



Автономная некоммерческая организация
дополнительного профессионального образования
(АНО ДПО «Инфосфера»)

Центр профессиональной подготовки
ИНСТИТУТ ПРОГРАММНЫХ СИСТЕМ

**Рабочая программа дисциплины
«Параллельное программирование»**

Разработал:
преподаватель ИПС
АНО ДПО «Инфосфера»
В. Кузнецов

Йошкар-Ола, 2017

Пояснительная записка

Потребность решения сложных прикладных задач с большим объемом вычислений и принципиальная ограниченность максимального быстродействия "классических" - по схеме фон Неймана - ЭВМ привели к появлению многопроцессорных вычислительных систем (МВС). Использование таких средств вычислительной техники позволяет существенно увеличивать производительность ЭВМ при любом существующем уровне развития компьютерного оборудования. При этом, однако, необходимо "параллельное" обобщение традиционной - последовательной - технологии решения задач на ЭВМ. Так, численные методы в случае МВС должны проектироваться как системы параллельных и взаимодействующих между собой процессов, допускающих исполнение на независимых процессорах. Применяемые алгоритмические языки и системное программное обеспечение должны обеспечивать создание параллельных программ, организовывать синхронизацию и взаимоисключение асинхронных процессов и т.п.

Предметом рассмотрения настоящего курса и является изучение перечисленного круга вопросов. **Цель курса состоит** в изложении математических моделей и методов параллельного программирования для многопроцессорных вычислительных систем.

Задачи изучения дисциплины:

Использование многопроцессорных вычислительных систем предполагает практическое освоение следующих разделов параллельного программирования:

- ✓ Использование многопроцессорных вычислительных систем предполагает практическое освоение следующих разделов параллельного программирования:
- ✓ Архитектурные принципы реализации параллельной обработки в вычислительных машинах;
- ✓ Методы и языковые механизмы конструирования параллельных программ;
- ✓ Параллельные вычислительные методы.
- ✓ Изучение перечисленных тем достигается наличием в настоящем курсе следующего набора разделов:
- ✓ Цели и задачи параллельной обработки данных.
- ✓ Принципы построения параллельных вычислительных систем.
- ✓ Моделирование и анализ параллельных вычислений.
- ✓ Принципы разработки параллельных алгоритмов и программ.
- ✓ Системы разработки параллельных программ.
- ✓ Параллельные численные алгоритмы для решения типовых задач вычислительной математики.

Планируемые результаты обучения

Осуществляется предварительная самостоятельная или под руководством разработка алгоритмов с использованием графических средств (блок-схемы, UML-диаграммы и др.). Не требуется взаимодействие с другими программистами, системным аналитиком и архитектором программного обеспечения. Осуществляется решение типовых задач. Полученные результаты представляются руководителю разработки программного обеспечения.

Выполняются самостоятельная разработка процедур сборки модулей и компонент программного обеспечения и верификация выпусков программного продукта. Производится разработка процедур развертывания и обновления программного обеспечения, процедур миграции и преобразования (конвертации) данных и программных интерфейсов с использованием выбранных программных средств, технологий создания открытых систем. Осуществляется решение различных типов задач проектирования программных комплексов различной сложности, выбор способов

реализации взаимодействия программных компонент/модулей. Требуется взаимодействие с программистами-разработчиками модулей, архитектором программного обеспечения. Полученные результаты представляются руководителю разработки программного обеспечения.

В процессе интеграции программных модулей и компонент и верификации выпусков программного продукта осуществляется сборка модулей и компонент программного обеспечения, производится интеграция с внешней средой. Обеспечивается согласованное функционирование и требуемый уровень качества.

Проведение интеграции программных модулей и компонент и верификация выпусков программного продукта предполагают определение задач программной интеграции, распределение задач между подчиненными, обеспечение взаимодействия подчиненных сотрудников.

Программист несет ответственность за результат выполнения работ на уровне группы программистов. В процессе интеграции требуется взаимодействие с архитектором программного обеспечения. Полученные результаты представляются руководителю разработки программного обеспечения.

Учебно-тематический план

				Лекции	Лабораторные занятия
1	2	3	4	5	6
I	1	Цели и задачи введения параллельной обработки данных	10	4	2
	2	Принципы построения параллельных вычислительных систем	8	8	
	3	Моделирование и анализ параллельных вычислений	30	6	8
	4	Принципы разработки параллельных алгоритмов и программ	6	6	
	5	Параллельные численные алгоритмы для решения типовых задач вычислительной математики	66	8	22
		ИТОГО:	120	32	32

Содержание курса

Цели и задачи введения параллельной обработки данных

1.1. Необходимость. Ограничение максимальной производительности однопроцессорных ЭВМ. Постоянная необходимость решения задач, превышающих возможности современных ЭВМ (проблемы "большого вызова"). Необходимость коллективного режима решения задач.

Автоматизация управления распределенных технических систем. Технические требования по снижению стоимости и повышению надежности.

1.2. История введения параллелизма (ENIAC, IBM-701, 704, 709, ATLAS, CDC 6600, 7600, ILLIAC IV, Cray-1, Эльбрус)

1.3. Различие многозадачных, параллельных и распределенных вычислений

1.4. Проблемы использования параллелизма. Существование последовательных алгоритмов (закон Амдаля). Повышение производительности последовательных компьютеров (закон Мура). Потери на взаимодействие и передачу данных (гипотеза Минского). Высокая стоимость параллельных систем (закон Гроша). "Последовательность" существующих алгоритмов и программного обеспечения. Зависимость эффективности параллельных вычислений от учета особенностей аппаратуры. Сложность разработки параллельных алгоритмов. Трудоемкость проверки правильности параллельных программ.

Принципы построения параллельных вычислительных систем

2.1. Пути достижения параллелизма. Функциональные вычислительные устройства. Многоуровневая и модульная память. Конвейерные и векторные вычисления. Процессорные матрицы. Многопроцессорные вычислительные системы с общей и распределенной памятью (мультипроцессоры и мультикомпьютеры). Микропроцессорные системы.

2.2. Способы построения многопроцессорных вычислительных систем. Схемы коммутации (полная коммутация - общая память, перекрестные коммутаторы, локальные схемы коммутации - общая шина, решетки, кластеры). Анализ параллельных алгоритмов и типовые топологии схем коммутации - кольцо, линейка, решетки, полный граф, гиперкуб, тор, дерево. Аппаратная реализация и программная эмуляция топологий.

2.3. Виды параллельных вычислительных систем. СуперЭВМ. Многопроцессорные вычислительные комплексы (МВС). Многомашинные вычислительные комплексы. Сети ЭВМ. Примеры современных высокопроизводительных вычислительных систем (Cray T932, IBM SP2, HP Exemplar, ASCI RED). Суперкомпьютерные вычислительные системы в России.

2.4. Классификация МВС. Систематики Флинна и Шора. Потоки данных (команд). Структурная нотация Хокни и Джесхоупа.

2.5. Оценка производительности МВС. Общее выражение для оценки производительности для разного типа МВС. Максимальная (пиковая) производительность. Степень параллелизма (длина полупроизводительности). Удельная производительность. Значения показателей для ряда МВС.

Моделирование и анализ параллельных вычислений

3.1. Модели параллельных вычислительных систем. Компьютер с неограниченным параллелизмом (паракомпьютер). Модели многопроцессорных систем с общей и распределенной памятью. Модель конвейерной системы.

3.2. Модель алгоритма в виде графа "операнд - операции". Представление алгоритма в виде графа потока данных. Расписание параллельных вычислений. Показатель временной сложности алгоритма. Оценка времени выполнения алгоритма для паракомпьютера (предельное распараллеливание) и для систем с конечным количеством процессоров. Зависимость оценок от

топологии графа алгоритма и необходимость оптимизации структуры графа. Способы получения оптимального расписания вычислений.

3.3. Модель параллельных вычислений в виде сети Петри. Основные понятия теории сетей Петри. Использование сетей Петри для описания параллельных вычислений. Демонстрация основных проблем параллельных вычислений: синхронизация, взаимное исключение, блокировка (тупики).

3.4. Модель параллельных вычислений в виде графа "процесс-ресурс". Понятие процесса. Синхронизация параллельных процессов. Аппарат событий. Пример реализации в операционной системе Unix. Взаимное исключение параллельных процессов. Концепция ресурса. Механизмы взаимного исключения: алгоритм Деккера, семафоры (Дейкстра), мониторы (Вирт). Примеры решения стандартных задач взаимного исключения: кольцевой буфер, проблема "читатели и писатели". Взаимодействие параллельных процессов посредством механизма передачи сообщений. Механизмы передачи: очереди, почтовые ящики, порты. Принцип рандеву в языках Ада и ОККАМ.

Проблемы взаимодействия процессов. Понятие тупика и условия его возникновения. Предотвращение тупиков. Алгоритм банкира. Обнаружение тупиков и восстановление состояния процессов. Многозадачный режим работы ЭВМ как частный случай параллельной обработки.

Принципы разработки параллельных алгоритмов и программ.

4.1. Оценка эффективности параллельных вычислений. Показатель эффекта распараллеливания (ускорение). Эффективность использования вычислительной системы. Способы оценки показателей. Основные характеристики вычислительной системы, влияющие на величины ускорения и эффективности (архитектура, количество процессоров, топология каналов передачи данных).

4.2. Оценка коммуникационной трудоемкости параллельных алгоритмов. Характеристики топологий сети передачи данных. Алгоритмы маршрутизации. Методы передачи данных. Анализ трудоемкости основных операций передачи данных. Передача данных между двумя процессорами сети. Одиночная и множественная рассылка сообщений. Операция циклического сдвига. Методы логического представления топологии коммуникационной среды. Отображение кольцевой топологии и топологии решетки на гиперкуб.

4.3. Уровни распараллеливания вычислений. Распараллеливание вычислений на уровне команд, выражений, программных модулей, отдельно выполняемых заданий.

4.4. Этапы построения параллельных алгоритмов и программ.

Выбор параллельного алгоритма. Реализация алгоритма в виде параллельной программы. Построение исполняемой программы для параллельной вычислительной системы. Параллельное исполнение машинной программы. Частные постановки: выбор оптимального алгоритма для конкретной вычислительной системы, нахождение наилучшей топологии вычислительной системы для решения определенной задачи, распараллеливание существующего алгоритма.

4.5. Технологические аспекты распараллеливания. Декомпозиция алгоритма на параллельно исполняемые фрагменты вычислений. Распределение заданий по процессорам и балансировка. Синхронизация и взаимное исключение. Организация взаимодействия.

Параллельные численные алгоритмы для решения типовых задач вычислительной математики.

- 5.1. Общие способы распараллеливания алгоритмов. Выявление функциональной независимости отдельных фрагментов алгоритма (параллелизм команд). Геометрическое разделение вычислений (параллелизм данных). Иерархическая декомпозиция обработки данных.
- 5.2. Организация параллельного исполнения рекурсивных вычислений. Проблема рекурсивной зависимости этапов обработки данных. Каскадная схема. Подход для получения асимптотически ненулевой эффективности. Метод Оутса. Пример для вычисления частичных и общей сумм.
- 5.3. Параллельные численные алгоритмы линейной алгебры. Способы разбиения матриц (горизонтальная, вертикальная, блочные схемы). Методы вычисления произведения матриц с использованием разных схем разбиения матриц. Обеспечение предельно допустимого параллелизма. Обращение матриц. Параллельные методы решения систем линейных уравнений.
- 5.4. Параллельные численные алгоритмы решения дифференциальных уравнений в частных производных. Параллельная реализация прямых и итерационных методов решения дифференциальных уравнений в частных производных. Анализ разностных схем для эффективного разделения области определения решаемых задач.
- 5.5. Параллельные численные алгоритмы многомерной многоэкстремальной оптимизации. Характеристическая схема представления методов глобального поиска. Общий принцип распараллеливания методов. Оценка эффективности введения параллелизма: эффективность и безызбыточность. Синхронные и асинхронные варианты алгоритмов. Определение наилучших топологий вычислительной системы для реализации методов.

Основная литература

1. Гергель В.П., Стронгин Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных систем. - Н.Новгород, ННГУ, 2001.
2. Богачев К.Ю. Основы параллельного программирования. - М.: БИНОМ. Лаборатория знаний, 2003.
3. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. - СПб.: БХВ-Петербург, 2002.
4. Немнюгин С., Стесик О. Параллельное программирование для много-процессорных вычислительных систем - СПб.: БХВ-Петербург, 2002.

Дополнительная литература

5. Березин И.С., Жидков И.П. Методы вычислений. - М.: Наука, 1966.
6. Дейтел Г. Введение в операционные системы. Т.1.- М.: Мир, 1987.
7. Кнут Д. Искусство программирования для ЭВМ. Т. 3. Сортировка и по-иск. - М.: Мир, 1981.
8. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. - М.: МЦНТО, 1999.
9. Корнеев В.В.. Параллельные вычислительные системы. - М.: Нолидж, 1999.
10. Корнеев В.В. Параллельное программирование в MPI. Москва-Ижевск: Институт компьютерных исследований, 2003.

Вопросы к зачету

1. Цели и задачи введения параллельной обработки данных.
2. Существование последовательных алгоритмов (закон Амдаля).
3. Повышение производительности последовательных компьютеров (закон Мура).
4. Потери на взаимодействие и передачу данных (гипотеза Минского).
5. Высокая стоимость параллельных систем (закон Гроша).
6. "Последовательность" существующих алгоритмов и программного обеспечения.
7. Зависимость эффективности параллельных вычислений от учета особенностей аппаратуры.
8. Сложность разработки параллельных алгоритмов.
9. Трудоемкость проверки правильности параллельных программ.
10. Принципы построения параллельных вычислительных систем.
11. Функциональные вычислительные устройства.
12. Многоуровневая и модульная память.
13. Конвейерные и векторные вычисления.
14. Процессорные матрицы.
15. Многопроцессорные вычислительные системы с общей и распределенной памятью (мультипроцессоры и мультикомпьютеры).
16. Микропроцессорные системы.
17. Схемы коммутации (полная коммутация - общая память, перекрестные коммутаторы, локальные схемы коммутации - общая шина, решетки, кластеры).
18. Анализ параллельных алгоритмов и типовые топологии схем коммутации - кольцо, линейка, решетки, полный граф, гиперкуб, тор, дерево.
19. Аппаратная реализация и программная эмуляция топологий.
20. СуперЭВМ.
21. Многопроцессорные вычислительные комплексы (МВС).
22. Многомашинные вычислительные комплексы.
23. Сети ЭВМ.
24. Классификация МВС.
25. Систематики Флинна и Шора. Потоки данных (команд).
26. Структурная нотация Хокни и Джесхоупа.
27. Оценка производительности МВС.
28. Степень параллелизма (длина полупроизводительности). Удельная производительность. Значения показателей для ряда МВС.
29. Компьютер с неограниченным параллелизмом (паракомпьютер). Модели многопроцессорных систем с общей и распределенной памятью. Модель конвейерной системы.
30. Модель алгоритма в виде графа "операнд - операции".
31. Оценка времени выполнения алгоритма для паракомпьютера (предельное распараллеливание) и для систем с конечным количеством процессоров. Зависимость оценок от топологии графа алгоритма и необходимость оптимизации структуры графа.
32. Способы получения оптимального расписания вычислений.
33. Модель параллельных вычислений в виде сети Петри.
34. Модель параллельных вычислений в виде графа "процесс-ресурс".
35. Понятие процесса. Синхронизация параллельных процессов. Аппарат событий.

36. Взаимоисключение параллельных процессов. Концепция ресурса.
37. Механизмы взаимоисключения: алгоритм Деккера, семафоры (Дейкстра), мониторы (Вирт).
38. Взаимодействие параллельных процессов посредством механизма передачи сообщений. Механизмы передачи: очереди, почтовые ящики, порты.
39. Принцип рандеву в языках Ада и ОККАМ.
40. Проблемы взаимодействия процессов. Понятие тупика и условия его возникновения. Предотвращение тупиков. Алгоритм банкира. Обнаружение тупиков и восстановление состояния процессов.
41. Многозадачный режим работы ЭВМ как частный случай параллельной обработки.
42. Принципы разработки параллельных алгоритмов и программ
43. Оценка эффективности параллельных вычислений
44. Показатель эффекта распараллеливания (ускорение). Эффективность использования вычислительной системы. Способы оценки показателей. Основные характеристики вычислительной системы, влияющие на величины ускорения и эффективности (архитектура, количество процессоров, топология каналов передачи данных).
45. Оценка коммуникационной трудоемкости параллельных алгоритмов
46. Характеристики топологий сети передачи данных. Алгоритмы маршрутизации. Методы передачи данных.
47. Анализ трудоемкости основных операций передачи данных. Передача данных между двумя процессорами сети. Одиночная и множественная рассылка сообщений. Операция циклического сдвига.
48. Методы логического представления топологии коммуникационной среды. Отображение кольцевой топологии и топологии решетки на гиперкуб.
49. Этапы построения параллельных алгоритмов и программ
50. Технологические аспекты распараллеливания
51. Система MPI. Общая характеристика. Поддержка модели взаимодействия параллельных вычислителей при помощи передачи сообщений.
52. Организация вычислений на многопроцессорной системе Parsytec PowerXplorer. Использование технологии кросс эмуляции при разработке параллельных программ. Концепция виртуальных процессоров и каналов передачи данных. Виртуальные топологии системы: кольцо, линейка, звезда, решетка, дерево. Основные программные примитивы. Пример использования.
53. Параллельные численные алгоритмы для решения типовых задач вычислительной математики
54. Общие способы распараллеливания алгоритмов
55. Организация параллельного исполнения рекурсивных вычислений
56. Параллельные численные алгоритмы линейной алгебры
57. Способы разбиения матриц (горизонтальная, вертикальная, блочные схемы). Методы вычисления произведения матриц с использованием разных схем разбиения матриц. Обеспечение предельно допустимого параллелизма. Обращение матриц. Параллельные методы решения систем линейных уравнений.
58. Параллельные численные алгоритмы решения дифференциальных уравнений в частных производных

59. Параллельная реализация прямых и итерационных методов решения дифференциальных уравнений в частных производных. Анализ разностных схем для эффективного разделения области определения решаемых задач.
60. Параллельные численные алгоритмы многомерной многоэкстремальной оптимизации