

Практическая работа №5

ст. преп. каф. ВпВ ИКИТ СФУ Тарасов С. А.

Цель работы

Закрепить навык работы с разделяемой памятью и примитивами синхронизации нитей. Освоить программирование уровня warp'a и технику широковещания регистров.

Задание

1. Разработать кёрнел `kernel_vecred_nobr` (1), который принимает экземпляр `VectorView` по значению и вычисляет сумму элементов вектора. Для хранения частичных сумм блоков использовать разделяемую память.
2. Разработать кёрнел `kernel_vecred_br` (2), который принимает экземпляр `VectorView` по значению и вычисляет сумму элементов вектора, используя функцию `__shfl_down_sync`. Для хранения частичных сумм warp'ов использовать разделяемую память.
3. Используя фреймворк `Google Test`, разработать модульные тесты для функций (1) и (2) с размерами векторов $n \in \{1, 2, 3, 127, 129, 512, 541, 1037\}$. В качестве эталона для сравнения использовать результат аналогичной операции для `Eigen::Matrix (sum)`; для верификации результатов применять макрос `EXPECT_NEAR` с абсолютной точностью 10^{-4} .
4. Используя фреймворк `Google Benchmark`, разработать бенчмарки для функций (1) и (2) с размерами векторов $n \in \{8 \cdot 2^0, 8 \cdot 2^1, 8 \cdot 2^2, \dots, 8 \cdot 2^{28}\}$. Бенчмарки должны игнорировать время, затраченное на выделение, копирование и освобождение памяти. Для корректного измерения времени выполнения CUDA-кода необходимо использовать CUDA Events API.
5. Построить графики реальной вычислительной сложности для (1) и (2).
6. Построить график ускорения (2) относительно (1).
7. Объяснить экспериментальные результаты.
8. Подготовить отчёт, содержащий:
 - ключевые фрагменты кода;
 - ссылку на репозиторий с полной реализацией;
 - графики результатов измерений;
 - анализ и интерпретацию полученных результатов.

Критерии оценки

- **Корректность реализации и тестирование (50%):**
 - отсутствие утечек памяти, корректная работа с CUDA API;
 - правильность результатов редукции векторов различных размеров;
 - полнота тестового покрытия, включая граничные случаи;
 - соответствие результатов эталонной реализации.
- **Качество кода и архитектура (25%):**
 - чистота архитектуры, разделение ответственности между классами;
 - единообразие стиля, качество форматирования и читаемость кода.
- **Качество вычислительного эксперимента (15%):**
 - корректность методики измерений производительности;
 - глубина анализа результатов, сравнение с теоретическими оценками.
- **Документация и оформление (10%):**
 - полнота и структурированность отчёта;
 - ясность изложения;
 - оформление репозитория;
 - оформление отчёта (СТУ 7.5-07-2021).

Рекомендации по выполнению

- Реализуйте эффективный алгоритм параллельной редукции, рассмотренный на лекции.
- Ознакомьтесь с официальным руководством CUDA.