

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
«Томский государственный университет систем управления и радиоэлектроники»
(ТУСУР)

УТВЕРЖДАЮ
Заведующий кафедрой УИ
_____ А.Ф. Уваров
« ____ » _____ 2012 г.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ
по дисциплине

«Параллельное программирование»

Составлены кафедрой «Управление инновациями»
Для студентов, обучающихся по направлению подготовки 220600 «Инноватика»
Магистерская программа «Мультимедийные многопроцессорные системы на кристалле»

Форма обучения очная

Разработчик:

профессор, д.т.н.

_____ М.Ю. Катаев
« ____ » _____ 2012 г.

Томск 2012 г.

Содержание

Введение	3
Практическое занятие №1.	4
Практическое занятие №2.	5
Практическое занятие №3.	6
Библиографический список.....	8

Введение

Цель данного пособия состоит в выработке навыков в разработке программных средств и интерпретации параллельных программ.

Для полноценного понимания и усвоения материала необходимо предварительно изучить дисциплины "Информатика" и "Основы программирования".

Для углубленного изучения и освоения материала целесообразно применение различных форм самопроверки знаний студентов: тесты, задачи, упражнения. Они могут быть использованы при проведении практических занятий в университете, выполнении курсовых, контрольных и аудиторных работ, а также при самостоятельном изучении данных дисциплин.

Одним из наиболее интенсивных способов изучения дисциплины является самостоятельная реализация различных алгоритмов, в виде программного кода. При этом вырабатывается опыт и навыки, необходимые при разработке сложных программных средств.

Предлагаемые задания позволят глубже освоить теоретические и практические вопросы парадигмы параллельного программирования, понять принципы написания параллельных программ, научиться грамотно применять эти теоретические знания при разработке программных средств различной сложности.

Практическое занятие №1.

Работа в системе Paralab (ПараЛаб)

1. Общая характеристика системы. Правила использования системы ПараЛаб
2. Сортировка данных (Пузырьковая сортировка, Сортировка Шелла, Быстрая сортировка)
3. Матричное умножение. (Ленточный алгоритм, Алгоритмы Фокса и Кэннона)
4. Обработка графов (Алгоритм Прима поиска минимального охватывающего дерева, Алгоритм Дейкстры поиска кратчайших путей)
5. Выполнение вычислительных экспериментов (Использование результатов экспериментов: запоминание, печать и перенос в другие программы) серии экспериментов.

Практическое занятие №2.

Изучение программирования под стандартом MPI

1. Обмены типа "точка-точка".

MPI. Терминология и обозначения; Общие процедуры MPI (MPI_Init, MPI_Finalize, MPI_Comm_size, MPI_Comm_rank, MPI_Wtime); Прием/передача сообщений между отдельными процессами (MPI_Send, MPI_Recv и др.); Объединение запросов на взаимодействие (MPI_Send_init, MPI_Recv_init, MPI_Startall); Совмещенные прием/передача сообщений (MPI_Sendrecv);

2. Коллективные обмены.

Функция широковещательной рассылки сообщения процессам - MPI_Bcast; Функция для сбора данных со всех процессов - MPI_Gather; Функция рассылки порции данных всем процессам - MPI_Scatter (обратная функция к MPI_Gather);

3. Прием/передача сообщений без блокировки. Различные виды операторов Wait и Test

4. Отложенные взаимодействия. Предотвращение тупиковых ситуаций, совмещенные прием и передача сообщений. Барьерная синхронизация

5. Примеры параллельных программ умножения матрицы на вектор. Практическое освоение программирования параллельных процессов, исполняющихся на декартовой топологии связей и топологии "граф". Освоение функций задания декартовой топологии и топологии "граф". Освоение методов распараллеливания алгоритмов решения задач, таких как умножение матрицы на вектор. Этот алгоритм является, в свою очередь, макро операцией в итерационных задачах. В лабораторной работе приведены три примера. Один простой пример связан с взаимодействиями параллельных процессов на декартовых структурах, два других примера связаны с разными способами параллельного умножения матрицы на вектор на топологии "кольцо" и "полный граф".

Практическое занятие №3.

Изучение программирования под стандартом OpenMP

1. OpenMP - модель параллелизма по управлению

Выполнение OpenMP-программы (Fork and Join Model). Модель памяти. Понятие консистентности памяти. Консистентность памяти в OpenMP (weak ordering). Классы переменных (клаузы SHARED, PRIVATE; директива THREADPRIVATE).

2. Основные понятия.

Директивы и клаузы. Понятие структурного блока. Компиляция OpenMP-программы. Параллельная область (директива PARALLEL). Понятие задачи (директива TASK).

3. Конструкции распределения работы

Распределение витков циклов. Циклы с зависимостью по данным. Организация конвейерного выполнения для циклов с зависимостью по данным. Распределение нескольких структурных блоков между нитями (директива SECTION). Редукционные операторы (клауза REDUCTION). Выполнение структурного блока одной нитью (директива SINGLE). Распределение операторов одного структурного блока между нитями (директива WORKSHARE). Копирование значений частных переменных (клаузы FIRSTPRIVATE, LASTPRIVATE, COPYIN, COPYPRIVATE).

4. онструкции для синхронизации нитей

Директивы MASTER, CRITICAL, BARRIER, TASKWAIT, ATOMIC, FLUSH, ORDERED.

5. Система поддержки выполнения OpenMP-программ. Переменные окружения, управляющие выполнением OpenMP-программы. Задание/опрос количества нитей, выполняющих параллельную область. Определение номера нити. Внутренние переменные, управляющие выполнением OpenMP-программы (ICV-Internal Control Variables). Задание/опрос значений ICV-переменных. Функции для синхронизации нитей (семафоры). Функции работы со временем.

6. Вложенный параллелизм

Создание вложенной параллельной области. Распределение витков тесно-вложенных циклов (с использованием вложенных параллельных областей и клаузы COLLAPSE). Задание количества нитей для вложенной параллельной области. Задание/опрос максимального уровня вложенности для параллельных областей. Задание максимального количества создаваемых нитей (переменная OMP_THREAD_LIMIT).

7. Отладка эффективности OpenMP-программ

Стратегии распределения витков цикла между нитями (клауза SCHEDULE). Отмена барьерной синхронизации по окончании выполнения цикла (клауза NOWAIT). Локализация данных. Задание поведения нитей во время ожидания (переменная OMP_WAIT_POLICY). Оптимизация OpenMP-программы при помощи Intel Thread Profiler (Intel Parallel Studio).

Библиографический список

1. Катаев М.Ю. Конспект лекций по дисциплине «Параллельное Программирование».
2. 1. Воеводин В.В. Математические модели и методы в параллельных процессах. М., 1986. – 296 с.
2. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб., 2002. – 608 с.