример.
21. Точечные обмены данными между процессами MPI-программы. Базовые типы MPI. Пример. Одновременные прием и передача в MPI. Буферизованный, блокирующий, неблокирующий методы передачи данных.
22. Коллективные взаимодействия процессов MPI. Передача данных от одного процесса всем процессам. Сбор данных на одном процессе со всех процессов.
23. Коллективные взаимодействия процессов MPI. Обобщенная передача данных от одного процесса всем процессам. Обобщенный сбор данных на одном процессе со всех процессов.
24. Коллективные взаимодействия процессов MPI. Синхронизация вычислений. Аварийное завершение.
25. Создание групп в MPI и управление ими.
26. Управление коммуникаторами в MPI. Создание и уничтожение коммуникатора.
27. Производные типы в MPI. Карта типа. Характеристики типа.
28. Производные типы в MPI. Непрерывный способ создания. Регистрация и освобождение типа.
29. Производные типы в MPI. Векторный способ создания. Регистрация и освобождение типа.
30. Производные типы в MPI. Индексный способ создания. Регистрация и освобождение типа.
31. Производные типы в MPI. Структурный способ создания. Регистрация и освобождение типа.
32. Виртуальные топологии в MPI: декартова топология.
33. Виртуальные топологии в MPI: топология графа.
34. Технология OpenMP: общая характеристика, модель программирования, модель данных, последовательные и параллельные области, замер времени.
35. Технология OpenMP: мастер-поток, получение ранга, количества потоков, количества процессоров, задание количества порождаемых потоков, директивы master и single
36. Технология OpenMP: параллельные циклы, сокращенная запись,

редукция.

37.Технология OpenMP: синхронизация потоков, барьеры, критические секции, атомарные операции, замки.

38.Характеристики типовых схем коммуникации в многопроцессорных вычислительных системах. Примеры топологий сети передачи данных. Топология сети вычислительных кластеров. Характеристики топологий.

39.Оценка трудоемкости основных операций передачи данных. Передача данных между двумя процессорами сети. Передача данных от одного процессора всем процессорам сети.

40.Оценка трудоемкости основных операций передачи данных. Передача данных от всех процессоров всем процессорам сети.

41.Оценка трудоемкости операций передачи данных для кластерных систем.

Модель Хокни.

42.Граф «операции и операнды» и возможности распараллеливания вычислений, минимально возможное время выполнения параллельного алгоритма. Пример.

43. Потоки и примитивы синхронизации Java.

44. Параллельное программирование и транзакционная память в Clojure.

ФОС промежуточной аттестации состоит из вопросов к экзамену и задач текущего контроля по дисциплине.

Экзамен по дисциплине преследует цель оценить работу студента за курс, получение теоретических знаний, их прочность, развитие творческого мышления, приобретение навыков самостоятельной работы, умение применять полученные знания для решения практических задач.

Форма проведения экзамена: письменно, письменно.

Экзаменатору предоставляется право задавать студентам дополнительные вопросы по всей учебной программе дисциплины.

Результат сдачи экзамена заноситься преподавателем в экзаменационную ведомость и зачетную книжку.

Оценивание уровня освоения дисциплины основывается на качестве выполнения студентом задач за семестр, индивидуального задания / проекта и ответов на вопросы экзамена. За ответы на вопросы экзамена дается до 20 баллов.

0 баллов: непонимание сущности излагаемых вопросов, грубые ошибки в ответе, неуверенные и неточные ответы на дополнительные вопросы экзаменаторов.

1-10 баллов: знание и понимание основных вопросов программы, студент указал направление решения индивидуальной задачи; частично ответил на два вопроса билета или достаточно полно ответил хотя бы на один вопрос.

11-15 баллов: твёрдые и достаточно полные знания всего программного материала, последовательные, правильные, конкретные ответы на поставленные вопросы при свободном реагировании на замечания по отдельным вопросам; достаточно полно ответил на два вопроса.

16-20 баллов: глубокие исчерпывающие знания всего программного материала, логически последовательные, полные, грамматически правильные и конкретные ответы на вопросы экзаменационного билета и дополнительные вопросы.

Критерии оценки:

Оценка		
Удовлетворительно	Хорошо	Отлично
от 70 до 79 баллов	от 80 до 89 баллов	90 или более баллов

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

- при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

- при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

- при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины.

5.1 Основная литература:

1. Программирование на JAVA [Текст] : учебное пособие / С. Г. Сеница, А. В. Уварова ; М-во образования и науки Рос. Федерации, Кубанский гос. ун-т. - Краснодар : [Кубанский государственный университет], 2016. - 117 с. : ил. - Библиогр.: с. 116. - ISBN 978-5-8209-1215-3
2. Туральчук, К.А. Параллельное программирование с помощью языка C# / К.А. Туральчук. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 190 с. : ил. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429098>
3. Савельев В. А. , Штейнберг Б. Я. Распараллеливание программ: учебник. Издательство Южного федерального университета, 2008. 192 стр. ISBN: 978-5-9275-0547-0
http://biblioclub.ru/index.php?page=book_view_red&book_id=240965
4. Основы высокопроизводительных вычислений : учебное пособие / К.Е. Афанасьев, С.В. Стуколов, В.В. Малышенко и др. - Кемерово : Кемеровский государственный университет, 2012. - Т. 2. Технологии параллельного программирования. - 412 с. - ISBN 978-5-8353-1246-7 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=232204>

Для освоения дисциплины инвалидами и лицами с ограниченными возможностями здоровья имеются издания в электронном виде.

5.2 Дополнительная литература:

1. Алексеев, А.А. Основы параллельного программирования с использованием Visual Studio 2010 / А.А. Алексеев. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 332 с. http://biblioclub.ru/index.php?page=book_red&id=428829&sr=1
2. Теория и практика параллельных вычислений : учебное пособие / Гергель, Виктор Павлович ; В. П. Гергель. - М. : БИНОМ. Лаборатория знаний: Интернет-Университет Информационных Технологий, 2007. - 423 с.
3. Вычислительная математика и структура алгоритмов / В.В. Воеводин; Вл. В. Воеводин ; Моск. гос. ун-т им. М. В. Ломоносова. - 2-е изд., стер. - [М.] : Издательство Московского университета , 2010. - 166 с.

6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

- Учебный курс CS338. Многопроцессорные вычислительные системы и параллельное программирование
<http://www.hpcc.unn.ru/?doc=98>
- Свод знаний по программе сертификации ННГУ – Intel
<http://nncc.unn.ru/?id=1035>
- Интернет-Университет Суперкомпьютерных Технологий
<http://hpcu.ru/>
- Учебные курсы интернет-центра системы образовательных ресурсов в области СКТ
<http://hpc-education.ru/?q=node/19>

7. Методические указания для обучающихся по освоению дисциплины

При решении задач лабораторных работ студентам рекомендуется ознакомиться, скомпилировать и выполнить примеры кода, разработанные в первой учебной программе.

При решении задач с использованием OpenMP и MPI рекомендуется ознакомиться с материалами и примерами из курса CS338. Многопроцессорные вычислительные системы и параллельное программирование.

В качестве систем программирования для решения задач и изучения методов и алгоритмов, приведенных в лекциях, рекомендуется использовать на практических занятиях и при самостоятельной работе компиляторы Microsoft VisualC++ или GNUGCC. Для эффективного программирования рекомендуется использовать встроенный отладчик Visual C++ или, при использовании GCC, GDB. Также необходимы библиотеки POSIX Threads и MPI (например, MPICH2 или OpenMPI).

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

8.1 Перечень информационных технологий.

- Проверка домашних заданий и консультирование посредством электронной почты и социальной сети Вконтакте.
- Использование электронных презентаций при проведении практических занятий.

8.2 Перечень необходимого программного обеспечения.

1. Компиляторы языка C/C++: Microsoft Visual C++ или GNU GCC;
2. Библиотека POSIX Threads;
3. Библиотека MPI (например, MPICH2 или OpenMP);
4. Java, Clojure.
5. Программное обеспечение для безопасного отображения презентаций

8.3 Перечень информационных справочных систем:

1. Справочно-правовая система «Консультант Плюс» (<http://www.consultant.ru>)
2. Электронная библиотечная система eLIBRARY.RU (<http://www.elibrary.ru/>)

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

№	Вид работ	Материально-техническое обеспечение дисциплины и оснащенность
1.	Лекционные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
2.	Лабораторные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, проектором, программным обеспечением
3.	Текущий контроль, промежуточная аттестация	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
4.	Самостоятельная работа	Кабинет для самостоятельной работы, оснащенный компьютерной техникой с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета.