Практическая работа №8: Разработка гибридного приложения для CPU и GPU

Цель работы:

Разработка программы, которая распределяет вычисления между CPU и GPU для обработки массива данных. Изучение принципов гибридных вычислений, оптимизации передачи данных и балансировки нагрузки между CPU и GPU.

Задачи:

1. Реализовать обработку массива данных на CPU с использованием OpenMP.
2. Реализовать обработку массива данных на GPU с использованием CUDA.
3. Организовать передачу данных между CPU и GPU.
4. Провести анализ производительности гибридного приложения.

Теоретическая часть:

1. Гибридные вычисления:
   - Гибридные вычисления — это подход, при котором задачи распределяются между CPU и GPU для достижения максимальной производительности.
   - CPU используется для последовательных задач и управления данными, а GPU — для параллельных вычислений.

2. OpenMP:
   - OpenMP — это API для многопоточного программирования на CPU. Позволяет легко распараллеливать циклы и задачи.

3. CUDA:
   - CUDA — это платформа для параллельных вычислений на GPU. Позволяет использовать тысячи ядер GPU для выполнения параллельных задач.

4. Передача данных между CPU и GPU:
   - Данные передаются между CPU и GPU через шину PCI Express. Это может стать узким местом в производительности, поэтому важно минимизировать количество передач.

Практическая часть:

Задание 1: Реализация обработки массива на CPU с использованием OpenMP

1. Создайте массив данных размером `N` (например, `N = 1 000 000`).
2. Реализуйте функцию для обработки массива на CPU с использованием OpenMP. Например, умножьте каждый элемент массива на 2.
3. Замерьте время выполнения обработки на CPU.

Задание 2: Реализация обработки массива на GPU с использованием CUDA

1. Скопируйте массив данных на GPU.
2. Реализуйте ядро CUDA для обработки массива на GPU. Например, умножьте каждый элемент массива на 2.
3. Скопируйте обработанные данные обратно на CPU.
4. Замерьте время выполнения обработки на GPU.

Задание 3: Гибридная обработка массива

1. Разделите массив на две части: первая половина обрабатывается на CPU, вторая — на GPU.
2. Реализуйте гибридное приложение, которое выполняет обработку массива на CPU и GPU одновременно.
3. Замерьте общее время выполнения гибридной обработки.

Задание 4: Анализ производительности

1. Сравните время выполнения обработки массива на CPU, GPU и в гибридном режиме.
2. Проведите анализ производительности и определите, в каких случаях гибридный подход дает наибольший выигрыш.

Результаты работы:

1. Отчет:
   - Описание выполненных заданий.
   - Исходный код программы.
   - Результаты замеров времени выполнения для CPU, GPU и гибридного режима.
   - Анализ производительности.

2. Выводы:
   - В каких случаях гибридный подход эффективен?
   - Какие факторы влияют на производительность гибридных вычислений?
   - Как можно оптимизировать передачу данных между CPU и GPU?

Контрольные вопросы:

1. Какие преимущества предоставляют гибридные вычисления?
2. Как минимизировать накладные расходы при передаче данных между CPU и GPU?
3. Какие задачи лучше выполнять на CPU, а какие — на GPU?
4. Как можно улучшить производительность гибридного приложения?

Дополнительные задания (по желанию):

1. Реализуйте обработку массива, где каждая часть массива обрабатывается разными операциями (например, сложение на CPU и умножение на GPU).
2. Проведите эксперименты с разными размерами массива и определите, при каких условиях гибридный подход наиболее эффективен.
3. Используйте инструменты профилирования (например, `nvprof` для CUDA) для анализа производительности программы.