

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет математики и компьютерных наук

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Технологии параллельных вычислений

Кафедра дискретной математики и информатики

Образовательная программа
01.03.02 Прикладная математика и информатика

Профиль подготовки
Математическое моделирование и вычислительная математика

Уровень высшего образования
бакалавриат

Форма обучения
очная

Статус дисциплины: вариативная по выбору

Махачкала, 2018

Рабочая программа дисциплины «Технологии параллельных вычислений» составлена в 2018 году в соответствии с требованиями ФГОС ВО по направлению подготовки 01.03.02 – Прикладная математика и информатика (уровень бакалавриата) от 12 марта 2015 г. №228.

Разработчик: д.ф.-м.н., Магомедов Абдулкарим Магомедович

Рабочая программа дисциплины одобрена:

на заседании кафедры дискретной математики и информатики от 27.04.2018, протокол № 8

Зав. кафедрой  Магомедов А.М.

на заседании Методической комиссии факультета математики и компьютерных наук от 27.06.2018, протокол №6;

Председатель  В.Д. Бейбалаев

Рабочая программа дисциплины согласована с учебно-методическим

управлением «28» июня 2018 г. 

(подпись)

Аннотация рабочей программы дисциплины

Дисциплина “Технологии параллельных вычислений” входит в вариативную по выбору часть образовательной программы бакалавриата по направлению 01.03.02 – Прикладная математика и информатика.

Дисциплина реализуется на факультете математики и компьютерных наук кафедрой дискретной математики и информатики.

Содержание дисциплины охватывает круг вопросов, связанных с современными знаниями в области параллельного и распределенного программирования.

Дисциплина нацелена на формирование следующих компетенций выпускника: общепрофессиональных – ОПК-3; профессиональных – ПК-1, ПК-7.

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: лекции, лабораторные занятия.

Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости в форме контрольной работы и промежуточный контроль в форме зачета.

Объем дисциплины 3 зачетных единиц, в том числе в академических часах по видам учебных занятий

Се- местр	Учебные занятия							Форма проме- жуточной атте- стации (зачет, дифференциро- ванный зачет, экзамен
	в том числе							
	Контактная работа обучающихся с преподавателем						СРС, в том числе экза- мен	
	Все го	из них						
Лек- ции		Лабора- торные занятия	Практи- ческие занятия	КСР	Кон- сульта- ции			
6	108	32	32				44	зачет

1.

Цели освоения дисциплины

Цель изучения дисциплины «Технологии параллельных вычислений» - приобретение студентами теоретических знаний и устойчивых навыков использования параллельного программирования. Основная задача дисциплины: подготовка студентов к работе по созданию и поддержке современных распределенных информационных систем.

2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина «Технологии параллельных вычислений» входит в вариативную часть образовательной программы бакалавриата по направлению 01.03.02 – Прикладная математика и информатика.

Дисциплина «Технологии параллельных вычислений» логически и содержательно взаимосвязана с такими дисциплинами, как «Основы программирования», «Языки программирования».

3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения).

Компетенции	Формулировка компетенции из ФГОС ВО	Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)
ОПК-3	способностью к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям	Знать: языки программирования Уметь: разрабатывать и оценивать алгоритмы Владеть: навыками работы в средах программирования
ПК-1	способностью собирать, обрабатывать и интерпретировать данные современных научных исследований, необходимые для формирова-	Знать: архитектуру современных высокопроизводительных вычислительных систем

	ния выводов по соответствующим научным исследованиям	Уметь: обеспечивать передачу информации между приложениями Владеть: навыками работы с современными вычислительными средствами
ПК-7	способностью к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения	Знать: современные программные средства и библиотеки параллельных вычислений – MPI, OpenMP, CUDA Уметь: свободно пользоваться основными системными средствами современных ОС Владеть: средствами параллельного и распределенного программирования

4. Объем, структура и содержание дисциплины.

4.1. Объем дисциплины составляет 3 зачетных единиц, 108 академических часов.

4.2. Структура дисциплины.

№ п/п	Разделы и темы дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоя- тельную работу сту- дентов и трудоем- кость (в часах)				Самостоятельная работа	Формы текущего кон- троля успеваемости (по неделям семест- ра) Форма промежуточ- ной аттестации (по семестрам)
				Лекции	Практические занятия	Лаборатор- ные занятия	Контроль самост. раб.		
	Модуль 1. Принципы построения параллельных вычислительных систем								
1	Введение в предмет. Основные понятия и определения.	6	1	2				4	
2	Принципы построе- ния параллельных вычислительных си- стем.	6	2-3	4		4		10	Прием лабораторных работ
3	Классификация па- раллельных компь- ютеров и систем.	6	4-5	4		4		4	Прием лабораторных работ
	Итого по модулю 1:			10		8		18	Контрольная работа

Модуль 2. Технология OpenMP									
1	Технология OpenMP	6	6-7	4		4		2	Прием лабораторных работ
2	Параллельные и последовательные области	6	8-9	4		4		2	Прием лабораторных работ
3	Распределение работы	6	10-11	4		4		2	Прием лабораторных работ
4	Синхронизация	6	12	2		2		2	Прием лабораторных работ
	<i>Итого по модулю 2:</i>			14		14		8	Контрольная работа
Модуль 3. Технология CUDA									
1	Введение в CUDA C	6	13	2		4		8	Прием лабораторных работ
2	Параллельное программирование на CUDA C	6	14-16	6		6		10	Прием лабораторных работ
	<i>Итого по модулю 3:</i>			8		10		18	Зачет
	ИТОГО:			32		32		44	

4.3. Содержание дисциплины, структурированное по темам (разделам).

4.3.1. Содержание лекционных занятий по дисциплине

Первый модуль. Принципы построения параллельных вычислительных систем

Тема 1. Введение в предмет. Основные понятия и определения.

Понятие параллельных вычислений. Необходимость параллельных вычислений. Сдерживающие факторы. Характеристика необходимых знаний и умений.

Тема 2. Принципы построения параллельных вычислительных систем.

Основные виды ЭВМ. Принципы фон-Неймана. Структура традиционных ЭВМ. Способы повышения производительности компьютеров.

Тема 3. Классификация параллельных компьютеров и систем

Классификация Флинна, Хокни, Шнайдера. Взаимосвязь классификаций. Параллельные компьютеры с общей и разделенной памятью. Параллельные компьютеры с сетевой структурой. Характеристика типовых схем коммуникации в многопроцессорных вычислительных системах.

Второй модуль. Технология OpenMP

Тема 1. Технология OpenMP

Компиляция программы. Модель параллельной программы. Директивы и функции. Выполнение программы. Замер времени.

Тема 2. Параллельные и последовательные области

Директива parallel. Переменные среды и вспомогательные функции. Директива single. Директива master.

Тема 3. Распределение работы

Параллельные циклы. Параллельные секции. Директива workshare. Задачи (tasks).

Тема 4. Синхронизация

Барьер. Директива ordered. Критические секции. Директива atomic.

Третий модуль. Технология CUDA

Тема 1. Введение в CUDA C

Развитие GPU-вычислений. Технология CUDA.

Тема 2. Параллельное программирование на CUDA C

Организация параллелизма в CUDA. Написание программ на CUDA C.

4.3.2. Содержание лабораторных занятий по дисциплине

Первый модуль. Принципы построения параллельных вычислительных систем

Тема 1. Принципы построения параллельных вычислительных систем.

Тема 2. Классификация параллельных компьютеров и систем

Второй модуль. Технология OpenMP

Тема 1. Технология OpenMP

Тема 2. Параллельные и последовательные области

Тема 3. Распределение работы

Тема 4. Синхронизация

Третий модуль. Технология CUDA

Тема 1. Введение в CUDA C

Тема 2. Параллельное программирование на CUDA C

5. Образовательные технологии

Процесс изложения учебного материала сопровождается презентациями и демонстрацией решения задач в интерактивном режиме с использованием мультимедийного проектора.

Систематически проводится сравнение реализации потоков и нитей средствами различных языков Delphi, C# и др. Задания, предлагаемые студентам по различным темам, индивидуализированы, рекомендации по их выполнению, а также по исправлению ошибок, даются как непосредственно на лабораторных занятиях, так и по электронной почте.

Часть рекомендаций и указаний к решению заданий, которая относится ко всей учебной группе, размещается на сайте кафедры. Предусмотрено ведение образовательного блога.

Общение студентов по электронной почте с преподавателями кафедры носит регулярный характер, практикуется (в эпизодической форме) и общение с выпускниками кафедры – представителями российских и зарубежных компаний.

6. Учебно-методическое обеспечение самостоятельной работы студентов.

6.1 Виды самостоятельной работы и её контроля

Для обеспечения самостоятельной работы используется разработанный на кафедре пакет заданий и методических указаний, изданы три учебных пособия с заданиями и программами.

6.1. Виды и порядок выполнения самостоятельной работы

1. Изучение презентационных материалов и рекомендованной литературы.
2. Выполнение индивидуального пакета заданий на составление программ.
3. Решение задач и упражнений, изложенных на лекции.
4. Подготовка к отчету по индивидуальному пакету заданий
5. Подготовка к экзамену

6.2. Порядок контроля: 1. Блиц-опрос на лабораторных занятиях, 2. Проверка выполнения пакета заданий и прием отчета по ним, 3. Текущий контроль за выполнением задач, сформулированных на лекции, 4. Коллоквиумы, 5. Экзамен.

№	Вид самостоятельной работы	Вид контроля	Учебно-методическое обеспечение
1.	Проработка лекционного материала	Контрольный фронтальный опрос	См. разделы 7.3, 8, 9 данного документа
2.	Изучение рекомендованной литературы и материалов соответствующих форумов интернет	Контрольный фронтальный опрос, прием и представление рефератов.	См. разделы 7.3, 8, 9 данного документа
3.	Подготовка к сдаче промежуточных форм контроля	Контрольные работы по каждому модулю.	См. разделы 7.3, 8, 9 данного документа

Текущий контроль включает систематический блиц-опрос и проверку выполнения компьютерных программ.

Промежуточный контроль проводится в виде отчета по пакетам заданий, предварительная проверка решений практикуется по файлам, отправленным по электронной почте.

Итоговый контроль проводится в виде письменной экзаменационной работы с обязательным устным собеседованием по результатам предварительной проверки письменной работы.

Критерии выставления оценок «отлично», «хорошо», «удовлетворительно» определяются не только степенью владения материалом и практическими навыками составления программ, но также (с учетом низкого уровня школьной подготовки в регионе) и динамикой знаний и навыков студента за оцениваемый период.

7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

7.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образова-

тельной программы.

Перечень компетенций с указанием этапов их формирования приведен в описании образовательной программы.

Компетенция	Знания, умения, навыки	Процедура освоения
ОПК-3	Знать: языки программирования Уметь: разрабатывать и оценивать алгоритмы Владеть: навыками работы в средах программирования	Устный опрос, письменный опрос
ПК-1	Знать: архитектуру современных высокопроизводительных вычислительных систем Уметь: обеспечивать передачу информации между приложениями Владеть: навыками работы с современными вычислительными средствами	Письменный опрос
ПК-7	Знать: современные программные средства и библиотеки параллельных вычислений – MPI, OpenMP, CUDA Уметь: свободно пользоваться основными системными средствами современных ОС Владеть: средствами параллельного и распределенного программирования	Устный опрос

7.2. Типовые контрольные задания

Задание 1.

Напишите программу, в которой создается k нитей, и каждая нить выводит на экран свой номер и общее количество нитей в параллельной области в формате:

I am <Номер нити> thread from <Количество нитей> threads!

Задание 2.

Изучите конструкции для управления работой с данными `shared` и `private`. Напишите программу, в которой создается k нитей, и каждая нить выводит на экран свой номер через переменную `rank` следующим образом:

```
rank = omp_get_thread_num();  
printf("I am %d thread.\n", rank);
```

Экспериментами определите, общей или частной должна быть переменная `rank`.

Задание 3.

Напишите программу, в которой две нити параллельно вычисляют сумму чисел от 1 до N . Распределите работу по нитям с помощью оператора `if` языка C. Для сложения результатов вычисления нитей воспользуйтесь OpenMP-параметром `reduction`.

Задание 4.

Изучите OpenMP-директиву параллельного выполнения цикла `for`. Напишите программу, в которой k нитей параллельно вычисляют сумму чисел от 1 до N . Распределите работу по нитям с помощью OpenMP-директивы `for`.

Задание 5.

Изучите параметр `schedule` директивы `for`. Модифицируйте программу «Сумма чисел» из задания 4 таким образом, чтобы дополнительно выводилось на экран сообщение о том, какая нить, какую итерацию цикла выполняет:

[<Номер нити>]: calculation of the iteration number <Номер итерации>.

Задайте $k = 4$, $N = 10$.

Задание 6.

Напишите OpenMP-программу, которая вычисляет число π с точностью до N знаков после запятой. Используйте следующую формулу:

$$\pi = \left(\frac{4}{1+x_0^2} + \frac{4}{1+x_1^2} + \dots + \frac{4}{1+x_{N-1}^2} \right) \times \frac{1}{N}, \text{ где } x_i = (i+0.5) \times \frac{1}{N}, i = \overline{0, N-1}$$

Распределите работу по нитям с помощью OpenMP-директивы `for`.

Задание 7.

Напишите OpenMP-программу, которая вычисляет произведение двух квадратных матриц $A \times B = C$ размера $n \times n$. Используйте следующую формулу:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \cdot & \cdot & \cdot & \dots & \cdot \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} \quad B = \begin{pmatrix} b_{11} & b_{12} & b_{13} & \dots & b_{1n} \\ b_{21} & b_{22} & b_{23} & \dots & b_{2n} \\ \cdot & \cdot & \cdot & \dots & \cdot \\ b_{n1} & b_{n2} & b_{n3} & \dots & b_{nn} \end{pmatrix}$$

$$c_{im} = \sum_{j=1}^n a_{ij} \cdot b_{jm}; i = 1, 2, \dots, n; m = 1, 2, \dots, n$$

$$C = \begin{pmatrix} \sum_{j=1}^n a_{1j} \cdot b_{j1} & \sum_{j=1}^n a_{1j} \cdot b_{j2} & \sum_{j=1}^n a_{1j} \cdot b_{j3} & \dots & \sum_{j=1}^n a_{1j} \cdot b_{jn} \\ \sum_{j=1}^n a_{2j} \cdot b_{j1} & \sum_{j=1}^n a_{2j} \cdot b_{j2} & \sum_{j=1}^n a_{2j} \cdot b_{j3} & \dots & \sum_{j=1}^n a_{2j} \cdot b_{jn} \\ \dots & \dots & \dots & \dots & \dots \\ \sum_{j=1}^n a_{nj} \cdot b_{j1} & \sum_{j=1}^n a_{nj} \cdot b_{j2} & \sum_{j=1}^n a_{nj} \cdot b_{j3} & \dots & \sum_{j=1}^n a_{nj} \cdot b_{jn} \end{pmatrix}$$

Задание 8.

Изучите OpenMP-директивы создания параллельных секций sections и section. Напишите программу, содержащую 3 параллельные секции, внутри каждой из которых должно выводиться сообщение:

[<Номер нити>]: same in section <Номер секции>

Вне секций внутри параллельной области должно выводиться следующее сообщение:

[<Номер нити>]: parallel region

Запустите приложение на 2-х, 3-х, 4-х нитях. Проследите, как нити распределяются по параллельным секциям.

Задание 9.

Перепишите программу, в которой параллельно вычисляется сумма чисел от 1 до N (см. задание 4), без использования параметра reduction. Вместо параметра reduction используйте директиву atomic.

Задание 10.

Перепишите параллельную программу вычисления числа π (см. задание 6) без использования параметра reduction. Вместо параметра reduction используйте директиву critical.

Темы для рефератов:

1. Сравнение архитектуры CPU и GPU
2. Эволюция GPU
3. Использование нескольких GPU
4. Современные направления развития параллельных вычислительных систем
5. Создание потоков и нитей в Delphi
6. Средства распараллеливания в современных языках программирования
7. Вытесняющая мультизадачность
8. Алгоритмы исключения тупиковых ситуаций
9. Мультизадачность в Windows 10.0
10. Отечественные супер-ЭВМ

7.3. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Общий результат выводится как интегральная оценка, складывающаяся из текущего контроля - 50% и промежуточного контроля - 50%.

Текущий контроль по дисциплине включает:

- посещение занятий - 30 баллов,
- выполнение лабораторных заданий – 20 баллов,
- выполнение домашних (аудиторных) контрольных работ - 50 баллов.

Промежуточный контроль по дисциплине включает:

- устный опрос - 30 баллов,
- письменная контрольная работа - 50 баллов,
- тестирование - 20 баллов.

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

а) основная литература:

1. Барский А.Б. Параллельные информационные технологии [Электронный ресурс] : учебное пособие / А.Б. Барский. — Электрон. текстовые данные. — Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. — 503 с. — 978-5-4487-0087-3. — Режим доступа: <http://www.iprbookshop.ru/67379.html>
2. 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. — 978-5-94774-857-4. — Режим доступа: <http://www.iprbookshop.ru/52216.html>
3. Косяков М.С. Введение в распределенные вычисления. – СПб: НИУ ИТМО, 2014. – 155 с.
4. Боресков А.В. и др. Параллельные вычисления на GPU. Архитектура и программная модель CUDA. Учеб. пособие. 2-е изд. М.: Издательство Московского университета, 2015. – 336с.

б) дополнительная литература:

1. Поток выполнения [Электронный ресурс] // URL = <https://ru.wikipedia.org/wiki/> / (дата обра-

щения: 21.03.2018)

2. Операционная платформа. Определения и классификация [Электронный ресурс] // URL = <http://gigabaza.ru/doc/33138.html> / время доступа: 21.04.2018.

3. Процессы и потоки in-depth. Обзор различных потоковых моделей [Электронный ресурс] // URL = <http://habrahabr.ru/post/40227/> время доступа: 22.03.2018.

4. Автоматическое распараллеливание программ для распределенных систем. Статическое построение расширенного графа управления [Электронный ресурс] // URL = <http://xreferat.com/33/4673-1/> время доступа: 20.02.2018.

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», полезных для освоения дисциплины

1. Поток выполнения [Электронный ресурс] // URL = <https://ru.wikipedia.org/wiki/> / время доступа: 21.05.2015

2. Операционная платформа. Определения и классификация [Электронный ресурс] // URL = <http://gigabaza.ru/doc/33138.html> / время доступа: 21.05.2015.

3. Процессы и потоки in-depth. Обзор различных потоковых моделей [Электронный ресурс] // URL = <http://habrahabr.ru/post/40227/> время доступа: 22.05.2015.

4. Автоматическое распараллеливание программ для распределенных систем. Статическое построение расширенного графа управления [Электронный ресурс] // URL = <http://xreferat.com/33/4673-1/> время доступа: 22.05.2015.

5. <http://www.intuit.ru> [Электронный ресурс]

6. <http://www.parallel.ru> Материалы на сайте Лаборатории параллельных информационных технологий МГУ.

10. Методические указания для обучающихся по освоению дисциплины.

1. При решении лабораторных заданий программистский подход непременно должен присутствовать (без него решение не будет полноценным), однако, он не должен заслонять сугубо математические (доказательство и др.) и алгоритмические (построение, оптимизация, верификация и др.) аспекты.

2. Необходимо обратить внимание на распознавание ситуаций, когда распараллеливание: а) допустимо, б) целесообразно, в) необходимо. Нельзя игнорировать «накладные расходы» ресурсов, выделяемых собственно распараллеливанию вычислений.

3. Важно различать архитектурные и теоретические проблемы распараллеливания.

4. При решении проблемы автоматического распараллеливания особое внимание следует уделить созданию внутреннего представления программы, органично соответствующего проблематике решаемой задачи.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем.

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используются следующее программное обеспечение: Microsoft Visual Studio Express, Microsoft Windows, Ubuntu Linux, Skype. Также студентам предоставляется доступ к россий-

ским и международным электронным библиотекам через компьютеры университета.

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.

Имеется необходимая литература в библиотеке, медиапроектор и компьютер для проведения лекций-презентаций.

Лабораторные занятия проводятся в компьютерных классах с необходимым программным обеспечением.