Практическая работа №7: Редукция и сканирование на GPU

Цель работы

Изучение и реализация параллельных алгоритмов редукции и сканирования (префиксной суммы) на GPU с использованием CUDA. Исследование производительности этих алгоритмов и их оптимизация.

Задачи

1. Реализовать алгоритм редукции для вычисления суммы элементов массива на GPU.
2. Реализовать алгоритм сканирования (префиксной суммы) на GPU.
3. Провести анализ производительности реализованных алгоритмов.
4. Оптимизировать код с использованием различных типов памяти CUDA.

Теоретическая часть

1. Редукция

Редукция — это операция, которая сводит множество элементов к одному значению (например, сумма, минимум, максимум). На GPU редукция выполняется с использованием иерархии потоков и блоков.

2. Сканирование (префиксная сумма)

Сканирование — это операция, которая вычисляет накопленный результат для каждого элемента массива. Например, для массива [1, 2, 3, 4] префиксная сумма будет [1, 3, 6, 10].

3. Типы памяти в CUDA

- **Глобальная память**: Основная память GPU, доступная всем потокам.
- **Разделяемая память**: Быстрая память, доступная только потокам внутри одного блока.
- **Локальная память**: Память, выделенная для каждого потока.

Практическая часть

Задание 1: Реализация редукции

1. Напишите ядро CUDA для выполнения редукции (суммирования элементов массива).
2. Используйте разделяемую память для оптимизации доступа к данным.
3. Проверьте корректность работы на тестовом массиве.

Задание 2: Реализация префиксной суммы

1. Напишите ядро CUDA для выполнения префиксной суммы.

2. Используйте разделяемую память для оптимизации доступа к данным.
3. Проверьте корректность работы на тестовом массиве.

Задание 3: Анализ производительности

1. Замерьте время выполнения редукции и сканирования для массивов разного размера.
2. Сравните производительность с CPU-реализацией.
3. Проведите оптимизацию кода, используя различные типы памяти CUDA.

Отчёт по практической работе

Отчет должен содержать:

1. **Теоретическую часть**: Краткое описание редукции и сканирования, их применение.
2. **Практическую часть**:
    a. Исходный код реализованных алгоритмов.
    b. Результаты тестирования на различных данных.
    c. Графики производительности (время выполнения в зависимости от размера массива).
3. **Выводы**:
    a. Анализ результатов.
    b. Сравнение производительности CPU и GPU.
    c. Рекомендации по оптимизации.

Контрольные вопросы

1. В чём разница между редукцией и сканированием?
2. Какие типы памяти CUDA используются для оптимизации редукции и сканирования?
3. Как можно оптимизировать префиксную сумму на GPU?
4. Приведите пример задачи, где применяется сканирование.

Пример тестовых данных

Для тестирования можно использовать следующие массивы:

- Массив из 1024 случайных чисел.
- Массив из 1 000 000 случайных чисел.
- Массив из 10 000 000 случайных чисел.

Дополнительные задания

1. Реализуйте редукцию для нахождения минимума и максимума.
2. Реализуйте алгоритм Blelloch Scan для более эффективного сканирования.
3. Исследуйте влияние размера блока на производительность.