

МИНОБРНАУКИ РОССИИ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ИНФОРМАТИКИ

УТВЕРЖДАЮ

Декан факультета информатики

Сущенко С.П.

" 29 " декабря 2011 г.

ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Параллельное и распределенное программирование

Направление подготовки

010300 Фундаментальная информатика и информационные технологии

Наименование магистерской программы

Управление проектами по разработке программного обеспечения

Квалификация (степень)

Магистр

Форма обучения

Очная

Томск 2011

1. Цели освоения дисциплины

Целями освоения дисциплины «Параллельное и распределенное программирование» является изучение основных принципов теории параллельных алгоритмов и овладение навыком разработки эффективных программ для вычислителей с общей и распределенной памятью (кластеры).

2. Место дисциплины в структуре основной образовательной программы магистратуры

Раздел образовательной программы: «М.2. Профессиональный цикл. Базовая часть».

Данная дисциплина содержательно и логически связана с дисциплинами, которые изучают методы создания эффективных программ в полной мере использующих ресурсы современных высокопроизводительных вычислительных систем и сетей: «Программирование», «Алгоритмы и анализ сложности», «Операционные системы», «Архитектура вычислительных систем».

Для того, чтобы приступить к изучению курса «Параллельное и распределенное программирование», студент должен обладать следующими знаниями и умениями:

- знать теорию алгоритмов, способы оценки эффективности алгоритмов;
- иметь представление об архитектуре вычислительных систем и сетей;
- знать основные принципы, функции, блоки современных операционных систем;
- иметь опыт работы с операционными системами Windows, Unix;
- владеть одним из языков программирования: C++, Фортран, иметь практический опыт работы со стандартными библиотеками и инструментами разработки программного обеспечения.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины

Данная дисциплина способствует формированию следующих компетенций, предусмотренных ФГОС-3 по направлению подготовки ВПО 010300 – Фундаментальная информатика и информационные технологии:

- способность к профессиональному использованию оборудования и приборов (в соответствии с целями магистерской программы) (ОК- 8);
- способность применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые

технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий, (в соответствии с профилизацией) (ПК-1);

- способность профессионально решать задачи производственной и технологической деятельности с учетом современных достижений науки и техники, включая: разработку алгоритмических и программных решений в области системного и прикладного программирования; разработку математических, информационных и имитационных моделей по тематике выполняемых исследований; создание информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных; разработку тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям; разработку эргономичных человеко-машинных интерфейсов (ПК-2);
- способность углубленного анализа проблем, постановки и обоснования задач научной и проектно-технологической деятельности (ПК-7);
- способность разрабатывать концептуальные и теоретические модели решаемых научных проблем и задач проектной и производственно-технологической деятельности (ПК-8);
- способность разрабатывать архитектурные и функциональные спецификации создаваемых систем и средств, а также разрабатывать абстрактные методы их тестирования (ПК-10);
- способность разрабатывать процедуры и процессы управления качеством производственной деятельности, связанной с созданием и использованием систем информационных технологий (ПК-11);
- способность выполнять работу экспертов в ведомственных, отраслевых или государственных экспертных группах по экспертизе проектов, тематика которых соответствует профилю подготовки магистра информационных технологий (ПК-21);
- способность оказывать консалтинговые услуги по тематике, соответствующей профилю подготовки магистра (ПК-22);
- умение выполнять оценки степени сложности, бюджета, рисков и времени выполнения проекта по разработке программного обеспечения как на стратегическом, так и на тактическом уровне, навыки в разработке и контроля рабочего графика проекта (СК-2).
- умение в проектной деятельности в сфере разработки программного обеспечения, основанное на применении системного подхода, умение строить, применять и читать профессиональные модели для описания и прогнозирования различных артефактов процесса разработки программного обеспечения, осуществлять их качественный и количественный анализ (СК-3).
- умение применять современные технологии разработки программного обеспечения с использованием автоматизированных систем управления и планирования, осуществлять контроль качества разрабатываемых программных продуктов (СК-6).

В результате освоения дисциплины обучающийся должен

знать:

- классификации вычислительных систем;
- основные архитектурные приемы увеличения производительности микропроцессоров;
- классификации, строение многоуровневой памяти;
- теорию параллельных алгоритмов, способы создания параллельных алгоритмов;
- механизмы синхронизации многопоточных программ;
- основные ошибки синхронизации;

уметь:

- правильно выбирать структуры данных и алгоритмы для решения поставленной задачи с использованием ресурсов конкретной вычислительной установки;
- оценивать эффективность параллельного алгоритма; оценивать баланс вычислений и коммуникаций между узлами параллельного алгоритма;
- использовать стандартные средства создания и синхронизации потоков;
- разрабатывать многопоточные программы для систем с общей памятью; преобразовывать готовые последовательные программы в их параллельные аналоги;
- разрабатывать многопоточные программы для систем с массовым параллелизмом;

владеть:

- средствами и библиотеками программирования для вычислителей с общей памятью;
- библиотекой OpenMP для распараллеливания последовательных программ;
- библиотекой MPI для разработки параллельных программ для вычислителей с распределенной памятью (кластеры);
- инструментами разработки, отладки, профилирования параллельных программ.

4. Структура и содержание дисциплины «Параллельное и распределенное программирование»

Общая трудоемкость дисциплины составляет 2 зачетных единицы, 72 часа, из них: лекции – 20 часов, лабораторные работы – 12 часов, самостоятельная работа – 40 часов.

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы (в часах)	Формы текущего контроля успеваемости
-------	-------------------	---------	-----------------	-------------------------------	--------------------------------------

				Лекции	Лабораторные	Самостоятельные	
1	Архитектура высоко-производительных вычислительных систем	2	1-3	6		2	Контрольная работа (зачет)
2	Параллельные алгоритмы и программы	2	4-6	4		4	Опрос Сдача лабораторных работ
3	Классические задачи параллельного программирования	2	7-11	4	6	14	Опрос Сдача лабораторных работ
4	Инструменты многопоточного программирования	2	12-16	4		6	Опрос
5	Реализация параллельных алгоритмов	2	17-21	2	6	14	Контрольная работа (зачет) Сдача лабораторных работ

Тема 1. Архитектура высокопроизводительных вычислительных систем

- параллелизм – как перспективный способ построения вычислительных систем
- классификации вычислительных систем
- архитектурные приемы повышения производительности микропроцессоров
- подсистема памяти (в том числе кэш-память)
- гибридные и специализированные вычислители

Тема 2. Параллельные алгоритмы и программы.

- принципы разработки параллельных алгоритмов
- оценка эффективности параллельных алгоритмов
- методы синхронизации потоков (критическая секция, мьютекс, семафор)

Тема 3. Классические задачи параллельного программирования

- задача об обедающих философах
- задача о читателях и писателях
- команда потоков
- задача о спящем парикмахере

Тема 4. Инструменты многопоточного программирования

- параллельное программирование для систем с общей памятью (POSIX threads, OpenMP).
- параллельное программирование для систем с распределенной памятью (MPI).
- проблемы разработки параллельных программ
- отладчики и профилировщики

Тема 5. Реализация параллельных алгоритмов

- умножение матрицы на вектор
- умножение матрицы на матрицу
- решение системы линейных уравнений
- сортировка
- алгоритмы на графах

5. Образовательные технологии

В ходе преподавания дисциплины используются следующие образовательные технологии:

- лекции;
- лабораторные занятия;
- самостоятельная работа студентов;
- активные и интерактивные формы занятий:
 - проблемная лекция;
 - лекция с разбором конкретной ситуации.

6. Учебно-методическое обеспечение самостоятельной работы студентов. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

Самостоятельная работа студентов по предмету организуется в следующих формах:

- 1) самостоятельное изучение основного теоретического материала, ознакомление с дополнительной литературой, Интернет-ресурсами;
- 2) выполнение лабораторных работ, решение профессиональных задач из реальной предметной области.

В качестве учебно-методического обеспечения самостоятельной работы используется основная и дополнительная литература по предмету, Интернет-ресурсы, материал лекций, указания, выданные преподавателем при проведении лабораторных работ.

Темы лабораторных работ для самостоятельной работы:

1. Классические задачи синхронизации потоков:

- Задача обедающих философов
- Задача «читатели-писатели»
- Задача «спящий парикмахер»
- Команда потоков

2. Задачи для использования библиотеки OpenMP

- Найти максимальный и минимальный элемент в массиве
- Найти сумму элементов в массиве
- Найти скалярное произведение двух векторов a и b длины N .
- Реализовать простейший линейный фильтр «размытие».

3. Задачи для использования библиотеки MPI (выполняется и сдается на кластере)

- Сортировка массива (аналог быстрой сортировки)
- Умножение матрицы на вектор (разбиение по строкам, блочная схема)
- Умножение матрицы на матрицу (разбиение по строкам и столбцам, по строкам и строкам, блочная схема)
- Решение системы линейных уравнений
- Задача коммивояжера (полный перебор).

Вопросы и задания для промежуточного контроля:

1. Два принципиальных пути повышения производительности ВС.

Уровни параллелизма по Треливену и средства их реализации.

Проблемы повышения производительности систем: ILP Wall, Power Wall, Memory Wall.

Многоядерные и многонитевые процессоры. Описание. Отличия. Примеры.

2. Классы вычислительных систем по Флинну (SISD, SIMD, MISD, MIMD)

Системы с общей памятью. (SMP). Описание и примеры. NUMA.

Системы с распределенной памятью (MPP и кластеры). Описание и примеры.

Наборы команд CISC, RISC, VLIW. Описание, особенности, примеры.

3. Методы повышения производительности микропроцессоров (конвейеризация, суперскалярность, неупорядоченное исполнение, переименование регистров, предсказание переходов).

4. Многоуровневая память (Кэш). Классификации кэша (раздельный/совместный, со сквозной записью / с обратной записью, КПО / ПАК / МАК).

Механизм работы кэша. Принципы временной и пространственной локальности.

Многоуровневый кэш. Показатели эффективности работы памяти (задержка или латентность, пропускная способность).

5. Последовательный / параллельный алгоритм (программа).

Показатели эффективности (ускорение, эффективность). Закон Амдала.

Цели создания параллельных программ?

Особенности параллельных программ (сложность написания, сложность отладки).

Требования к "хорошим" параллельным программам.

6. Граф алгоритма, концепция неограниченного параллелизма (см. Воеводин, лекция 3)

Пиковая и реальная производительность.

Простые и конвейерные ФУ (Воеводин, лекция 4).

7. Потоки и процессы.

Средства управления потоками/процессами. Вытесняющая/невытесняющая многозадачность.

Алгоритмы планирования потоков (с приоритетами).

Виртуальное адресное пространство и средства его реализации.

8. Средства синхронизации потоков (критическая секция, мьютекс, семафор).

Тупики, взаимоблокировки. Причины их возникновения, методы устранения.

Ошибки при синхронизации потоков - race conditions (гонки).

9. OpenMP. Что такое, зачем нужен, концепция. Основные конструкции.

10. MPI. Что такое, зачем нужен, для каких машин используется? Основные возможности.

Синхронные и не синхронные функции.

11. GRID. Что это такое. Достоинства и недостатки по сравнению с суперкомпьютерами (например, по сравнению с кластерами).

12. Реконфигурируемые вычислительные системы (FPGA). Вычисления на графических процессорах (GPGPU). Достоинства и недостатки по сравнению с вычислителями общего назначения.

13. Опишите схему параллельного алгоритма умножения матрицы на вектор.

14. Опишите схему параллельного алгоритма умножения матрицы на матрицу.

15. Опишите схему параллельного алгоритма сортировки (аналог быстрой сортировки).

16. Опишите параллельный алгоритм решения системы линейных уравнений.

17. Опишите параллельный вариант алгоритма Флойда (поиск всех кратчайших путей в графе).

18. Опишите, какие способы синхронизации и каким образом используются для решения задач: обедающие философы, читатели и писатели, спящий парикмахер, команда потоков.

19. Отладка, оптимизация, профилирование параллельных программ. Отличия от последовательных программ. Какие инструменты бывают, что позволяют делать.

Успешно освоившими курс считаются студенты, выполнившие все лабораторные работы, сдавшие контрольные работы по темам «Архитектура высокопроизводительных вычислительных систем» и «Реализация параллельных алгоритмов» на положительные оценки, а также успешно прошедшие опросы по темам «Параллельные алгоритмы и

программы», «Инструменты многопоточного программирования», «Инструменты многопоточного программирования». Итоговая оценка может быть получена как среднее арифметическое соответствующих оценок по указанным контрольным мероприятиям.

7. Учебно-методическое и информационное обеспечение дисциплины «Параллельное и распределенное программирование»

Основная литература:

1. Танненбаум Э. Архитектура компьютера. СПб: Питер, 2003.
2. Воеводин Вл.В., Жуматий С.А. Вычислительное дело и кластерные системы. М.: Изд-во МГУ, 2007 – 150 с (электронный вариант доступен по адресу: <http://ClusterBook.Parallel.ru>).
3. Антонов А.С. Параллельное программирование с использованием технологии MPI: Учебное пособие. М.: Изд-во МГУ, 2004 – 71 с.
4. Гергель В.П. (2007). Теория и практика параллельных вычислений. – М.: Интернет-Университет, БИНОМ. Лаборатория знаний.

Дополнительная литература:

1. Воеводин В.В. Вычислительная математика и структура алгоритмов. – М.: Изд-во МГУ, 2006 – 112 с.
2. Каляев И.А., Левин И.И., Семерников Е.А., Шмойлов В.И. Реконфигурируемые мультиконвейерные вычислительные структуры. Ростов-на-Дону: Изд-во ЮНЦ РАН, 2008г.
3. Оленев Н.Н. Основы параллельного программирования в системе MPI, 2007г
4. Richard Gerber, Aart J.C. Bik, Kevin B. Smith, Xinmin Tian. The software optimization cookbook. – Intel Press, 2005.

Программное обеспечение и Интернет-ресурсы:

1. CIT-forum – Материалы сайта <http://citforum.ru>.
2. Russian Software Developer Network – материалы сайта <http://www.rsdn.ru>.
3. Материалы сервера www.parallel.ru
4. Материалы по стандарту OpenMP: <http://openmp.org/wp/>
5. Информация по стандарту MPI: http://parallel.ru/tech/tech_dev/mpi.html, <http://www.openmpi.org/>, <http://www.mcs.anl.gov/research/projects/mpi/>.
6. Материалы по использованию суперкомпьютера SKIF-Cyberia: <http://skif.tsu.ru/>

8. Материально-техническое обеспечение дисциплины

Для материально-технического обеспечения дисциплины требуется наличие компьютерной техники с установленным соответствующим программным обеспечением и другого оборудования, поддерживающего проведение презентаций, построение проектной документации, ведение групповой обработки, выход в сеть Интернет. Также требуется обеспечение литературой, которую в достаточном объеме может предложить книжный фонд Научной библиотеки Томского госуниверситета.

Программа составлена в соответствии с требованиями ФГОС ВПО с учетом рекомендаций и ООП ВПО по направлению подготовки «010300 – Фундаментальная информатика и информационные технологии» и профилю подготовки «Управление проектами по разработке программного обеспечения».

Автор: к.т.н., доцент кафедры прикладной информатики Д. Б. Биматов.

Рецензент: профессор кафедры прикладной информатики, доктор технических наук
Сущенко С.П.

Программа одобрена на заседании кафедры теоретических основ информатики ТГУ
от 14.12.2011 , протокол № 10/11