

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

КАФЕДРА «Робототехника и мехатроника»

**Практикум**

по выполнению практических работ

по дисциплине

«Программные и аппаратные средства ИИ в робототехнике и мехатронике»

Ростов-на-Дону

2024

Составители: доцент Дудкин Д.В.

Практикум по выполнению практических работы по дисциплине «Программные и аппаратные средства ИИ в робототехнике и мехатронике». ДГТУ, г. Ростов-на-Дону, 2024 г.

В практикуме кратко изложены теоретические вопросы, необходимые для успешного выполнения лабораторной работы, рабочее задание и контрольные вопросы для самопроверки.

Предназначено для обучающихся по направлению подготовки (код, название):

09.04.01 Информатика и вычислительная техника

Ответственный за выпуск:

Зав. кафедрой (руководитель структурного подразделения, ответственного за реализацию ОПОП): Изюмов Андрей Игоревич

© Издательский центр ДГТУ, 2023г.

**Практическая работа №1 «Ярусно-параллельные формы представления граф-схем алгоритмов и определение их характеристик»**

**Инструкция по технике безопасности при выполнении лабораторной работы**

При выполнении лабораторной работы необходимо соблюдать правила техники безопасности при работе с электрооборудованием, включение электрооборудования, в том числе компьютеров, производить только после разрешения преподавателя или сопровождающего инженера.

**Цель работы**

Изучить методы представления алгоритмов в виде ярусно-параллельных форм (ЯПФ) и научиться определять основные характеристики графов.

**Материально-техническое обеспечение работы**

1. Компьютер с программным обеспечением для создания графов (например, Microsoft Visio, Graphviz).
2. Интернет для доступа к теоретическим материалам и примерам графов.

**Порядок выполнения работы**

**Введение в граф-схемы алгоритмов:**

* Изучите, что такое граф-схема алгоритмов (ГСА) и её основные элементы: вершины, рёбра, пути.
* Ознакомьтесь с понятием ярусно-параллельных форм (ЯПФ), где вершины разбиваются на уровни (ярусы), и каждая вершина на одном ярусе зависит от вершин предыдущего.

**Построение ЯПФ:**

* Получите описание алгоритма (например, вычислительный алгоритм или процесс управления).
* Постройте граф-схему алгоритма, разложив его на ярусы с учётом зависимостей между операциями.
* Определите ширину каждого яруса — максимальное количество операций, которые могут выполняться параллельно на одном уровне.

**Определение характеристик ЯПФ:**

* Вычислите следующие характеристики:
  + **Глубина графа:** количество ярусов, через которые проходит самый длинный путь.
  + **Ширина ярусов:** количество вершин на каждом уровне.
  + **Время выполнения алгоритма:** для оценки времени выполнения используйте ширину и глубину графа.

**Анализ и представление результатов:**

* Постройте граф-схему с выделенными ярусами и посчитайте все перечисленные характеристики.
* Оцените эффективность представления алгоритма в виде ЯПФ.

**Контрольные вопросы**

1. Что такое ярусно-параллельная форма граф-схемы алгоритмов?
2. Какие основные характеристики графа можно определить?
3. Как влияют ширина и глубина графа на эффективность выполнения алгоритма?

**Практическая работа №2 «Определение критического пути ЯПФ»**

**Инструкция по технике безопасности при выполнении лабораторной работы**

При выполнении лабораторной работы необходимо соблюдать правила техники безопасности при работе с электрооборудованием, включение электрооборудования, в том числе компьютеров, производить только после разрешения преподавателя или сопровождающего инженера.

**Цель работы**

Научиться определять критический путь в ярусно-параллельной форме граф-схем алгоритмов и понимать его влияние на выполнение алгоритма.

**Материально-техническое обеспечение работы**

1. Программное обеспечение для построения графов и анализа путей (например, Graphviz, Microsoft Visio).
2. Программа для выполнения расчетов и анализа (например, MS Excel).

**Порядок выполнения работы**

1. **Ознакомление с критическим путем:**
   * Изучите, что такое критический путь в графе — самый длинный путь от начальной до конечной вершины.
   * Ознакомьтесь с ролью критического пути в параллельных вычислениях и планировании выполнения операций.
2. **Построение графа и определение критического пути:**
   * Постройте ярусно-параллельную форму графа для заданного алгоритма.
   * Используйте метод критического пути для определения самого длинного пути в графе.
   * Подсчитайте время выполнения алгоритма вдоль критического пути.
3. **Анализ полученных данных:**
   * Определите, какие зависимости между операциями приводят к удлинению критического пути.
   * Рассмотрите варианты оптимизации графа для сокращения длины критического пути.
4. **Оформление отчета:**
   * Отразите на графе критический путь и проведите анализ влияния его длины на общее время выполнения алгоритма.

**Материально-техническое обеспечение работы:**

1. Программное обеспечение для построения графов и анализа путей (например, Graphviz, Microsoft Visio).
2. Программа для выполнения расчетов и анализа (например, MS Excel).

**Контрольные вопросы**

1. Что такое критический путь в ярусно-параллельной форме графа?
2. Как определить критический путь в графе?
3. Как критический путь влияет на общее время выполнения алгоритма?

**Практическая работа №3 «Распределение загрузки в случае, когда количество разнотипных процессоров больше или равно ширине яруса»**

**Инструкция по технике безопасности при выполнении лабораторной работы**

При выполнении лабораторной работы необходимо соблюдать правила техники безопасности при работе с электрооборудованием, включение электрооборудования, в том числе компьютеров, производить только после разрешения преподавателя или сопровождающего инженера.

**Цель работы**

Изучить методы распределения задач между разнотипными процессорами при выполнении параллельных вычислений с использованием ЯПФ.

**Материально-техническое обеспечение работы**

Компьютер с программным обеспечением для моделирования параллельных процессов (например, MATLAB, AnyLogic).

Инструменты для расчетов и анализа данных (MS Excel).

**Порядок выполнения работы**

**Ознакомление с параллельными вычислениями:**

* Изучите основные принципы распределения задач на параллельные процессоры.
* Рассмотрите особенности выполнения задач на разнотипных процессорах (разные тактовые частоты, объем памяти, мощность вычислений).

**Построение ЯПФ и распределение задач:**

* Постройте ярусно-параллельную форму для заданного алгоритма.
* Определите, каким образом можно распределить задачи между процессорами при условии, что количество процессоров больше или равно ширине яруса.
* Учитывайте особенности каждого процессора и соответствие задач.

**Оценка эффективности распределения:**

* Проведите расчет времени выполнения алгоритма при предложенном распределении.
* Сравните эффективность использования разнотипных процессоров.

**Подготовка отчета:**

* Представьте результаты распределения и анализа.
* Проанализируйте, как оптимизировать выполнение задач для увеличения скорости выполнения алгоритма.

**Контрольные вопросы**

**Практическая работа №4 «Распределение загрузки для однотипных процессоров, количество которых меньше ширины яруса»**

**Инструкция по технике безопасности при выполнении лабораторной работы**

При выполнении лабораторной работы необходимо соблюдать правила техники безопасности при работе с электрооборудованием, включение электрооборудования, в том числе компьютеров, производить только после разрешения преподавателя или сопровождающего инженера.

**Цель работы**

Научиться распределять вычислительную нагрузку между процессорами в системе с несколькими процессорами, число которых меньше требуемой вычислительной ширины яруса, используя интеллектуальные методы оптимизации и балансировки загрузки.

**Материально-техническое обеспечение работы**

1. Компьютер с установленной средой разработки (Python, MATLAB, или другие языки, поддерживающие многопоточность).
2. Программное обеспечение для симуляции и визуализации работы многопроцессорных систем (например, AnyLogic, SimPy).
3. Средства для создания отчетов и графиков (MS Excel, Matplotlib в Python).

**Порядок выполнения работы**

**Ознакомление с исходными данными:**

* Определите архитектуру многопроцессорной системы, где процессоры однотипные, но их количество ограничено и меньше необходимой ширины яруса для выполнения всех задач.
* Исходные данные должны содержать:
  + Количество процессоров.
  + Количество задач и их сложность (время выполнения).
  + Структуру задач, которая определяет, какие задачи могут выполняться параллельно (зависимости между задачами).

**Определение методов распределения:**

* Изучите методы распределения вычислительной нагрузки для параллельных систем. Наиболее подходящими методами являются:
  + **Метод очередей**: задачи помещаются в очередь и распределяются по мере освобождения процессоров.
  + **Метод минимальной загрузки**: задачи распределяются так, чтобы минимизировать максимальную загрузку процессора (load balancing).
  + **Динамическое распределение**: задачи распределяются в зависимости от текущей загрузки процессоров.

**Разработка алгоритма распределения задач:**

* Создайте алгоритм, который будет распределять задачи между процессорами в условиях ограниченного количества ресурсов:
  + Для этого можно использовать алгоритмы жадного поиска (greedy algorithms), которые распределяют задачи на основе текущей загрузки процессоров.
  + Если задача занимает больше одного такта, алгоритм должен учитывать возможности параллельного выполнения задач.
* Рассмотрите стратегию с учетом зависимостей между задачами. Если задача B зависит от задачи A, она не может быть выполнена, пока A не завершится.

**Моделирование и симуляция:**

* Реализуйте алгоритм в среде моделирования (например, Python, MATLAB или другой язык программирования с поддержкой многопоточности):
  + Создайте задачу для системы, где количество процессоров меньше ширины яруса.
  + Запустите симуляцию выполнения задач на процессорах, используя разработанный алгоритм.
* Зафиксируйте время выполнения всех задач в зависимости от метода распределения.

**Анализ результатов:**

* Сравните результаты, полученные при разных подходах к распределению задач:
  + Общее время выполнения всех задач (makespan).
  + Максимальная загрузка на один процессор.
  + Среднее время выполнения задачи на процессор.
* Проанализируйте, какой метод распределения оказался наиболее эффективным в условиях недостатка ресурсов.

**Оптимизация распределения:**

* Изучите возможности оптимизации алгоритма:
  + Возможно ли уменьшить время выполнения за счет перераспределения задач?
  + Как повлияет на работу системы использование другой стратегии балансировки загрузки?

**Подготовка отчета:**

* Составьте отчет, в который войдут:
  + Описание системы и задач.
  + Описание разработанного алгоритма распределения задач.
  + Результаты моделирования (графики, таблицы, выводы).
  + Рекомендации по улучшению работы алгоритма.
* Включите в отчет скриншоты с процессом симуляции и графиками загрузки процессоров.

**Контрольные вопросы**

1. Какие основные стратегии можно использовать для распределения вычислительной нагрузки в многопроцессорных системах?
2. В чем заключается проблема "узкого места" в системах с недостаточным количеством процессоров?
3. Как влияют зависимости между задачами на их выполнение в параллельных системах?
4. Каким образом можно улучшить распределение задач для минимизации общего времени выполнения?
5. Какие алгоритмы подходят для динамического распределения задач в условиях ограниченных ресурсов?