


Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

УТВЕРЖДЕНО

решением Учёного совета факультета математики,
информационных и авиационных технологий

от «16» июня 2020 г., протокол № 5/20

Председатель _____ / М.А. Волков
«16» июня 2020 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Дисциплина	Параллельное программирование
Факультет	Факультет математики, информационных и авиационных технологий
Кафедра	Телекоммуникационные технологии и сети
Курс	4

Направление (специальность) 09.03.02 Информационные системы и технологии
код направления (специальности), полное наименование

Направленность (профиль/специализация) Разработка информационных систем
полное наименование

Форма обучения очная, заочная
очная, заочная, очно-заочная

Дата введения в учебный процесс УлГУ: «1» сентября 2020 г.



Программа актуализирована на заседании кафедры: протокол № _____ от _____ 20____ г.


Программа актуализирована на заседании кафедры: протокол № _____ от _____ 20____ г.

Программа актуализирована на заседании кафедры: протокол № _____ от _____ 20____ г.

Сведения о разработчиках:

ФИО	Кафедра	Должность, ученая степень, звание
Булаев Алексей Александрович	ТТС	к.т.н., доцент

СОГЛАСОВАНО	СОГЛАСОВАНО
Заведующий кафедрой телекоммуникационных технологий и сетей, реализующей дисциплину	Заведующий выпускающей кафедрой телекоммуникационных технологий и сетей
( / Смагин А.А. / Подпись ФИО «16» июня 2020 г.	( / Смагин А.А. / Подпись ФИО «16» июня 2020 г.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ:

Цели освоения дисциплины: формирование общекультурных и профессиональных компетенций, необходимых для реализации информационно-аналитической и научно-исследовательской деятельности.

Задачи освоения дисциплины: приобретение в рамках освоения предусмотренного курсом занятий следующих знаний, умений и навыков, характеризующих определённый уровень сформированности целевых компетенций (см. подробнее п.3):

- 1) знать:
 - структуры программного обеспечения;
 - алгоритмы обработки числовой и текстовой информации;
 - средства реализации алгоритмов на языке программирования;
 - новейших направлений в области технологий программирования;
 - языки программирования и методы разработки алгоритмов
- 2) уметь:
 - составлять алгоритм решения задачи
 - правильно формулировать требования к параллельным и распределённым программам в соответствии с аппаратными ресурсами;
 - производить декомпозицию задачи;
 - реализовывать параллельные алгоритмы;
- 3) владеть:
 - навыками работы с различными информационно-коммуникационными технологиями;
 - навыками программирования в современных средах.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП:


Дисциплина «Параллельное программирование» относится к числу дисциплин блока Б1.В.ДВ.08, предназначенного для студентов, обучающихся по направлению: 09.03.02 Информационные системы и технологии.

Для успешного изучения дисциплины необходимы знания и умения, приобретённые в результате освоения курсов «Дискретная математика и математическая логика», «Информатика и программирование» и полностью или частично сформированные компетенции ПК-6.

Основные положения дисциплины используются в дальнейшем при изучении таких дисциплин как: «Базы данных», «Технологии программирования».

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Код и наименование реализуемой компетенции	Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций
ПК-6 Способен проводить рабочее проектирование информационных систем и	Знать: <ul style="list-style-type: none"> – структуры программного обеспечения; – алгоритмы обработки числовой и текстовой

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Код и наименование реализуемой компетенции	Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций
технологий	<p>информации;</p> <ul style="list-style-type: none"> – средства реализации алгоритмов на языке программирования; – новейших направлений в области технологий программирования; – языки программирования и методы разработки алгоритмов; <p>Уметь:</p> <ul style="list-style-type: none"> – составлять алгоритм решения задачи – правильно формулировать требования к параллельным и распределенным программам в соответствии с аппаратными ресурсами; – производить декомпозицию задачи; – реализовывать параллельные алгоритмы; <p>Владеть:</p> <ul style="list-style-type: none"> – навыками работы с различными информационно-коммуникационными технологиями; – навыками программирования в современных средах;


4. ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ

4.1. Объем дисциплины в зачётных единицах (всего) 4

4.2. Объем дисциплины по видам учебной работы (в часах)

Форма обучения: очная


Вид учебной работы	Количество часов (форма обучения очная)	
	Всего по плану	В т.ч. по семестрам
		7
1	2	3
Контактная работа обучающихся с преподавателем в соответствии с УП	54	54
Аудиторные занятия:	54	54
Лекции	18\18*	18\18*
Семинары и практические занятия	18\18*	18\18*
Лабораторные работы, практикумы	18\18*	18\18*
Самостоятельная работа	54	54

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Вид учебной работы	Количество часов (форма обучения очная)	
	Всего по плану	В т.ч. по семестрам
		7
1	2	3
Форма текущего контроля знаний и контроля самостоятельной работы: тестирование, контр. работа, коллоквиум, реферат и др. (не менее 2 видов)	Тест	Тест
Курсовая работа	-	-
Виды промежуточной аттестации (экзамен, зачет)	зачёт	зачёт
Всего часов по дисциплине	108	108

Форма обучения: заочная

Вид учебной работы	Количество часов (форма обучения заочная)	
	Всего по плану	В т.ч. по сессиям
		15
1	2	3
Контактная работа обучающихся с преподавателем в соответствии с УП	20	20
Аудиторные занятия:	20	20
Лекции	8\8*	8\8*
Семинары и практические занятия	6\6*	6\6*
Лабораторные работы, практикумы	6\6*	6\6*
Самостоятельная работа	84	84
Форма текущего контроля знаний и контроля самостоятельной работы: тестирование, контр. работа, коллоквиум, реферат и др. (не менее 2 видов)	Тест	Тест
Курсовая работа	-	-
Виды промежуточной аттестации (экзамен, зачет)	зачёт	зачёт (4)
Всего часов по дисциплине	108	108

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		


4.3. Содержание дисциплины (модуля.) Распределение часов по темам и видам учебной работы:

Форма обучения: очная

Название разделов и тем	Всего	Виды учебных занятий					Форма текущего контроля знаний
		Аудиторные занятия			Занятия в интерактивной форме	Самостоятельная работа	
		Лекции	Практические занятия, семинары	Лабораторные работы, практикумы			
1	2	3	4	5	6	7	
Понятие высокопроизводительных ПВС (ВПВС).	6	2	2	2	2	6	-
Параллельные алгоритмы.	6	2	2	2	2	6	-
Алгоритмизация параллельных вычислений.	6	2	2	2	2	6	-
Средства разработки параллельных вычислений с использованием потоков (нитей) кода.	6	2	2	2	2	6	-
Средства разработки распределенных вычислений.	6	2	2	2	2	6	-
Протокол и библиотеки MPI.	6	2	2	2	2	6	-
Средства создания и отладки параллельных задач.	6	2	2	2	2	6	-
Язык OCCAM, конструкции языка, понятие процессов, каналов.	6	2	2	2	2	6	-
Удаленные вызовы процедур.	6	2	2	2	2	6	-
Итого	108	18	18	18	18	54	-

Форма обучения: заочная

Название разделов и тем	Всего	Виды учебных занятий					Форма текущего контроля знаний
		Аудиторные занятия			Занятия в интер	Самостоятель	
		Лекции	Практические	Лабораторные			

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		


			занятия, семинары	работы, практику мы	актив ной форме	ная работа	
1	2	3	4	5	6	7	
Понятие высокопроизводительных ПВС (ВПВС).	11	1	-	-	1	10	-
Параллельные алгоритмы.	11	1	-	-	1	10	-
Алгоритмизация параллельных вычислений.	13	1	1	1	1	10	-
Средства разработки параллельных вычислений с использованием потоков (нитей) кода.	13	1	1	1	1	10	-
Средства разработки распределенных вычислений.	13	1	1	1	1	10	-
Протокол и библиотеки MPI.	13	1	1	1	1	10	-
Средства создания и отладки параллельных задач.	13	1	1	1	1	10	-
Язык OCCAM, конструкции языка, понятие процессов, каналов.	13	1	1	1	1	14	-
Итого	108	8	6	6	8	84	-

**В интерактивной форме проводятся все лабораторные работы. Тема и содержание занятия приведены в пункте «ЛАБОРАТОРНЫЕ РАБОТЫ (ЛАБОРАТОРНЫЙ ПРАКТИКУМ)». Столбец «Занятия в интерактивной форме» в подсчёте итогов не участвует, т.к. дублирует столбец «Лабораторная работа».*

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Тема 1. Понятие высокопроизводительных ПВС (ВПВС).

Задачи дисциплины. Классификация параллельных вычислений по Флинну и Кришнамерфи. Способы повышения производительности компьютера: конвейеризация, кэширование команд и данных, увеличение количества регистров, суперскалярность и микросуперскалярность процессоров. Модели многопроцессорных систем (переход от RAM к PRAM), локальная память, модули с шинной связью, разделяемая общая память. Топологии межмодульной связи: шина, двумерная сетка, гиперкуб, деревья (в т.ч. толстые деревья), коммутирующий граф. Архитектура суперкомпьютеров. Транспьютерные

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

сетевые архитектуры. Распределенные системы: кластеры и сети (динамические, статические). Метрические характеристики вычислительной сети: Диаметр, связность, ширина сечения, поток через сечение, стоимость. Оценки метрик для различных сетей. Логическая и физическая организации ПВС. Сериализуемость, требования когерентности, стоимость пересылки в параллельных системах. Маршрутизация.

Тема 2. Параллельные алгоритмы.

Отличия параллельных алгоритмов от последовательных алгоритмов. Постановка задачи распараллеливания существующих сериальных алгоритмов. Пример суммирования массива. Реализация параллельных алгоритмов с использованием процессов. Виды межпроцессной коммуникации (сигналы, семафоры, разделяемая память, каналы, сообщения). Оценка эффективности алгоритма с учетом времени доставки кода и данных вычислительному узлу. Законы Амдала. Целесообразность распараллеливания. Как анализировать зависимость данных и кода в последовательной программе. Порядок вычислений, граф зависимости. Модель «грубой силы». Примеры решений для PRAM, BSP, LogP. Преобразования параллелизации циклов. Доставка данных, отображение массива данных на распределенную систему, избыточность. Влияние аппаратной реализации PRAM и проблемы когерентности данных. Модели конкурентной и исключительной примитивных операций с памятью (EREW, CRCW, CREW, QRQW).

Тема 3. Алгоритмизация параллельных вычислений.

Принцип «Разделяй-и-властвуй». Виды декомпозиции задачи: по данным, по коду, по выполнению, комбинированные. Примеры различных декомпозиций. Сортировка, выпуклые оболочки, матричные операции: примеры декомпозиции рекурсивного вида. Матричные операции: пример декомпозиции по данным, одно-, двух- и трехмерное разбиения.


Тема 4. Средства разработки параллельных вычислений с использованием потоков (нитей) кода.

Процессы и потоки (нити) управления. Понятие процесса. Процессы с поддержкой многопоточности. Диспетчеризация процессов. Динамическая и статистическая связь между процессором и очередью. Ведущие и разделяемые планировщики процессов. Взаимодействие планировщика со структурами системных данных и примитивов ядра ОС. Состав и функции диспетчера и системных управляющих программ – программы инициализации, программы таймирования, программы управления восстановлениями. Краткосрочное планирование процессов и потоков в многопроцессорной системе. Составляющие ядра и пользователя в процессах в UNIX. Сигналы как простейшие средства коммуникации. Коммуникация и синхронизация процессов в централизованных архитектурах, основные понятия и определения. Задача передачи данных между процессами «читатель-писатель». Тупики. Модели для анализа свойств асинхронных процессов. Алгоритм Дейкстры для задачи об обедающих философах. Понятие тупиковой ситуации и способы борьбы с ней. API потоков выполнения фирмы Sun и определения POSIX. Синхронизация потоков выполнения: взаимоисключающие блокировки, условные переменные, барьеры, блокировки чтения-записи.

Тема 5. Средства разработки распределенных вычислений.

Средства разработки распределенных вычислений с использованием параллельной виртуальной машины PVM. Использование функций библиотеки PvmLib. Состав функций. Компиляция и запуск программ в среде PVM.

Тема 6. Протокол и библиотеки MPI.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Состав функций, типы данных, организация обмена данными. Пример использования библиотеки MPI. Среда выполнения LAM. Установка, использование (компиляция и запуск программ, конфигурация вычислительной системы).

Тема 7. Средства создания и отладки параллельных задач.

Трансляция, компоновка, загрузка задач. Языки параллельного программирования и среды программирования. Расширения стандартных языков программирования для создания параллельных программ. SILK, mPC. Дополнительные операторы, их синтаксис и семантика. Особенности реализаций и примеры программирования.

Тема 8. Язык OSSAM, конструкции языка, понятие процессов, каналов.

Происхождение языка и его перенос на нетранспьютерные системы. Примеры простых приложений, отображающих конвейерную обработку данных. Пример умножения матрицы на вектор. Процесс доставки данных в вычислительный узел. Процессы-заглушки.

Тема 9. Удаленные вызовы процедур.

Высокоуровневый и низкоуровневый интерфейс программирования. XDR-преобразования, аутентификация, широковещательный режим. Использование технологий DCOM и CORBA для создания распределенных приложений. Язык описания интерфейсов IDL.

6. ТЕМЫ ПРАКТИЧЕСКИХ И СЕМИНАРСКИХ ЗАНЯТИЙ

Тема 1. Понятие высокопроизводительных ПВС (ВПВС) (форма проведения – семинар)

Вопросы по темам раздела

1. Задачи дисциплины.
2. Классификация параллельных вычислений по Флинну и Кришнамерфи.
3. Способы повышения производительности компьютера: конвейеризация, кэширование команд и данных, увеличение количества регистров, суперскалярность и микросуперскалярность процессоров.
4. Модели многопроцессорных систем

Тема 2. Параллельные алгоритмы (форма проведения – семинар)

Вопросы по темам раздела


1. Отличия параллельных алгоритмов от последовательных алгоритмов.
2. Постановка задачи распараллеливания существующих сериальных алгоритмов.
3. Реализация параллельных алгоритмов с использованием процессов.
4. Виды межпроцессной коммуникации.

Тема 3. Алгоритмизация параллельных вычислений (форма проведения – семинар)

Вопросы по темам раздела

1. Принцип «Разделяй-и-властвуй».
2. Виды декомпозиции задачи: по данным, по коду, по выполнению, комбинированные.
3. Сортировка, выпуклые оболочки.

Тема 4. Средства разработки параллельных вычислений с использованием потоков (нитей) кода (форма проведения – семинар)

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Вопросы по темам раздела

1. Процессы и потоки (нити) управления.
2. Понятие процесса.
3. Процессы с поддержкой многопоточности.
4. Диспетчеризация процессов.
5. Динамическая и статистическая связь между процессором и очередью.

Тема 5. Средства разработки распределенных вычислений (форма проведения – семинар)

Вопросы по темам раздела

1. Средства разработки распределенных вычислений с использованием параллельной виртуальной машины PVM.
2. Использование функций библиотеки PvmLib.
3. Состав функций.
4. Компиляция и запуск программ в среде PVM

Тема 6. Протокол и библиотеки MPI (форма проведения – семинар)

Вопросы по темам раздела

1. Состав функций, типы данных, организация обмена данными.
2. Среда выполнения LAM.

Тема 7. Средства создания и отладки параллельных задач (форма проведения – семинар)

Вопросы по темам раздела

1. Трансляция, компоновка, загрузка задач.
2. Языки параллельного программирования и среды программирования.

Тема 8. Язык OSSAM, конструкции языка, понятие процессов, каналов (форма проведения – семинар)

Вопросы по темам раздела

1. Происхождение языка и его перенос на нетранспьютерные системы.
2. Процесс доставки данных в вычислительный узел.
3. Процессы-заглушки.

Тема 9. Удаленные вызовы процедур (форма проведения – семинар)

Вопросы по темам раздела

1. Высокоуровневый и низкоуровневый интерфейс программирования.
2. XDR-преобразования, аутентификация, широковежательный режим.
3. Использование технологий DCOM и CORBA для создания распределенных приложений.


7.ЛАБОРАТОРНЫЕ РАБОТЫ, ПРАКТИКУМЫ

Лабораторная работа № 1

Цель – дать представление о построении простых параллельных программ на языке параллельного программирования MPI; представление о параллельных программах, настраиваемых на размер вычислительной системы, как на параметр; закрепить практическое освоение функций парных взаимодействий между ветвями параллельной программы.

Задание 1. Программирование конвейера.

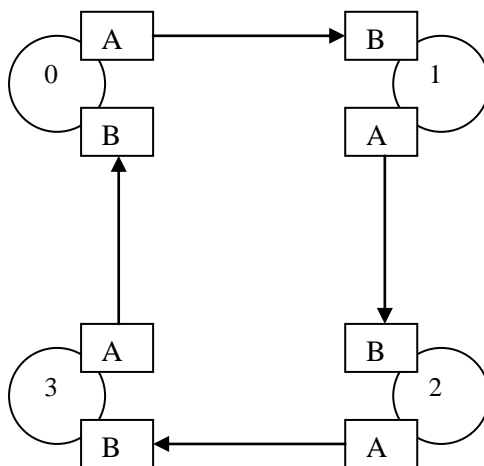
Ветвь 0 пересылает некоторые данные ветви 1, ветвь 1 принятые данные передает ветви 2, ветвь 2 принятые данные передает ветви 3 и т.д. по цепочке увеличения номеров. И,

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

наконец, ветвь 0 принимает пересылаемые данные от ветви size-1 и выводит на экран свой номер и принятые данные. Т.е. пересылка информации по кольцу компьютеров. В этом примере программу алгоритма нужно написать настраиваемой автоматически на размер вычислительной системы, т.е. алгоритм не должен зависеть от числа компьютеров и программа должна запускаться на любом допустимом количестве компьютеров.


Задание 2. Программирование кольцевых сдвигов данных

Все ветви одновременно пересылают некоторые свои данные ветвям с номерами на единицу большими, чем передающие ветви. Т.е. пересылка информации вдоль кольца компьютеров, см. рисунок ниже. В этом примере программу алгоритма нужно написать настраиваемой автоматически на размер вычислительной системы, т.е. алгоритм не должен зависеть от числа компьютеров и программа должна запускаться на любом допустимом количестве компьютеров.



Основные функции MPI

Система MPI является основным средством программирования таких современных высокопроизводительных мультимикомпьютеров, как Silicon Graphics Origin 2000, Cray T3D, Cray T3E, IBM SP2 и многих других. MPI работает на самых разных архитектурах мультимикомпьютеров, как с распределенной памятью, так и с разделяемой памятью. Кроме того, MPI работает на сетях компьютеров (кластерах) однородных и гетерогенных. Количество компьютеров в сети может быть от одного и более. Важнейшей особенностью MPI является то, что пользователь при написании своих параллельных программ не должен учитывать все эти архитектурные особенности конкретных мультимикомпьютеров,

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

поскольку MPI предоставляет пользователю виртуальный мультикомпьютер с распределенной памятью и с виртуальной сетью связи между виртуальными компьютерами. Пользователь заказывает количество компьютеров, необходимых для решения его задачи, и определяет топологию связей между этими компьютерами. MPI реализует этот заказ на конкретной физической системе. Ограничением является объем оперативной памяти физического мультикомпьютера. Таким образом пользователь работает в виртуальной среде, что обеспечивает переносимость его параллельных программ. Система MPI представляет собой библиотеку средств параллельного программирования для языков C и Fortran 77.

Методы распараллеливания и модели программ, поддерживаемые MPI

Одной из целей, преследуемых при решении задач на вычислительных системах, в том числе и на параллельных, – является эффективность. Эффективность параллельной программы существенно зависит от соотношения времени вычислений ко времени коммуникаций между компьютерами (при обмене данными). И чем меньше в процентном отношении доля времени, затраченного на коммуникации, в общем времени вычислений, тем больше эффективность. Для параллельных систем с передачей сообщений оптимальное соотношение между вычислениями и коммуникациями обеспечивают методы крупнозернистого распараллеливания, когда параллельные алгоритмы строятся из крупных и редко взаимодействующих блоков [2–8]. Задачи линейной алгебры, задачи, решаемые сеточными методами и многие другие, достаточно эффективно *распараллеливаются крупнозернистыми методами*.


MPMD - модель вычислений. MPI - программа представляет собой совокупность автономных процессов, функционирующих под управлением своих собственных программ и взаимодействующих посредством стандартного набора библиотечных процедур для передачи и приема сообщений. Таким образом, в самом общем случае MPI - программа реализует MPMD - модель программирования (Multiple program - Multiple Data).

SPMD - модель вычислений. Все процессы исполняют в общем случае различные ветви одной и той же программы. Такой подход обусловлен тем обстоятельством, что задача может быть достаточно естественным образом разбита на подзадачи, решаемые по одному алгоритму. На практике чаще всего встречается именно эта модель программирования (Single program - Multiple Data) [1,12].

Последнюю модель иначе можно назвать моделью *распараллеливания по данным*. Кратко, суть этого способа заключается в следующем. Исходные данные задачи распределяются по процессам (ветвям параллельного алгоритма), а алгоритм является одним и тем же во всех процессах, но действия этого алгоритма распределяются в соответствии с имеющимися в этих процессах *данными*. Распределение действий алгоритма заключается, например, в присвоении разных значений переменным одних и тех же циклов в разных ветвях, либо в исполнении в разных ветвях разного количества витков одних и тех же циклов и т.п. Другими словами, процесс в каждой ветви следует различными путями выполнения на той же самой программе.

Указанные модели, помимо поддержки на языковом уровне, поддерживаются архитектурами таких самых современных суперкомпьютеров как: ASCI RED (более 9000 Pentium PRO/200 объединены в единую систему, имеет быстродействие около 3,2 Tflops), и ASCI WAIT (8192 - , имеет быстродействие 12,2 Tflops), Cray T3D, Cray T3E, IBM SP2.

В первой части дано описание системы параллельного программирования MPI. Даны операторы компиляции и запуска C-программ, программные средства задания системных взаимодействий, виртуальные топологии. В каждом подразделе приводятся примеры

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

программ, закрепляющие понимание и усвоение материала. Эти же примеры могут использоваться как образцы для написания новых программ.

Во второй части даны четыре лабораторных работы, построенные как последовательность шагов по изучению программных средств параллельного программирования и освоению основных навыков написания параллельных программ.

Лабораторная работа № 2

Цель – дать представление о построении простых параллельных программ на языке параллельного программирования MPI; представление о параллельных программах, настраиваемых на размер вычислительной системы, как на параметр; закрепить практическое освоение функций парных взаимодействий между ветвями параллельной программы.

Задание 1. Проработка примеров из пункта "Примеры параллельных программ" этой же лабораторной.

Внимательно изучить примеры 2.1, 2.2, 2.3. Примеры 2.2 и 2.3 (Умножение матрицы на вектор на топологиях "кольцо" и "полный граф") откомпилировать и запустить на 4-х компьютерах.

Задание 2. Параллельное умножение матрицы на вектор на топологии "кольцо".

Разработать алгоритм написать и отладить параллельную программу умножения матрицы на вектор в топологии "кольцо": $A \times B = C$, при условии, что количество строк матрицы и элементов вектора нацело не делится на количество компьютеров. Например, матрица A размером $[22 \times 22]$ и вектор B размером $[22]$ на четырех компьютерах.

Задание 3. Параллельное умножение матрицы на матрицу на топологии "кольцо".

Вариант 1.


Разработать алгоритм написать и отладить параллельную программу умножения матрицы на матрицу в топологии "кольцо": $A \times B = C$, при условии, что количество строк матрицы A и столбцов матрицы B нацело делится на количество компьютеров. Например, матрица A размером $[20 \times 20]$ и матрица B размером $[20 \times 20]$ на четырех компьютерах.

Вариант 2.

Разработать алгоритм написать и отладить параллельную программу умножения матрицы на матрицу в топологии "кольцо": $A \times B = C$, при условии, что количество строк матрицы A и столбцов матрицы B нацело не делится на количество компьютеров. Например, матрица A размером $[22 \times 20]$ и матрица B размером $[20 \times 22]$ на четырех компьютерах.

Задание 4. Сравнительные временные характеристики двух алгоритмов умножения матрицы на вектор на топологиях "кольцо" и "полный граф".

В качестве параллельных программ взять примеры 2.2 и 2.3 (Умножение матрицы на вектор на топологиях "кольцо" и "полный граф"). Для обеих программ построить небольшие графики зависимости времени решения задачи от размеров матрицы и вектора. Обе программы запустить последовательно для матриц: $A = [300 \times 300]$, $[500 \times 500]$, $[700$

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

х 700], [1000 x 1000] и соответствующих этим матрицам векторов. Откомпилировать и запустить на 4-х компьютерах, засечь время и построить указанные графики.

Лабораторная работа № 3

Цель – дать представление о построении простых параллельных программ на языке параллельного программирования OpenMP; закрепить практическое освоение директив языка.

Задание 1. Проработка примеров из пункта "Примеры параллельных программ" этой же лабораторной.

Внимательно изучить примеры 3.1-3.5. Откомпилировать и запустить на 2-х процессах.

Задание 2. Параллельное умножение матрицы на вектор

Вариант 1.

Разработать алгоритм написать и отладить параллельную программу умножения матрицы на вектор с использованием распределения работ для параллельных процессов директивой **sections**. Использовать алгоритм примера 3.5.

Вариант 2.

Разработать алгоритм написать и отладить параллельную программу умножения матрицы на вектор с использованием распределения работ для параллельных процессов с непосредственным заданием работы («вручную»). Использовать алгоритм примера 3.5.

Задание 3. Параллельное умножение матрицы на матрицу.

Вариант 1.


Разработать алгоритм написать и отладить параллельную программу умножения матрицы на матрицу с использованием директивы распараллеливания цикла по виткам.

Вариант 2.

Разработать алгоритм написать и отладить параллельную программу умножения матрицы на матрицу с использованием распределения работ для параллельных процессов с непосредственным заданием работы («вручную»).

Задание 4. Сравнительные временные характеристики двух алгоритмов умножения матрицы на вектор на языке MPI и языке OpenMP.

В качестве параллельных программ взять параллельные программы: «умножение матрицы на вектор в MPI на топологии "кольцо"» и «умножение матрицы на вектор в OpenMP (один из вариантов)». Для обеих программ построить небольшие графики зависимости времени решения задачи от размеров матрицы и вектора. Обе программы запустить последовательно для матриц: $A = [300 \times 300]$, $[500 \times 500]$, $[700 \times 700]$, $[1000 \times 1000]$ и соответствующих этим матрицам векторов. Откомпилировать и запустить на 2-х компьютерах, засечь время и построить указанные графики.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Основные директивы OpenMP

Что такое OpenMP?

Интерфейс OpenMP задуман как стандарт для программирования на масштабируемых SMP-системах (SSMP, ccNUMA, etc.) в модели общей памяти (shared memory model). В стандарт OpenMP входят спецификации набора директив компилятора, процедур и переменных среды. Примерами систем с общей памятью, масштабируемых до большого числа процессоров, могут служить суперкомпьютеры Cray Origin2000 (до 128 процессоров), HP 9000 V-class (до 32 процессоров в одном узле, а в конфигурации из 4 узлов - до 128 процессоров), Sun Starfire (до 64 процессоров).

Где найти информацию?

Основной источник информации - сервер <http://www.openmp.org/>. На сервере доступны спецификации, статьи, учебные материалы, ссылки.

Базовый потоковый параллелизм

Общедоступный процесс памяти может состоять из множественных потоков, несколько нитей управления, которые имеют общее адресное пространство, но разные потоки команд и отдельные стеки. В простейшем случае, процесс состоит из одной нити. Нити иногда называют также потоками, легковесными процессами, LWP (light-weight processes).. OpenMP основан на существовании множественных потоков в общедоступной памяти, программирующей парадигму.

Явный Параллелизм

OpenMP имеет явную (не автоматический) модель программирования, предлагая программисту полное управление по распараллеливанию.

Модель Fork-Join (Ветвление – Объединение)


OpenMP использует Fork-Join модель параллельного выполнения:

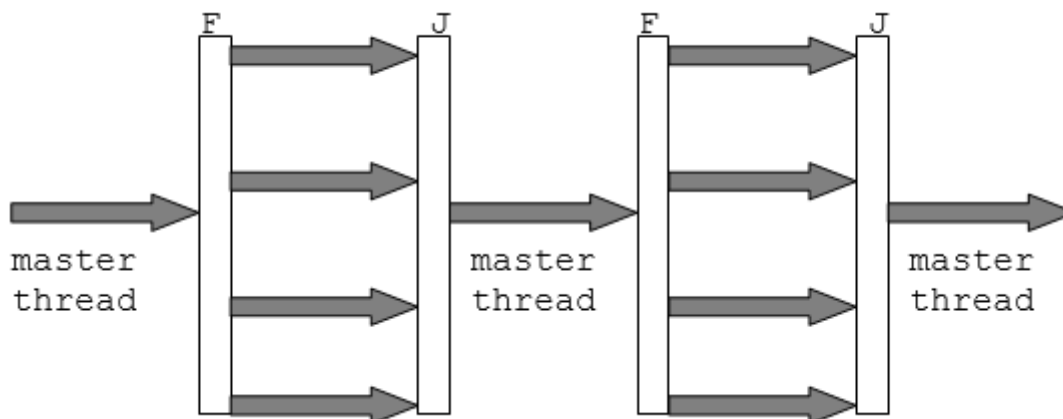
Все программы OpenMP начинаются как единственный процесс: главный поток. Главный поток выполняется последовательно, пока не столкнется с первой областью параллельной конструкции.

Fork (ВЕТВЛЕНИЕ): главный поток создает группу параллельных потоков.

Инструкции в программе, которые включены параллельной конструкцией области {*региона*}, тогда выполнены параллельно среди различных потоков группы

Join (ОБЪЕДИНЕНИЕ): Когда потоки группы завершают инструкции в области параллельной конструкции, они синхронизируются и закрываются, оставляя только главный поток.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		



2. Директивы

Директивы OpenMP с точки зрения языка C являются командами препроцессора:

`#pragma omp`

Фортрана являются комментариями и начинаются с комбинации символов:

`!$OMP.`

Формат директивы на C/C++:

<команда для препроцессора> <имя директивы> <предложение(klausu)>

`#pragma omp parallel [clause clause ...]`

{

.

.

.

}

Директивы можно разделить на 3 категории:

- определение параллельной секции,
- разделение работы,
- синхронизация.


Каждая директива может иметь несколько дополнительных атрибутов - клауз. Отдельно специфицируются клаузы для назначения классов переменных, которые могут быть атрибутами различных директив.

8. ТЕМАТИКА КУРСОВЫХ, КОНТРОЛЬНЫХ РАБОТ, РЕФЕРАТОВ

Не предусмотрены.

9. ПЕРЕЧЕНЬ ВОПРОСОВ К ЭКЗАМЕНУ (ЗАЧЕТУ)

1. Задачи дисциплины.


Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

2. Классификация параллельных вычислений по Флинну и Кришнамерфи.
3. Способы повышения производительности компьютера: конвейеризация, кэширование команд и данных, увеличение количества регистров, суперскалярность и микросуперскалярность процессоров.
4. Модели многопроцессорных систем
5. Отличия параллельных алгоритмов от последовательных алгоритмов.
6. Постановка задачи распараллеливания существующих сериальных алгоритмов.
7. Реализация параллельных алгоритмов с использованием процессов.
8. Виды межпроцессной коммуникации.
9. Принцип «Разделяй-и-властвуй».
10. Виды декомпозиции задачи: по данным, по коду, по выполнению, комбинированные.
11. Сортировка, выпуклые оболочки.
12. Процессы и потоки (нити) управления.
13. Понятие процесса.
14. Процессы с поддержкой многопоточности.
15. Диспетчеризация процессов.
16. Динамическая и статистическая связь между процессором и очередью.
17. Средства разработки распределенных вычислений с использованием параллельной виртуальной машины PVM.
18. Использование функций библиотеки PvmLib.
19. Состав функций.
20. Компиляция и запуск программ в среде PVM
21. Состав функций, типы данных, организация обмена данными.
22. Среда выполнения LAM.
23. Трансляция, компоновка, загрузка задач.
24. Языки параллельного программирования и среды программирования.
25. Происхождение языка и его перенос на нетранспьютерные системы.
26. Процесс доставки данных в вычислительный узел.
27. Процессы-заглушки.
28. Высокоуровневый и низкоуровневый интерфейс программирования.
29. XDR-преобразования, аутентификация, широковещательный режим.
30. Использование технологий DCOM и CORBA для создания распределенных приложений.

10. САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩИХСЯ

Форма обучения очная

Название разделов и тем	Вид самостоятельной работы (<i>проработка учебного материала, решение задач, реферат, доклад, контрольная работа, подготовка к сдаче зачета, экзамена и др.</i>)	Объем в часах	Форма контроля (<i>проверка решения задач, реферата и др.</i>)
Понятие высокопроизводительных ПВС (ВПВС).	чтение основной и дополнительной литературы, самостоятельное изучение	4	опрос

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		


	материала по литературным источникам;		
Параллельные алгоритмы.	чтение основной и дополнительной литературы, самостоятельное изучение материала по литературным источникам;	4	опрос
Алгоритмизация параллельных вычислений.	чтение основной и дополнительной литературы, самостоятельное изучение материала по литературным источникам;	4	опрос
Средства разработки параллельных вычислений с использованием потоков (нитей) кода.	чтение основной и дополнительной литературы, самостоятельное изучение материала по литературным источникам;	4	Проверка решения задач
Средства разработки распределенных вычислений.	самостоятельное выполнение практических заданий репродуктивного типа (ответы на вопросы, тренировочные упражнения, задачи, тесты);	4	опрос
Протокол и библиотеки MPI.	самостоятельное выполнение практических заданий репродуктивного типа (ответы на вопросы, тренировочные упражнения, задачи, тесты);	6	Проверка решения задач
Средства создания и отладки параллельных задач.	самостоятельное выполнение практических заданий репродуктивного типа (ответы на вопросы, тренировочные упражнения, задачи, тесты);	4	Проверка решения задач
Язык OCCAM, конструкции языка, понятие процессов, каналов.	чтение основной и дополнительной литературы, самостоятельное изучение материала по литературным источникам;	8	опрос
Удаленные вызовы процедур.	чтение основной и дополнительной литературы, самостоятельное изучение материала по литературным источникам;	10	опрос

10. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

а) Список рекомендуемой литературы

основная

- 1) Параллельное программирование с использованием OpenMP : учеб. пособие / Левин Михаил Петрович. - Москва : Интернет-Ун-т информ. технологий : БИНОМ

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

: Лаборатория знаний, 2008. - 118 с. : ил. - (Основы информационных технологий).
- Библиогр.: с. 113-118. - ISBN 978-5-94774-857-4 (в пер.).

- 2) Теория и практика параллельных вычислений : учеб. пособие / Гергель Виктор Павлович. - Москва : Интернет-Ун-т Информ. Технологий : БИНОМ : Лаборатория знаний, 2007. - 423 с. - (Основы информационных технологий). - Библиогр.: с. 418-423. - ISBN 978-5-9556-0096-3 (ИНТУИТ.РУ) (в пер.). - ISBN 978-5-94774-645-7 (БИНОМ.ЛЗ) (в пер.).

дополнительная

- 1) Элементы параллельного программирования / В. А. Вальковский, В. Е. Котов, А. Г. Марчук, Н. Н. Миренков; под ред. В. Е. Котова. - Москва : Радио и связь, 1983. - 239 с. : ил. - ISBN (в пер.).
- 2) Распараллеливание алгоритмов и программ : структурный подход / Вальковский Владимир Алексеевич. - Москва : Радио и связь, 1989. - 175 с. : ил. - ISBN 5-256-00195-7.

учебно-методическая

- 1) Параллельные алгоритмы для решения задач защиты информации : монография / Бабенко Людмила Климентьевна, Е. А. Ищукова, И. Д. Сидоров. - Москва : Горячая линия-Телеком, 2014. - 304 с. : ил. - Библиогр.: с. 222-224. - ISBN 978-5-9912-0426-2.

Согласовано:


Согласовано:
ДИРЕКТОР НБ / **БУРХАНОВА М.М.** /  / 
 Должность сотрудника научной библиотеки / ФИО / подпись / дата

1. Microsoft Word
2. OpenMP.

в) Профессиональные базы данных, информационно-справочные системы:

1. Электронно-библиотечные системы:

- 1.1. **IPRbooks**: электронно-библиотечная система/ группа компаний Ай Пи Эр Медиа. - Саратов, [2021]. – URL:<http://www.iprbookshop.ru>. Режим доступа: для зарегистрированных пользователей. – Текст: электронный.
- 1.2. **ЮРАЙТ** : электронно-библиотечная система : сайт / ООО Электронное издательство ЮРАЙТ.– Москва, [2021]. -: URL: <https://www.biblio-online.ru>. Режим доступа: для зарегистрированных пользователей. – Текст: электронный.
- 1.3. **Консультант студента** : электронно-библиотечная система: сайт / ООО Политехресурс.– Москва, [2021]. - URL: <http://www.studentlibrary.ru/cgi-bin/mb4x>. Режим доступа: для зарегистрированных пользователей. – Текст: электронный.
- 1.4. **Лань**: электронно-библиотечная система: сайт / ООО ЭБС Лань. - Санкт-Петербург, [2021]. - URL: <https://e.lanbook.com>. Режим доступа: для зарегистрированных пользователей. – Текст: электронный.
- 1.5. **Znanium.com** : электронно-библиотечная система: сайт / ООО Знаниум. – Москва, [2021]. - URL: <http://znanium.com>. Режим доступа: для зарегистрированных пользователей. – Текст: электронный.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

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:

– для лиц с нарушениями зрения: в форме электронного документа; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации;

– для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации;

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

В случае необходимости использования в учебном процессе частично/исключительно дистанционных образовательных технологий, организация работы ППС с обучающимися с ОВЗ и инвалидами предусматривается в электронной информационно-образовательной среде с учетом их индивидуальных психофизических особенностей.

Разработчик


подпись

доцент кафедры ТТС

должность

Булаев А.А.

ФИО