

# Практическая работа №4

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

## Цель работы

Получить навыки программирования тензорных ядер CUDA.

## Задание

1. Разработать кёрнел `kernel_matmul_wmma`, который принимает объекты `MatrixView` по значению и вычисляет произведение матриц, используя CUDA WMMA (см. рис. 1).
2. Перегрузить оператор `operator*` для класса `Matrix`, используя указанный кёрнел.
3. Используя фреймворк `Google Test`, разработать модульные тесты для `operator*` со следующими размерами матриц:  $A (m \times k)$  и  $B (k \times n)$ , где  $m, n, k \in \{16, 32, 64, 128, 256, 512\}$ . В качестве эталона для сравнения использовать результат аналогичной операции для `Eigen::Matrix<Eigen::half, Eigen::Dynamic, Eigen::Dynamic, Eigen::RowMajor>`; для верификации результатов применять метод `Eigen::MatrixXf::isApprox` с абсолютной точностью  $10^{-2}$ .
4. Используя фреймворк `Google Benchmark`, разработать бенчмарки для `operator*` со следующими размерами матриц:  $n \times n$ , где  $n \in \{16, 32, 64, 128, 256, 512, 1024\}$ . 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 график реальной вычислительной сложности умножения матриц типа `Matrix` с помощью новой реализации `operator*`, а также аналогичный график для предыдущей версии `operator*`.
6. Построить график ускорения (`speedup`) новой реализации `operator*` относительно предыдущей версии.
7. Объяснить экспериментальные результаты.
8. Подготовить отчёт, содержащий:
  - ключевые фрагменты реализованного 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 умножения матриц 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;

## Рекомендации по выполнению

- Используйте паттерн проектирования стратегия при разработке `operator*`.
- Ознакомьтесь с официальным руководством CUDA.

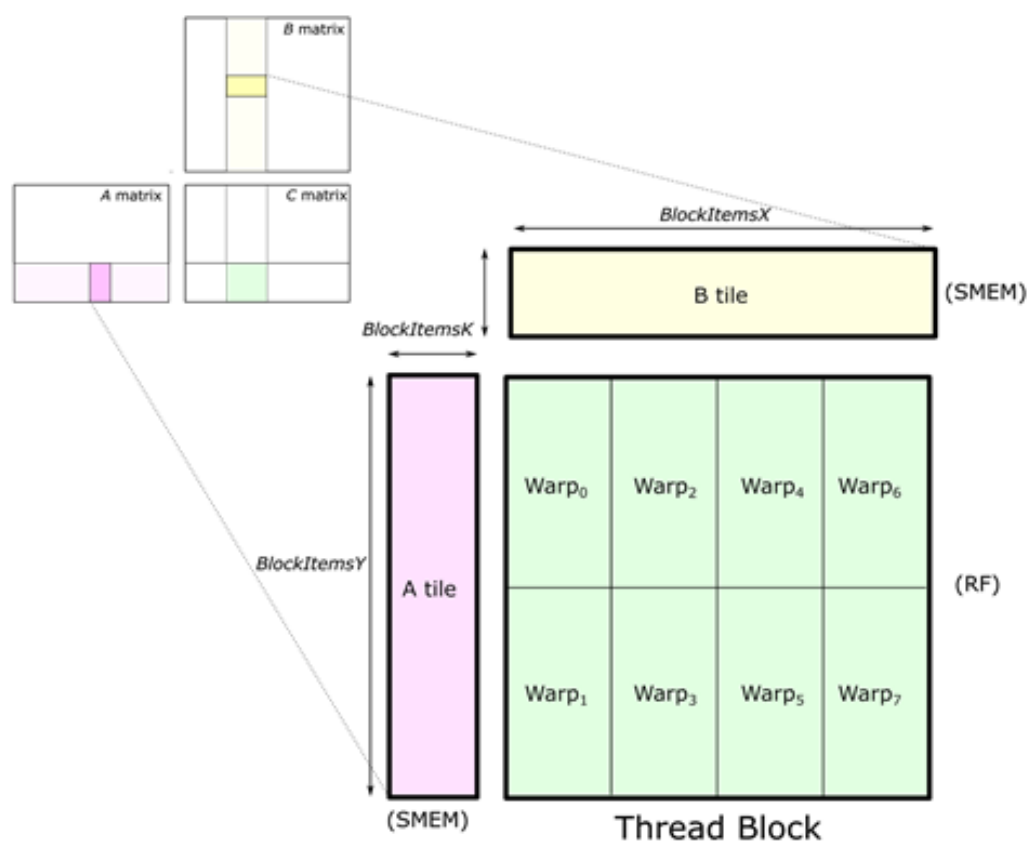


Рис. 1: Схема произведения матриц с помощью WMMA