

Практическая работа №2

Тема: Параллельная реализация простых алгоритмов сортировки (пузырьком, выбором, вставкой) на CPU с использованием OpenMP

Цель работы

- Освоить параллельное программирование с использованием OpenMP на CPU.
- Изучить и реализовать базовые алгоритмы сортировки (пузырьком, выбором, вставкой) в параллельном варианте.
- Провести исследование производительности и сравнить параллельные и последовательные реализации.

Теоретическая часть

1. Введение в алгоритмы сортировки

Алгоритмы сортировки — это базовые алгоритмы в информатике, которые используются для упорядочивания элементов. Некоторые из них, такие как сортировка пузырьком, выбором и вставкой, просты для реализации, но менее эффективны для больших объемов данных.

1. Сортировка пузырьком (Bubble Sort):

- а. Принцип работы: сравнивает соседние элементы и перемещает больший элемент вправо.
- б. Время выполнения: $O(n^2)$.
- в. Подходит для небольших массивов или частично отсортированных данных.

2. Сортировка выбором (Selection Sort):

- а. Принцип работы: находит минимальный элемент и перемещает его в начало массива.
- б. Время выполнения: $O(n^2)$.
- в. Меньше перемещений по сравнению с сортировкой пузырьком, но также менее эффективна на больших массивах.

3. Сортировка вставкой (Insertion Sort):

- а. Принцип работы: элементы последовательно перемещаются в отсортированную часть массива.
- б. Время выполнения: $O(n^2)$, но может быть эффективной на небольших или частично отсортированных массивах.

2. Параллельное программирование с OpenMP

OpenMP — это API для параллельного программирования, который позволяет распределить задачи между несколькими потоками на CPU. Для параллельной сортировки мы будем использовать директивы OpenMP, чтобы ускорить выполнение алгоритмов.

1. **Директива `#pragma omp parallel for`:** позволяет распараллелить выполнение цикла, разделяя итерации между потоками.
2. **`reduction`:** позволяет суммировать значения из разных потоков в одну переменную.
3. **Особенности использования:** некоторые алгоритмы, такие как сортировка вставкой, могут быть трудны для эффективного параллелизма, так как их выполнение зависит от последовательного доступа к данным.

3. Параллельная реализация сортировок с OpenMP

- **Сортировка пузырьком:** в параллельной реализации можно параллелить внешние циклы, но важно избегать изменений в позициях элементов, которые будут влиять на другие итерации.
- **Сортировка выбором:** каждый поток может искать минимальный элемент параллельно.
- **Сортировка вставкой:** алгоритм менее подходит для параллельной реализации из-за зависимости текущего шага от предыдущих.

Практическая часть

Задание

1. **Реализация сортировок без параллелизма:**

Напишите функции для сортировки пузырьком, выбором и вставкой без использования OpenMP.

2. **Параллельная реализация с использованием OpenMP:**

Используйте директивы OpenMP для распараллеливания внешних циклов. Протестируйте производительность каждой сортировки на массивах разного размера (например, 1000, 10,000 и 100,000 элементов).

3. **Сравнение производительности:**

Измерьте время выполнения последовательных и параллельных версий каждой сортировки, используя библиотеку `<chrono>`.
Сравните результаты и сделайте выводы.

Контрольные вопросы

1. В чём основные отличия алгоритмов сортировки пузырьком, выбором и вставкой?
2. Почему параллельная реализация сортировки вставкой сложнее для выполнения с использованием OpenMP?
3. Какие директивы OpenMP были использованы для параллельной реализации алгоритмов?

4. Какие преимущества и недостатки параллельной реализации алгоритмов сортировки на CPU?
5. Как можно измерить производительность программы в C++?
6. Как изменяется производительность сортировок при увеличении числа потоков?
7. В каких ситуациях параллельная сортировка может быть менее эффективной, чем последовательная?