

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кемеровский государственный университет»

Математический факультет



УТВЕРЖДАЮ

Декан математического факультета

ГУДОВ А.М.

2015 г.

Рабочая программа дисциплины (модуля)

Б1.В.ОД.10 Технологии параллельного программирования

Направление подготовки

01.03.02 Прикладная математика и информатика

Направленность (профиль) подготовки

Исследование операций и системный анализ

Уровень бакалавриата

Форма обучения

Очная

Кемерово 2015

Рабочая программа дисциплины утверждена Ученым советом факультета
(протокол Ученого совета факультета № 9 от 13. 04. 2015 г.)

Утверждена с обновлениями в части реорганизации структуры факультета
(протокол Ученого совета факультета № 12 от 22. 06. 2015 г.)

Рабочая программа дисциплины одобрена на заседании кафедры ЮНЕСКО по ИВТ

СОДЕРЖАНИЕ

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы.....	4
2. Место дисциплины в структуре программы бакалавриата.....	4
3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся.....	5
3.1. Объем дисциплины (модуля) по видам учебных занятий (в часах).....	5
4. Содержание дисциплины (модуля), структурированное по разделам (темам) с указанием отведенного на них количества академических часов и видов учебных занятий 5	
4.1. Разделы дисциплины (модуля) и трудоемкость по видам учебных занятий (в академических часах).....	5
4.2. Содержание дисциплины (модуля), структурированное по темам (разделам) 6	
5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю).....	11
6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю).....	11
6.1. Паспорт фонда оценочных средств по дисциплине (модулю).....	11
6.2. Типовые контрольные задания или иные материалы.....	12
6.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующие этапы формирования компетенций	22
7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля).....	23
8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (далее – сеть «Интернет»), необходимых для освоения дисциплины (модуля).....	24
9. Методические указания для обучающихся по освоению дисциплины (модуля) 24	
10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем.....	26
11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю).....	27
12. Иные сведения и материалы.....	27
12.1. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья	27

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения программы бакалавриата обучающийся должен овладеть следующими результатами обучения по дисциплине (модулю):

Коды компетенций	Результаты освоения ООП Содержание компетенций	Перечень планируемых результатов обучения по дисциплине
ОПК-3	способность 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: <ul style="list-style-type: none">– основные направления развития высокопроизводительных компьютеров; Владеть: <ul style="list-style-type: none">– методологией и навыками решения научных и практических задач, используя многопроцессорные вычислительные системы и параллельное программирование;
ПК-7	способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения	Уметь: <ul style="list-style-type: none">– применять на практике параллельное программирование;

2. Место дисциплины в структуре программы бакалавриата

Дисциплина входит в вариативную часть УП ООП с кодом **Б1.В.ОД.10.**

Целью освоения дисциплины “Технологии параллельного программирования” является – освоение базовых знаний в области архитектуры современных многопроцессорных вычислительных систем, параллельной обработки информации, технологий организации параллельных вычислений на многопроцессорных вычислительных комплексах с распределенной или общей оперативной памятью.

Курс занимает особое место в учебном плане среди дисциплин факультета по его значению. Вместе с курсами по программированию, курс “Технологии параллельного программирования” составляет основу образования студента в части современных информационных технологий. Курс рассчитан на студентов, имеющих подготовку по предшествующим курсам, касающихся основам программирования с использованием алгоритмических языков Си или Фортран, вычислительным методам. В течение преподавания курса предполагается, что студенты знакомы с основными понятиями алгебры, комбинаторики, логики, информатики, которые читаются на факультете перед изучением данной дисциплины.

Знания, навыки и умения, приобретенные в результате прохождения курса, будут востребованы при изучении дисциплин, связанных с распараллеливанием различных

вычислительных алгоритмов с применением наиболее популярных технологий параллельных вычислений, а также при проведении вычислительных экспериментов в случае выполнения итоговой квалификационной работы, связанной с реализацией высокоэффективных параллельных алгоритмов.

Дисциплина изучается на 2 курсе в 3 семестре.

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины составляет 2 зачетных единиц, 72 академических часа.

3.1. Объем дисциплины (модуля) по видам учебных занятий (в часах)

Объём дисциплины	Всего часов	
	для очной формы обучения	
Общая трудоемкость дисциплины	72	
Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)	54	
Аудиторная работа (всего):	54	
в том числе:		
лекции	18	
лабораторные работы	36	
Самостоятельная работа обучающихся (всего)	18	
Вид промежуточной аттестации обучающегося (зачет / экзамен)	зачет	

4. Содержание дисциплины (модуля), структурированное по разделам (темам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкость по видам учебных занятий (в академических часах)

№ п/п	Раздел дисциплины	Общая трудоемкость (часы)	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость (в часах)		Формы текущего контроля успеваемости	
			аудиторные учебные занятия			
			Лек.	Лаб.		
1	Основные направления развития высокопроизводительных компьютеров	1	1	0	0	Проверка тестовых заданий, домашнего задания.

2	Однопроцессорная оптимизация алгоритмов	5	1	2	2	Проверка тестовых заданий, домашнего задания.
3	Оценки производительности вычислительных систем	3	1	2	0	Проверка тестовых заданий, домашнего задания.
4	Классификация многопроцессорных вычислительных систем	1	1	0	0	Проверка тестовых заданий.
5	Проблемы создания кластерных систем	6	2	2	2	Проверка тестовых заданий.
6	Парадигмы, модели и технологии параллельного программирования	2	2	0	0	Проверка тестовых заданий.
7	Параллельное программирование с использованием интерфейса передачи сообщений MPI	28	2	22	4	Проверка тестовых заданий, домашних заданий, семестрового задания.
8	Параллельное программирование на системах с общей памятью (OpenMP)	6	2	2	2	Проверка тестовых заданий, домашнего задания.
9	Параллельное программирование на системах смешанного типа	2	2	0	0	Проверка тестовых заданий.
10	Отладка, трассировка и профилирование параллельных программ	5	1	2	2	Проверка тестовых заданий, домашнего задания.
11	Основные понятия параллелизма алгоритмов	5	1	2	2	Проверка тестовых заданий, домашнего задания.
12	Алгоритмы матричной алгебры и их распараллеливание	6	2	2	2	Проверка тестовых заданий, домашнего задания.
		72	18	36	18	
	Всего					

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Содержание лекционных занятий

	Наименование раздела дисциплины	Содержание раздела дисциплины
	Основные направления развития высокопроизводительных компьютеров	Пути достижения параллелизма: независимость функционирования отдельных функциональных устройств, избыточность элементов вычислительной системы, дублирование устройств. Векторная и конвейерная обработка данных. Многопроцессорная и многомашинная, параллельная обработка данных. Закон Мура, сдерживающие факторы наращивания количества

		транзисторов на кристалле и частоты процессоров. Привлекательность подхода параллельной обработки данных. Сдерживающие факторы повсеместного внедрения параллельных вычислений. Ведомственные, национальные и другие программы, направленные на развитие параллельных вычислений в России. Необходимость изучения дисциплины параллельного программирования. Перечень критических задач, решение которых без использования параллельных вычислений затруднено или вовсе невозможно. Содержание курса параллельного программирования – характеристика основных составляющих блоков лекционного курса, практических занятий. Список основной и дополнительной литературы.
	Однопроцессорная оптимизация	Повышение производительности процессора при обработке массивов с использованием циклов. Явные циклы с постоянными границами. Многократное использование КЭШа. Использование предварительной выборки при загрузке данных из оперативной памяти в кэш второго уровня. Изменение инструкций задачи для лучшего кэширования. Архитектурно-зависимая оптимизация; отличия развертывания циклов для векторных и кэш-ориентированных архитектур. Конвейерная обработка данных. Зависимость производительности процессора от способа описания и хранения данных.
	Оценки производительности вычислительных систем	Стандартные методики измерения производительности MIPS, MFLOPS и т.д. Производительность кластера – латентность, пропускная способность. Общепризнанные методики измерения производительности многопроцессорных вычислительных систем. TOP500 – мировой рейтинг суперкомпьютеров (краткая характеристика первых 5-и суперкомпьютеров мира). TOP50 – Российский рейтинг суперкомпьютеров (краткая характеристика первых 5-и суперкомпьютеров России). Перечень с характеристикой Сибирских суперкомпьютерных центров (Томск, Новосибирск, Красноярск, Омск). Состояние вопроса параллельных вычислений в Кемеровском государственном университете.
	Классификация многопроцессорных вычислительных систем	Системы с распределенной, общей памятью, примеры систем. Массивно-параллельные системы (MPP). Симметричные мультипроцессорные системы (SMP). Параллельные векторные системы (PVP). Системы с неоднородным доступом к памяти (Numa), примеры систем. Компьютерные кластеры – специализированные и полнофункциональные. История возникновения компьютерных кластеров – проект Beowulf. Мета-компьютинг – примеры

		действующих проектов. Классификация Флинна, Шора и т.д. Организация межпроцессорных связей – коммуникационные топологии. Примеры сетевых решений для создания кластерных систем. Современные микропроцессоры, используемые при построении кластерных решений. Компания Т-платформы.
	Проблемы создания кластерных систем	Будущий кластер определяют существующие задачи. Плюсы и минусы своего кластера и арендуемого. Бюджет на обслуживание и содержание кластера. Кадровое обеспечение. Базовая инфраструктура: помещение, системы охлаждения воздуха, пол, безопасность. Проектирование архитектуры: базовые компоненты кластера, компоновка кластера, выбор процессора, оперативная память, диски, головной узел кластера, сетевая инфраструктура (коммуникационная, транспортная, сервисная). Поставка и монтаж оборудования. Установка и настройка программного обеспечения. Средства разработки и прикладное программное обеспечение. Мониторинг кластера. Обслуживание кластера. Информационно-вычислительный портал Кемеровского государственного университета.
	Парадигмы, модели и технологии параллельного программирования	Функциональный параллелизм, параллелизм по данным. Парадигма master-slave. Парадигма SPMD. Парадигма конвейеризации. Парадигма “разделяй и властвуй”. Спекулятивный параллелизм. Важность выбора технологии для реализации алгоритма. Модель обмена сообщениями – MPI. Модель общей памяти – OPENMP. Концепция виртуальной, разделяемой памяти – Linda. Российские разработки – Т-система, система DVM. Проблемы создания средства автоматического распараллеливания программ.
	Параллельное программирование с использованием интерфейса передачи сообщений MPI	Библиотека MPI. Модель SIMD. Инициализация и завершение MPI-приложения. Точечные обмены данными между процессами MPI-программы. Режимы буферизации. Проблема deadlock'ов. Коллективные взаимодействия процессов в MPI. Управление группами и коммуникаторами в MPI.
	Параллельное программирование на системах с общей памятью (OpenMP)	Введение в OpenMP. Стандарты программирования для систем с разделяемой памятью. Создание многопоточных приложений. Использование многопоточности при программировании для многоядерных платформ. Синхронизация данных между ветвями в параллельной программе. Директивы языка OpenMP.
	Параллельное программирование на системах смешанного типа	Гибридные модели программирования SMP-систем. Передача данных между узлами кластера функциями MPI, обмен данными внутри узла между ядрами процессора через потоки OpenMP. Правила запуска параллельных приложений, написанных с

		использованием OpenMP+MPI.
0	Отладка, трассировка и профилирование параллельных программ	Классификация ошибок параллельных программ (сильные, слабые ошибки ...). Особенности отладки параллельных приложений. Трассировка. Отладка с помощью последовательных отладчиков. Параллельный Отладчик TotalView. Профилирование. Библиотека MPE. Средства MPI для визуализации трассы параллельного приложения.
1	Основные понятия параллелизма алгоритмов	Степень параллелизма численного алгоритма. Средняя степень параллелизма численного алгоритма. Зернистость алгоритма. Ускорение и эффективность. Закон Амдала. Алгоритм исследования свойств параллельного алгоритма. Определение параллелизма: анализ задачи с целью выделить подзадачи, которые могут выполняться одновременно. Выявление параллелизма: изменение структуры задачи таким образом, чтобы можно было эффективно выполнять подзадачи. Для этого часто требуется найти зависимости между подзадачами и организовать исходный код так, чтобы ими можно было эффективно управлять. Выражение параллелизма: реализация параллельного алгоритма в исходном коде с помощью системы обозначений параллельного программирования.
2	Алгоритмы матричной алгебры и их распараллеливание	Параллельный алгоритм умножения матрицы на вектор и его ускорение по сравнению с последовательным алгоритмом. Параллельный алгоритм умножения матрицы на матрицу и его ускорение по сравнению с последовательным алгоритмом. Параллельный алгоритм решения СЛАУ прямым методом Гаусса и его ускорение по сравнению с последовательным алгоритмом. Параллельный алгоритм решения СЛАУ итерационными методами Якоби, Гаусса - Зейделя и их ускорение по сравнению с последовательным алгоритмом.

Содержание лабораторных занятий

	Наименование раздела дисциплины	Содержание раздела дисциплины
	Основные направления развития высокопроизводительных компьютеров	Домашнее задание: обзор направлений развития вычислительных систем с нетрадиционной архитектурой.
	Однопроцессорная оптимизация	Содержание лабораторной работы: – Однопроцессорная оптимизация предложенных алгоритмов. Домашнее задание: оптимизация предложенных алгоритмов, определение эффективности оптимизации.

	Оценки производительности вычислительных систем	<p>Содержание лабораторной работы:</p> <ul style="list-style-type: none"> – Определение производительности кластерной системы. <p>Домашнее задание: определение производительности домашнего компьютера.</p>
	Проблемы создания кластерных систем	<p>Содержание лабораторной работы:</p> <ul style="list-style-type: none"> – Изучение настройки кластерной системы. <p>Домашнее задание: настройка домашнего компьютера для выполнения последующих домашних заданий.</p>
	Параллельное программирование с использованием интерфейса передачи сообщений MPI	<p>Содержание лабораторных работ:</p> <ul style="list-style-type: none"> – Ознакомление с правилами работы на кластере. MPI: Первая параллельная программа. Компиляция и запуск программ. Тестирование коммуникационной среды кластера. Тестирование производительности кластера. – MPI: Передача данных с помощью блокирующих коммуникационных функций типа “Точка-Точка”. – MPI: Другие виды передачи данных с помощью коммуникационных функций типа “Точка-Точка”. – MPI: Одновременная передача данных. – MPI: Коллективные операции. – MPI: Глобальные вычислительные операции. – MPI: Создание производных типов данных. – MPI: Передача упакованных данных. – MPI: Работа с группами и коммуникаторами, создание виртуальных топологий. <p>Домашние задания: создание параллельных MPI-программ.</p>
	Параллельное программирование на системах с общей памятью (OpenMP)	<p>Содержание лабораторных работ:</p> <ul style="list-style-type: none"> – OpenMP: Первая параллельная программа. Компиляция и запуск программ. – OpenMP: Директивы OpenMP, Переменные окружения. – OpenMP: Библиотечные функции. – OpenMP: Средства синхронизации. – OpenMP: Расширенные возможности. <p>Домашние задания: создание параллельных OpenMP-программ.</p>
0	Отладка, трассировка и профилирование параллельных программ	<p>Содержание лабораторных работ:</p> <ul style="list-style-type: none"> – MPI: Отладка, трассировка, профилирование параллельных программ, оптимизация. – OpenMP: Отладка, трассировка, профилирование параллельных программ, оптимизация. <p>Домашние задания: установка ПО на домашнем компьютере для отладки, трассировки, профилирования параллельных программ.</p>
	Основные понятия	Содержание лабораторных работ:

1	параллелизма алгоритмов	<ul style="list-style-type: none"> – Сравнение теоретических оценок эффективности распараллеливания с полученными численными оценками. <p>Домашние задания: провести теоретическую оценку эффективности распараллеливания параллельной программы и сравнить с численной оценкой.</p>
2	Алгоритмы матричной алгебры и их распараллеливание	<p>Содержание лабораторных и домашних работ:</p> <ul style="list-style-type: none"> – Реализация простейших параллельных алгоритмов матричной алгебры. MPI, OpenMP, MPI+OpenMP.

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

1. Афанасьев К.Е. Основы высокопроизводительных вычислений. Том II. Технологии параллельного программирования: учебное пособие / К. Е. Афанасьев, С. В. Стуколов, В.В. Малышенко, С. Н. Карабцев, Н. Е. Андреев; Кемеровский государственный университет. – Кемерово, 2012. – 416 с. (http://e.lanbook.com/books/element.php?pl1_id=44309&p_f_1_78=1)
2. Афанасьев К.Е. Основы высокопроизводительных вычислений. Том I. Высокопроизводительные вычислительные системы: учебное пособие / К. Е. Афанасьев, С. Ю. Завозкин, С. Н. Трофимов, А. Ю. Власенко; Кемеровский государственный университет. – Кемерово, 2011. – 228 с. (http://e.lanbook.com/books/element.php?pl1_id=30123&p_f_1_78=1)
3. Подбельский В.В., Фомин С.С. Курс программирования на языке Си, Издательство: "ДМК Пресс", 2012 г, 384 с. http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=4148

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)

6.1. Паспорт фонда оценочных средств по дисциплине (модулю)

№ п/п	Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции (или её части) / и ее формулировка – по желанию	наименование оценочного средства
1.	Основные направления развития высокопроизводительных компьютеров	ОПК-3	Тест
2.	Однопроцессорная оптимизация	ПК-7	Тест, домашнее задание
3.	Оценки производительности вычислительных систем	ОПК-3	Тест, домашнее задание
4.	Классификация многопроцессорных вычислительных систем	ПК-7	Тест
5.	Проблемы создания	ОПК-3	Тест, домашнее

№ п/п	Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции (или её части) / и ее формулировка – по желанию	наименование оценочного средства
	кластерных систем		задание
6.	Парадигмы, модели и технологии параллельного программирования	ПК-7	Тест
7.	Параллельное программирование с использованием интерфейса передачи сообщений MPI	ОПК-3, ПК-7	Тест, домашнее задание, контрольная работа
8.	Параллельное программирование на системах с общей памятью (OpenMP)	ОПК-3, ПК-7	Тест, домашнее задание, контрольная работа
9.	Параллельное программирование на системах смешанного типа	ОПК-3, ПК-7	Тест, домашнее задание, контрольная работа
10.	Отладка, трассировка и профилирование параллельных программ	ПК-7	Тест, домашнее задание
11.	Основные понятия параллелизма алгоритмов	ОПК-3	Тест, домашнее задание
12.	Алгоритмы матричной алгебры и их распараллеливание	ОПК-3	Тест, семестровое задание

6.2. Типовые контрольные задания или иные материалы

6.2.1. Зачет

Вопросы к зачету:

- Основные направления развития высокопроизводительных компьютеров. Пути достижения параллелизма: независимость функционирования отдельных функциональных устройств, избыточность элементов вычислительной системы, дублирование устройств.
- Векторная и конвейерная обработка данных.
- Многопроцессорная и многомашинная, параллельная обработка данных.
- Закон Мура, сдерживающие факторы наращивания количества транзисторов на кристалле и частоты процессоров. Сдерживающие факторы повсеместного внедрения параллельных вычислений.
- Перечень критических задач, решение которых без использования параллельных вычислений затруднено или вовсе невозможно.
- Стандартные методики измерения производительности MIPS, MFLOPS и т.д.
- Классификация многопроцессорных вычислительных систем.
- Парадигмы, модели и технологии параллельного программирования.

9. Параллельное программирование с использованием интерфейса передачи сообщений MPI.
10. Параллельное программирование на системах с общей памятью (OpenMP).
11. Параллельное программирование на системах смешанного типа.
12. Параллельное программирование на графических процессорах.
13. Классификация ошибок параллельных программ (сильные, слабые ошибки ...).
14. Особенности отладки параллельных приложений. Трассировка.
15. Определение параллелизма: анализ задачи с целью выделить подзадачи, которые могут выполняться одновременно. Выявление параллелизма: изменение структуры задачи таким образом, чтобы можно было эффективно выполнять подзадачи. Выражение параллелизма: реализация параллельного алгоритма в исходном коде с помощью системы обозначений параллельного программирования.

Тесты по курсу

Проверка знаний основана на тестирующей программе (<http://tests.kemsu.ru>).

Данная программа позволяет разбить общее количество вопросов на подгруппы по темам, а затем, в процессе тестирования случайным образом выбирает вопросы из каждой темы, пропорционально подготовленным вопросам по этой теме. Каждый тест представлен 20 вопросами. После прохождения тестирования проводится собеседование с каждым студентом с разбором результатов тестирования. Итоговая оценка выставляется исходя из: результатов беседы со студентом, рекомендации преподавателя лабораторных занятий, оценки выполнения семестровой работы, качества и обоснованности ответов на дополнительные вопросы.

Есть две возможности прохождения тестирования: в режиме экзамена и в режиме пробного тестирования.

Прохождение тестирования в режиме экзамена проходит при непосредственном участии преподавателя (при помощи специального ПО назначаются студентам логины и пароли).

Прохождение пробного тестирования возможно без участия преподавателя.

Из всего банка вопросов для пробного тестирования доступно около 20% вопросов.

Рекомендуется пройти пробное тестирование, для прохождения которого требуется указание только логина. Логин для пробного тестирования - exam1.

Для этого необходимо:

- 1) в браузере набрать адрес размещения теста в сети Интернет
<http://tests.kemsu.ru/>
- 2) выбрать пункт "Пробное тестирование"
- 3) в появившемся окне ввести логин - "exam1"
- 4) далее выбрать курс и указать одно из двух: "Тестирование по всем темам" или "Выбор темы для тестирования"

Более подробная справочная информация доступна на самом сайте тестирования в пункте "Помощь".

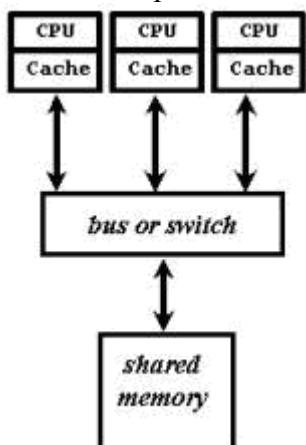
Пройти пробное тестирование сейчас - <http://tests.kemsu.ru/>

Банк тестов насчитывает около 300 вопросов, однако в процессе ведения дисциплины он постоянно пополняется. Так как дисциплина ведется сравнительно недавно, то возможно дополнение его некоторыми темами и рассматриваемыми в них вопросами, что приведет к дополнению теста новыми заданиями.

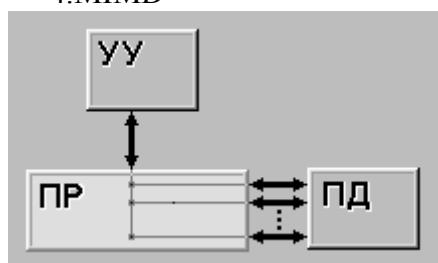
Примеры тестовых заданий:

- 1) Укажите характеристики кластера, оказывающие наибольшее влияние на его вычислительную производительность
 1. количество узлов кластера +
 2. пропускная способность коммуникаций +
 3. объем оперативной памяти

4. объем дискового пространства
 5. вычислительная производительность отдельных узлов кластера +
- 2) Для чего применяется тест LINPACK?
 1. Для тестирования производительности одного вычислительного узла +
 2. Для тестирования производительности кластера
 3. Для тестирования производительности пропускной способности сети
 4. Для тестирования латентности сети
- 3) Для чего применяется тест SCALAPACK?
 1. Для тестирования производительности одного вычислительного узла
 2. Для тестирования производительности кластера +
 3. Для тестирования производительности пропускной способности сети
 4. Для тестирования латентности сети
- 4) Какая архитектура схематично изображена на рисунке?
 1. векторно-конвеерная
 2. массивно-параллельная
 3. с общей памятью +
 4. кластерная



- 5) К какой категории, согласно классификации Флинна, относится архитектура, изображенная на рисунке?
- 1.SISD
 2.SIMD +
 3.MISD
 4.MIMD



Пример билета на зачете

1. Тест (tests.kemsu.ru, групповое тестирование, login – exam1, passwd – 1234567, тест – параллельное программирование).
2. Используя блокирующие коммуникационные функции типа “Точка-Точка”, создайте следующую параллельную программу: на 0-м процессе считывается значение переменной n, которое передается всем остальным процессам.
3. Используя коллективные коммуникационные или вычислительные операции, создайте следующую параллельную программу: на всех процессах задается одномерный массив РПД «Б1.В.ОД.10 Технологии параллельного программирования»

($a_i=rank$, $i=0\dots 10$), который передается на 0-й процесс, 0-й процесс принимает посылку и накапливает сумму в вектор s .

4. Используя конструкторы карт размещения данных в оперативной памяти, создайте следующую параллельную программу передачи данных: на 0-м процессе задается одномерный массив ($a_i=i$, $i=0\dots 20$), с 0-го на 1-й процесс требуется передать одной посылкой 0, 2, 3, 6, 7, 8, 12, 13, 14, 15 элементы массива.
5. Предположим, что многопроцессорная вычислительная система состоит из однородных узлов и для выполнения одной арифметической операции требуется время t , для рассылки машинного слова от одного узла произвольному числу других требуется время αt . Предположим, что последовательный алгоритм обладает максимальной степенью параллелизма и состоит из N^3 арифметических операций, а для сборки результата требуется N^2 пересылок данных.
 - 5.1. Подсчитайте время $T_1(n)$ на реализацию алгоритма на однопроцессорной машине.
 - 5.2. Определите время, затраченное на реализацию параллельного алгоритма $T_p(n)$, если число процессоров p равно 2.
 - 5.3. Вычислите ускорение параллельного алгоритма по сравнению с последовательным.

Шкала оценивания:

Зачет состоит из двух частей: тест из 20-и вопросов и 4-х заданий (3 – проверка приобретенных навыков параллельного программирования, 1 – теоретическая оценка эффективности распараллеливания алгоритма).

Время прохождения теста – 30 мин.

Время для подготовки решений по 4-м заданиям – 30 мин.

Работа оценивается по 100-балльной системе: 20% - тест, 80% - задания.

№	Наименование задания	Количество баллов
1	Тест	20
2-4	Использование функций передачи данных при написании параллельных программ	60
5	Теоретическая оценка эффективности распараллеливания алгоритма	20
Итого		100

6.2.2. Контрольная работа

Пример контрольной работы:

1. Используя функции Send-Recv, осуществите передачу значения переменной с 0-го процесса на 1-й и обратно. Определите имена узлов, задействованных в передаче данных.
2. Используя блокирующие коммуникационные функции типа “Точка-Точка”, создайте следующую параллельную программу: на 0-м процессе считывается значение переменной n , которое передается всем остальным процессам.
3. Используя коллективные коммуникационные или вычислительные операции, создайте следующую параллельную программу: на всех процессах задается

одномерный массив ($a_i=rank$, $i=0\dots 10$), который передается на 0-й процесс, 0-й процесс принимает посылку и накапливает сумму в вектор s .

4. Используя конструкторы карт размещения данных в оперативной памяти, создайте следующую параллельную программу передачи данных: на 0-м процессе задается одномерный массив ($a_i=i$, $i=0\dots 20$), с 0-го на 1-й процесс требуется передать одной посылкой 0, 1, 2, 6, 7, 8, 12, 13, 14 элементы массива.

Шкала оценивания:

Каждое задание оценивается 5 баллами, максимально – 20 баллов.

6.2.3. Домашняя работа

Для выполнения и проверки домашних работ используется информационная система «Информационное обеспечение учебного процесса», расположенная по адресу:

<http://iais.kemsu.ru/proc/prep/course/index.htm>

Форма доступа для студента – по сети Интернет с авторизацией по логину/паролю.

Интерактивность обеспечивается предоставляемым функционалом системы – есть возможность задать вопрос и получить от преподавателя ответ.

Роль преподавателя подразумевает размещение учебных материалов, размещение объявлений, размещение заданий, проверку решений, представленных каждым из студентов.

Примеры домашних заданий

Домашняя работа №1. Введение в MPI

	Текст задания	Количество баллов
	Установить компилятор, скачать и настроить MPICH. В файле с описанием выполнения задания привести скрины, демонстрирующие шаги подключения библиотеки, компиляции и запуска тестовой MPI-программы. Укажите характеристики компьютера, ОС, используемого ПО.	3
	Дать описание и продемонстрировать выполнение функции MPI_Wtick().	1
	Дать описание функции MPI_Get_processor_name и привести пример ее использования.	1

Домашняя работа №2. Блокирующие коммуникации типа точка-точка

	Текст задания	Количество баллов
	Написать программу, используя блокирующие коммуникационные функции (MPI_Send, MPI_Recv), реализующую алгоритм передачи данных по кольцу. В момент запуска параллельного приложения соберите трассу выполнения программы и приведите рисунок трассы в отчете.	3
	Написать программу, используя блокирующие коммуникационные функции (MPI_Send, MPI_Recv), реализующую следующий алгоритм: на 0-м процессе считывается (scanf) значение переменной n , которое передается на другие процессы, все, кроме 0-го, принимают n и распечатывают свой номер и принятое значение. В момент запуска параллельного приложения соберите трассу выполнения программы и приведите рисунок трассы в отчете.	2

Домашняя работа №3. Оценка производительности МВС

	Текст задания	Количество баллов
	РПД «Б1.В.ОД.10 Технологии параллельного программирования»	16

		тво баллов
	Провести обзор и анализ рейтингов ТОП500, ТОП50, дать описание и анализ первых 10 систем из этих рейтингов с указанием систем, вошедших в ТОП500. Укажите критерии анализа и подведите итоги проведенного анализа. Обязательные критерии анализа: архитектура вычислительного комплекса, тип сети (бренд), тип процессора, есть ли графические ускорители или сопроцессоры, ОС, инструменты параллельного программирования (коммуникационные библиотеки, компиляторы, пакеты прикладных программ), область использования вычислительного комплекса, страна размещения и сфера деятельности владельца.	3
	Скачать тест ЛИНПАК, провести тестирование домашнего компьютера, указать характеристики домашнего компьютера, указать откуда скачан тест и какие параметры его запуска. Проведите тестирование с помощью программы суммирования ряда малых чисел, подготовленной на занятии. Объясните причины расхождения полученных результатов тестирования.	3

Домашняя работа №4. Распределение и сборка данных, визуализация трасс выполнения параллельного приложения

	Текст задания	Количество баллов
	Написать программу, используя блокирующие коммуникационные функции (MPI_Send, MPI_Recv), реализующую построчное распределение двумерного массива, расположенного в памяти 0-го процессора. Все остальные процессы выводят номер процесса и полученную строку от 0-го процесса. В момент запуска параллельного приложения соберите трассу выполнения программы и приведите рисунок трассы в отчете.	2
	Написать программу, используя блокирующие коммуникационные функции (MPI_Send, MPI_Recv), реализующую построчную сборку двумерного массива: на каждом процессе задается одномерный массив, который передается на 0-й, 0-й принимает одномерный массив, сразу сохраняя в соответствующую строку двумерного массива. В момент запуска параллельного приложения соберите трассу выполнения программы и приведите рисунок трассы в отчете.	2

Шкала оценивания:

Максимально – 65 баллов.

6.2.4. Семестровая работа

Пример вариантов семестровых заданий:

Вариант №1. Задача о восьми ферзях

Запрограммируйте параллельную программу, реализующую задачу “о восьми ферзях”. Нужно расставить на шахматной доске восемь ферзей так, чтобы они не атаковали друг друга. разработайте программу, которая строит все 92 решения этой задачи. Для реализации используйте модель “управляющий-рабочие”. Пусть управляющий процессор распределяет по другим процессорам варианты восьми начальных позиций ферзей, какие те должны проверить. Обнаружив частичное решение, рабочий процесс должен передать его управляющему. Программа должна найти все решения и завершить работу. Управляющий процесс должен все решения записать в файл результатов.

Вариант №2. Задача о пяти ферзях

Запрограммируйте параллельную программу, реализующую задачу о расстановке пяти

ферзей на шахматной доске, при которой каждое поле будет находиться под ударом одного из них. Для реализации используйте модель “управляющий-рабочие”.

Вариант №3. Задача о коммивояжере

Запрограммируйте параллельную программу, реализующую задачу “о коммивояжере”. Эта классическая задача имеет практическое применение, например, при планировании обслуживания населения городским общественным транспортом. Дано n городов и симметричная матрица $A(N \times N)$. Значением элемента $A(i, j)$ является расстояние между городами i и j . Коммивояжер начинает путь в городе 1 и должен посетить по одному разу каждый город, закончив свой путь снова в городе 1. Требуется найти путь, минимизирующий расстояние, которое придется проехать коммивояжеру, а результат сохранить в векторе $b(N)$. Значением вектора b должна быть перестановка целых чисел от 1 до N , соответствующая порядку посещения городов коммивояжером. Разработайте параллельную программу для решения поставленной задачи, используя модель “управляющий-рабочие”.

Вариант №4. Задача “Игра в жизнь”

Запрограммируйте параллельную программу, реализующую задачу клеточный автомат “игра в жизнь”. Многие биологические и физические системы можно смоделировать в виде набора объектов, которые с течением времени циклически взаимодействуют и развиваются. Некоторые простейшие системы можно моделировать с помощью клеточных автоматов. Основная идея разделить пространство физической или биологической задачи на отдельные клетки. Каждая клетка – это конечный автомат. После инициализации все клетки сначала совершают один переход в новое состояние, затем второй переход и т.д. Результат каждого перехода зависит от текущего состояния клетки и ее соседей. Дано двухмерное поле клеток. Каждая клетка либо содержит организм (жива), либо пуста (мертва). Каждая клетка имеет восемь соседей, которые расположены сверху, снизу, слева, справа и по четырем диагоналям от нее. Игра “жизнь” происходит следующим образом. Сначала поле инициализируется: определяются мертвые и живые клетки (для этой цели в программе можно использовать генератор случайных чисел). Затем каждая клетка проверяет состояние свое и своих соседей и изменяет свое состояние в соответствии со следующими правилами:

- живая клетка, возле которой меньше двух живых клеток, умирает от одиночества;
- живая клетка, возле которой есть две или три живые клетки, выживает еще на одно поколение;
- живая клетка, возле которой находится больше трех живых клеток, умирает от перенаселения;
- мертвая клетка, рядом с которой есть ровно три живых соседа, оживает.

Этот процесс повторяется заданное число шагов (поколений). Поле требуется разделить на полосы или блоки клеток, при этом каждая полоса или блок клеток обрабатываются отдельным процессором. Для реализации алгоритма потребуется организовать передачу данных о состоянии пограничных клеток.

Вариант №5. Задача об “обедающих философах”



Запрограммируйте параллельную программу, реализующую задачу “об обедающих философах”. Для реализации потребуется пять процессоров. Суть задачи следующая: пять философов сидят за круглым столом. Они проводят жизнь, чередуя приемы пищи и размышления. В центре стола находится большое блюдо спагетти. Философам, чтобы съесть порцию спагетти, требуется две вилки. Вилок всего пять: между каждой парой философов лежит по одной вилке. Каждому философу дозволительно пользоваться только вилками, которые лежат рядом с ним (слева и справа). Задача – написать программу, моделирующую поведение философов. Программа должна избегать ситуации, в которой все философы голодны, то есть ни один из них не может взять себе две вилки (например, когда каждый философ держит по одной вилке и не хочет отдавать ее). Раз вилок всего пять, то одновременно могут есть не более, чем двое философов. Два сидящих рядом философа не могут есть одновременно. Предположим, что периоды раздумий и приемов пищи различны – для их имитации в программе можно использовать генератор случайных чисел. Имитация поведения каждого философа может быть разбита на следующие блоки: поразмыслить, взять вилки, поесть, отдать вилки. Вилки являются разделяемым ресурсом. Запрограммируйте остановку алгоритма по достижении контрольного времени. Выведите в файл результатов общее время реализации параллельного алгоритма, количество приемов пищи для каждого философа, постройте схему работы алгоритма.

Например:

Этап	1 философ	2 философ	3 философ	4 философ	5 философ

1	размышлени е	размышлен ие	размышлени е	размышлен ие	размышлени е
2	Взял левую вилку	размышлен ие	Взял левую вилку	размышлен ие	размышлени е
3	Взял правую вилку	размышлен ие	Взял правую вилку	размышлен ие	Взял левую вилку
4	поел	размышлен ие	поел	размышлен ие	размышлени е
5	Отдал вилки	размышлен ие	Отдал вилки	размышлен ие	Взял правую вилку

Вариант №6. “Решето Эратосфена”

Запрограммируйте параллельную программу, реализующую алгоритм “решето Эратосфена” для нахождения всех простых чисел меньше n . Решетом Эратосфена называют следующий способ. Выпишем подряд все целые числа от 2 до n . Первое простое число 2. Запомним его, а все большие числа, кратные 2, вычеркнем. Первое из оставшихся чисел 3. запомним его, а все большие числа, кратные 3, вычеркнем. Первое из оставшихся чисел 5 (4 уже вычеркнуто как кратное 2), запомним его, а все большие числа кратные 5, вычеркнем и т.д.

Вариант №7. Метод Монте-Карло

Запрограммируйте параллельную программу, реализующую алгоритм метода Монте-Карло для вычисления площади круга $x^2 + y^2 \leq 1$. Методы Монте-Карло – это общее название группы методов для решения различных задач с помощью случайных последовательностей. Эти методы (как и вся теория вероятностей) выросли из попыток людей улучшить свои шансы в азартных играх. Этим объясняется и тот факт, что название этой группе методов дал город Монте-Карло – столица европейского игорного бизнеса. Суть метода Монте-Карло для приближенного вычисления значения кратного интеграла

$$I = \iint_D f(x, y) dx dy$$

, где D - область интегрирования, вписанная в прямоугольник $D' = (a \leq x \leq b, c \leq y \leq d)$, сводится к следующему: выбирается случайным образом n точек $z_1 \dots z_n$, расположенных в прямоугольнике D' ; вычисляется приближенное

$$I \approx \frac{(b-a)(d-c)}{n} \sum_{k=1}^n f(z_k)$$

значение интеграла по формуле - , где m - количество точек, попавших внутрь D . Для подсчета площади круга требуется найти интеграл:

$$I = \iint_{x^2+y^2 \leq 1} dx dy$$

. При реализации параллельного алгоритма каждый процессор генерирует свое пространство случайных чисел и их обрабатывает. Запишите в файл результат выполнения параллельной программы, содержащий общее количество сгенерированных случайным образом точек n , количество попавших в заданную область точек m , приближенное вычисленное значение площади круга, погрешность вычисления.

Вариант №8. Умножение матрицы на вектор

Реализуйте последовательный алгоритм умножения матрицы на вектор, получите зависимость времени реализации алгоритма от размера матрицы. Реализуйте параллельный строчно-ориентированный алгоритм умножения матрицы на вектор, вычислите время реализации алгоритма на различном числе процессоров для размера матрицы от 100x100 до 1000x1000. Вычислите ускорение и эффективность параллельного алгоритма по сравнению с последовательным в зависимости от размера матрицы. Реализуйте параллельный столбцово-ориентированный алгоритм умножения матрицы на вектор, вычислите время реализации алгоритма на различном числе процессоров для размера матрицы от 100x100 до 1000x1000. Вычислите ускорение и эффективность параллельного алгоритма по сравнению с последовательным в зависимости от размера матрицы. Проведите сравнение параллельных алгоритмов (строчно- и столбцово-ориентированного) по ускорению и эффективности.

Вариант №9. Умножение матрицы на матрицу

Реализуйте последовательный алгоритм умножения матрицы на матрицу, получите зависимость времени реализации алгоритма от размера матрицы. Реализуйте параллельный алгоритм умножения матрицы на матрицу в случае, когда 1 матрица строчно-слоисто, а 2 - целиком распределены по процессорам, вычислите время реализации алгоритма на различном числе процессоров для размера матрицы от 100x100 до 1000x1000. Вычислите ускорение и эффективность параллельного алгоритма по сравнению с последовательным в зависимости от размера матрицы.

Вариант №10. Задача о многопроцессорном расписании (задача о кучах)

Даны m одинаковых процессоров и n независимых задач, каждая из которых может решаться на любом процессоре. Время решения каждой задачи равно t_i , $i = 1 \dots n$. Как распределить задачи по процессорам таким образом, чтобы выполнение всех задач было завершено в кратчайший срок?

Вариант №11. Сортировка последовательности чисел

Дан ряд случайных чисел, написать параллельную программу сортировки данных чисел.

Вариант №12. Определение частоты события

Дан текст, определить частоту встречи слов в тексте.

Требования к выполнению семестровых работ:

Семестровая работа направлена на самостоятельную работу по созданию параллельных программ и исследования эффективности проведенного распараллеливания.

При реализации алгоритмов семестрового задания на параллельных компьютерах для анализа эффективности варианта распараллеливания необходимо проводить исследование по следующей схеме:

1. написать последовательный алгоритм решения задачи;
2. определить время, затраченное на реализацию последовательного алгоритма $T_1(n)$;
3. написать параллельный алгоритм решения задачи с использованием указанной технологии параллельного программирования;
4. определить время, затраченное на реализацию параллельного алгоритма $T_p(n)$ (p – количество процессоров или ядер для многоядерных архитектур);

5. вычислить ускорение параллельного алгоритма по сравнению с последовательным

$$S_p(n) = \frac{T_1(n)}{T_p(n)};$$

$$E_p(n) = \frac{S_p(n)}{p}$$

6. вычислить эффективность параллельного алгоритма

7. Студент сдает параллельную и последовательную версии программ, реализующих заданный алгоритм, таблицы и графики, демонстрирующие основные характеристики результатов выполнения программ на вычислительном кластере, трассировки параллельных программ и результаты анализа эффективности, полученные специализированными программными средствами.

Шкала оценивания:

Оцениваемая характеристика	Количество баллов
оценка модели (описания) последовательного алгоритма	10
оценка модели (описания) параллельного алгоритма	10
оценка тестирования последовательного и параллельного алгоритма	20
оценка эффективности распараллеливания алгоритма	20
оценка теоретических выводов (степень параллелизма, время выполнения последовательного и параллельного алгоритма)	20
Итого	80

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

Для оценки успеваемости студента по дисциплине разработана балльно – рейтинговая система.

Для положительной оценки на зачете необходимо набрать не менее 61 балла.

Предусмотрена рейтинговая система оценки всех видов деятельности.

Текущий контроль (ТК): посещение лекций, лабораторных занятий, выполнение заданий на лабораторных занятиях, выполнение домашних заданий.

Рубежный контроль (РК): тестирование по разделам, проверка контрольных работ, проверка семестровой работы

Итоговый контроль (ИК): зачет.

Максимальное число баллов по каждому виду контроля – 100. Каждый вид деятельности оценивается следующим образом:

- Текущий контроль:

- посещение лекций – 1 балл каждое занятие (максимально 9 баллов);
- посещение лабораторных занятий – 1 балл каждое занятие (максимально 18 баллов);
- выполнение заданий на лабораторных занятиях – максимально 8 баллов;
- выполнение домашних заданий – (максимально 65 баллов)

- Рубежный контроль:

- Выполнение контрольных работ – максимально – 20 баллов;
- выполнение семестровых заданий – максимально – 80 баллов;

- Итоговый контроль:
 - Зачет – максимально – 100 баллов.

Рейтинговый балл (РБ) рассчитывается с учетом набранных баллов по всем видам контроля со следующими весовыми коэффициентами:

$$РБ=ТК*0,5+РК*0,3+ИК*0,2 \text{ (максимально 100 баллов)}$$

Соответствие рейтингового показателя традиционной оценке:

Рейтинговый балл	Оценка
Менее 60	Незачет
Более 60	Зачет

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

a) основная учебная литература:

Сведения об учебниках			Ко-во экз. в библ. на момент утвержд. РП
Наименование, гриф	Автор	Год издан ия	
Афанасьев К.Е. Основы высокопроизводительных вычислений. Том II. Технологии параллельного программирования: учебное пособие / К. Е. Афанасьев, С. В. Стуколов, В.В. Малышенко, С. Н. Карабцев, Н. Е. Андреев; Кемеровский государственный университет. – Кемерово, 2012. – 416 с. (http://e.lanbook.com/books/element.php?pl1_id=44309&p_f_1_78=1)	К.Е. Афанасьев и др.	2012	ЛАНЬ
Афанасьев К.Е. Основы высокопроизводительных вычислений. Том I. Высокопроизводительные вычислительные системы: учебное пособие / К. Е. Афанасьев, С. Ю. Завозкин, С. Н. Трофимов, А. Ю. Власенко; Кемеровский государственный университет. – Кемерово, 2011. – 228 с. (http://e.lanbook.com/books/element.php?pl1_id=30123&p_f_1_78=1)	К.Е. Афанасьев и др.	2011	ЛАНЬ
Подбельский В.В., Фомин С.С. Курс программирования на языке Си, Издательство: "ДМК Пресс", 2012 г, 384 с. http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=4148	Подбельский В.В., Фомин С.С.	2012	ЛАНЬ

б) дополнительная учебная литература:

1. Топорков В.В. Модели распределенных вычислений / М.: "Физматлит", 2011, 320 с., http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=2339

2. Уильямс Э. Параллельное программирование на C++ в действии. Практика разработки многопоточных программ [Текст] / Э. Уильямс, "ДМК Пресс", 2012 - 672 с. (http://e.lanbook.com/books/element.php?pl1_id=4813)
3. Федотов И.Е. Модели параллельного программирования [Текст] / И.Е. Федотов, "СОЛООН-Пресс", 2012 - 384 с. (http://e.lanbook.com/books/element.php?pl1_id=13807)
4. Богачёв К.Ю. Основы параллельного программирования: учебное пособие [Текст] / К.Ю. Богачёв, "Бином. Лаборатория знаний", 2013 - 342 с. (http://e.lanbook.com/books/element.php?pl1_id=42626)
5. Афанасьев, К.Е. Основы высокопроизводительных вычислений. Т.3. Параллельные вычислительные алгоритмы: учебное пособие [Текст] / К.Е. Афанасьев, И.В. Григорьева, Т.С. Рейн; Кемеровский государственный университет. – Кемерово, 2012. – 185 с. (http://e.lanbook.com/books/element.php?pl1_id=44308&p_f_1_78=1)

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (далее – сеть «Интернет»), необходимых для освоения дисциплины (модуля)

1. tests.kemsu.ru – web-ориентированная система тестирования КемГУ
2. iais.kemsu.ru – система поддержки учебного процесса КемГУ
3. Спецификация языка UPC: http://upc.lbl.gov/docs/user/upc_spec_1.2.pdf
4. <http://upc.gwu.edu/downloads/Manual-1.2.pdf>
5. Nvidia CUDA 2.2: Programming Guide. http://developer.download.nvidia.com/compute/cuda/2_21/toolkit/docs/NVIDIA_CUDA_Programming_Guide_2.2.1.pdf
6. CUDA University Courses, http://www.nvidia.com/object/cuda_university_courses.html
7. Букатов А.А., Дацюк В.Н., Жегуло А.И. Программирование многопроцессорных вычислительных систем. Ростов-на-Дону. Издательство ООО «ЦВВР», 2003, 208 с. (<http://rsusu1.rnd.runnet.ru/tutor/method/index.html>)
8. Tutorial on MPI: The Message-Passing Interface William Gropp Mathematics and Computer Science Division Argonne National Laboratory Argonne, IL 60439 www-unix.mcs.anl.gov/mpi/tutorial
9. Антонов А.С. Параллельное программирование с использованием технологии MPI: Учебное пособие. // <http://rsusu1.rnd.runnet.ru/tutor/antonov/>
10. Богданов А., Мареев В., Станнова Е., Корхов В. Архитектуры и топологии многопроцессорных вычислительных систем // электронный учебник <http://www.informika.ru/text/teach/topolog/index.htm>
11. Материалы информационно-аналитического центра НИВЦ МГУ – www.parallel.ru
12. Крюков В. А. Операционные системы распределенных вычислительных систем (распределенные ОС) / <http://parallel.ru/krukov/>
13. Томилин А. Н. Операционные системы супер-ЭВМ / <http://parallel.ru/info/education/tom-kurs.html>
14. Материалы информационно-аналитического центра НИВЦ МГУ – www.parallel.ru

9. Методические указания для обучающихся по освоению дисциплины (модуля)

К современному специалисту общество предъявляет достаточно широкий перечень требований. Данный курс предусматривает изучение теоретических вопросов, в соответствии с рабочей программой, а также самостоятельное выполнение контрольных работ.

В теоретической части курса изучаются базовые понятия по вопросам организации параллельных вычислительных систем, а также основных технологий организации параллельных вычислений на многопроцессорных вычислительных комплексах с распределенной или общей оперативной памятью.

В практической части курса изучаются основы технологий параллельного программирования: параллельное программирование с использованием интерфейса передачи сообщений MPI, параллельное программирование на системах с общей памятью (OpenMP).

Для изучения теоретической части курса необходимо изучить вопросы, рассматриваемые в лекциях. При изучении материала необходимо помимо лекционных материалов использовать рекомендуемую литературу для лучшего усвоения материала. Кроме этого, все необходимые сведения, касающиеся конкретных вопросов построения и использования многопроцессорных вычислительных комплексов в той или иной технологии, можно получить из открытых источников, например <http://parallel.ru>.

9.1. Советы по планированию и организации времени, необходимого для изучения дисциплины. Рекомендуется следующим образом организовать время, необходимое для изучения дисциплины:

Изучение конспекта лекции в тот же день, после лекции – 10-15 минут.

Изучение конспекта лекции за день перед следующей лекцией – 10-15 минут.

Изучение теоретического материала по учебнику и конспекту – 1 час в неделю.

Подготовка к практическому занятию (выполнение домашних заданий) – 1 час.

Всего в неделю – 3 часа 25 минут.

9.2. Описание последовательности действий студента («сценарий изучения дисциплины»).

При изучении дисциплины очень полезно самостоятельно изучать материал, который еще не прочитан на лекции. Тогда лекция будет гораздо понятнее. Однако легче при изучении курса следовать изложению материала на лекции. Для понимания материала и качественного его усвоения рекомендуется такая последовательность действий:

1. После прослушивания лекции и окончания учебных занятий, при подготовке к занятиям следующего дня, нужно сначала просмотреть и обдумать текст лекции, прослушанной сегодня (10-15 минут).

2. При подготовке к лекции следующего дня, нужно просмотреть текст предыдущей лекции, подумать о том, какая может быть тема следующей лекции (10-15 минут).

3. В течение недели выбрать время (1-час) для работы с рекомендованной литературой в библиотеке или ресурсами Интернет.

4. При подготовке к практическим занятиям следующего дня, необходимо сначала повторить пройденный теоретический материал предыдущего занятия по теме домашнего задания. При выполнении упражнения нужно сначала понять, что требуется, какой теоретический материал нужно использовать.

9.3. Рекомендации по использованию материалов учебно-методического комплекса.

Рекомендуется использовать слайд-лекции преподавателя (выдаются преподавателем после каждой лекции).

9.4 Советы по подготовке к зачету. Дополнительно к изучению конспектов лекций необходимо пользоваться учебными пособиями. Рекомендуется использовать электронные учебно-методические пособия по теме курса, имеющиеся на сайте кафедры Юнеско по НИТ (unesco.kemsu.ru), а также зеркале сайта citforum.kemsu.ru (citforum.kemsu.ru). Кроме «заучивания» материала, очень важно добиться состояния понимания изучаемых тем дисциплины. С этой целью рекомендуется после изучения очередного параграфа выполнить несколько упражнений на данную тему. Кроме того, очень полезно мысленно задать себе следующие вопросы (и попробовать ответить на них): о чем этот параграф?, какие новые понятия введены, каков их смысл?, для чего используется та или иная технология параллельного программирования? Рекомендовано пройти предварительное тестирование по курсу с использованием тестирующей программы, доступной по адресу:

tests.kemsu.ru (login для прохождения пробного тестирования - exam1).

9.5. Методические рекомендации по выполнению домашних работ

Для выполнения домашних работ необходимо использовать информационную систему поддержки учебного процесса (<http://iais.kemsu.ru>). Для доступа к данной системе преподаватель на первом занятии выдает каждому логин-пароль. Домашние задания будут появляться в системе еженедельно после проведения лабораторных занятий. Перед выполнением заданий необходимо повторить пройденный материал, а также изучить рекомендуемую преподавателем литературу для выполнения заданий. Оформление отчета по домашней работе следует выполнить по предлагаемому шаблону (размещен в разделе “Учебные материалы” в системе поддержки учебного процесса).

9.6. Методические рекомендации по выполнению семестровых работ

Семестровая работа направлена на самостоятельную работу по созданию параллельных программ и исследования эффективности проведенного распараллеливания.

При реализации алгоритмов семестрового задания на параллельных компьютерах для анализа эффективности варианта распараллеливания необходимо проводить исследование по следующей схеме:

8. написать последовательный алгоритм решения задачи;
9. определить время, затраченное на реализацию последовательного алгоритма $T_1(n)$;
10. написать параллельный алгоритм решения задачи с использованием указанной технологии параллельного программирования;
11. определить время, затраченное на реализацию параллельного алгоритма $T_p(n)$ (p – количество процессоров или ядер для многоядерных архитектур);
12. вычислить ускорение параллельного алгоритма по сравнению с последовательным

$$S_p(n) = \frac{T_1(n)}{T_p(n)};$$

13. вычислить эффективность параллельного алгоритма $E_p(n) = \frac{S_p(n)}{p}$.
14. Студент сдает параллельную и последовательную версии программ, реализующих заданный алгоритм, таблицы и графики, демонстрирующие основные характеристики результатов выполнения программ на вычислительном кластере, трассировки параллельных программ и результаты анализа эффективности, полученные специализированными программными средствами.

Семестровые задания даются каждому студенту индивидуально.

Оформление отчета по семестровой работе следует выполнить по предлагаемому шаблону (размещен в разделе “Учебные материалы” в системе поддержки учебного процесса).

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем

При проведении обучения используются следующие информационные системы, являющиеся частью электронной информационно-образовательной среды КемГУ:

1. Система компьютерного адаптивного тестирования (СКАТ).
2. ИС “Информационное обеспечение учебного процесса” (ИнфОУПро) – дистанционное взаимодействие с обучаемыми с целью управления процессом обучения и контроля полученных знаний.
3. ИС “Рейтинг студентов” – учет учебной деятельности студентов с использованием

балльно-рейтингового метода оценивания.

При проведении дистанционного обучения дополнительно используется Skype.

При проведении обучения по дисциплине используются активные и интерактивные формы обучения, включая: лекции-визуализации, лекции-беседы, лекции с разбором конкретных ситуаций, проектная деятельность.

Лекции-визуализации используются на этапе введения студентов в новую тему. Они основаны на использовании в качестве наглядного материала мультимедийной презентации, содержащей такие формы наглядности, как схемы, рисунки, диаграммы и т.д. После освоения студентам базовых знаний по изучаемой теме проводятся лекции беседы, когда студентам адресуются вопросы для обсуждения в начале лекции и по ее ходу. Для пояснения материала изучаемой темы на практическом примере используются лекции с разбором конкретных ситуаций.

На последних практических занятиях, когда студенты изучили теорию и имеют определённые практические навыки, используется проектный подход. Он заключается в выполнении студентами практико-ориентированных проектов по теме курса в форме семестровой работы.

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Лекционные занятия проходят в мультимедийной аудитории, оснащенной компьютером и проектором. Каждая лекция сопровождается презентацией, содержащей краткий теоретический материал и иллюстративный материал. Каждая презентация построена по следующему шаблону: название лекционного занятия, цель и задачи лекции, краткое содержание предыдущей лекции (при необходимости), теоретический материал (разбит на две части с учетом перемены), в конце приведены итоги лекционного занятия, обозначена тема следующей лекции, а также вопросы и задания для самостоятельного изучения.

Презентации по лекционному курсу разбиты по темам, по отдельно взятой теме может быть несколько лекций.

Лабораторные занятия проходят в компьютерном классе. Первая часть занятия посвящена разбору нового материала. Вторая часть – выполнению практических заданий с целью закрепления материала.

Для проведения лабораторных занятий используется кластер-полигон, который построен на базе 2-х ядерных процессоров Intel Core 2 Duo, позволяет использовать технологии MPI или OpenMP, или обе сразу вместе: MPI – для передачи сообщений между узлами кластера, OpenMP – для работы с общей памятью в рамках отдельного узла кластера, UPC – для работы в рамках модели распределенной общей памяти, CUDA – программирование на графических процессорах.

12. Иные сведения и материалы

12.1. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья.

Образование обучающихся с ограниченными возможностями здоровья может быть организовано по выбору студента в следующем виде:

- совместно с другими обучающимися: студент посещает занятия на общих основаниях и непосредственно вовлекается в учебный процесс;

- дистанционно посредством телекоммуникационных технологий: студент прослушивает материал занятий в режиме реального времени, по средствам прямого телемоста (применение Skype или других аналогичных программ и технологий), не находясь непосредственно в учебной аудитории;
- в индивидуальном порядке: преподаватель занимается со студентом индивидуально контактно или посредством телекоммуникационных технологий.

По окончании изучения курса со студентом проводится индивидуальное собеседование, на котором он демонстрирует полученные знания. В случае необходимости, студенту может заранее быть выдано индивидуальное практическое задание, для самостоятельной подготовки (за месяц или за две недели).

Для инвалидов по слуху предусмотрены следующие особенности проведения учебного процесса:

1. Преподаватель предоставляет студенту учебно-методические материалы, необходимые для освоения изучаемого материала (программа курса, план занятия, опорный конспект, методические пособия или слайд презентации, в случае наличия).
2. Лекционный материал преподается в наглядном виде слайд презентаций или сопровождается схемами, наглядными таблицами.
3. Вместо устного ответа студентам предлагается отвечать письменно.
4. Предусматриваются индивидуальные консультации со студентом, на котором может присутствовать сурдопереводчик (университет не обязуется предоставлять сурдопереводчика).

Для инвалидов по зрению предусмотрены следующие особенности проведения учебного процесса:

1. Предусматриваются индивидуальные консультации со студентом, во время которых преподаватель в медленном спокойном темпе объясняет учебный материал (возможно повторно), заостряя внимание на ключевых понятиях.
2. Вместо письменного ответа студентам предлагается отвечать устно.
3. Предлагается ознакомиться с литературой по курсу, написанной шрифтом Брайля, при наличии.

Для инвалидов опорно-двигательного аппарата предусмотрены следующие особенности проведения учебного процесса:

1. Преподаватель предоставляет студенту учебно-методические материалы, необходимые для освоения изучаемого материала (программа курса, план занятия, опорный конспект, методические пособия или слайд презентации, в случае наличия).
2. Лекционный материал преподается в наглядном виде слайд презентаций или сопровождается схемами, наглядными таблицами.
3. Вместо письменного ответа студентам предлагается отвечать устно.
4. Предусматриваются индивидуальные консультации со студентом.

Составитель (и): доцент кафедры ЮНЕСКО по ИВТ Стуколов С.В.

