

ПРАКТИЧЕСКАЯ ЧАСТЬ

АФАНАСЬЕВ ИЛЬЯ
AFANASIEV_ILYA@ICLOUD.COM

ОСНОВНЫЕ ИДЕИ

- К каждой теме прилагается набор из нескольких практических заданий
- В каждом задании есть некоторый готовый шаблон (файл с суффиксом `template`)
- К каждому заданию прилагается возможный вариант решения (файл с суффиксом `solution`)
- В первую очередь пробуем решить задание самостоятельно!
- В конце дня - конкурс, если всё успеем
- Задания не сложные

PINNED VECTOR ADD

- `cd practice/pinned_unified_vec_add`
- `nvcc -O3 sample_vector_add_template.cu`
- `./a.out 100000`
- Используйте шаблон функции `void pinned_sample(int size = 1048576)` чтобы реализовать копирования данных с использованием pinned-памяти
- Измерьте разницу во времени выполнения (какой участок кода нужно измерять?)
- Измерьте пропускную способность копирований при помощи pinned / non-pinned памяти
- При помощи Вашего любимого редактора постройте график

UNIFIED VECTOR ADD

- Сделайте аналогичное сложение при помощи `cudaMallocManaged` (Unified memory)
- Инициализируйте вектора при помощи сгенерированных заранее данных
- Сделайте аналогичные выводы (при необходимости используйте профилировщик NVVP)

PINNED MATRIX MULTIPLICATION

- Вчера вы реализовывали матричное умножение...
- Добавьте Pinned память и туда!
- Сравните полученный результат с Vector Add, сделайте выводы.

CUDA STREAMS

1. Concurrent data-copies
2. Concurrent kernels

VECTOR ADD & STREAMS

- Возьмем снова наш простой пример сложения векторов.
- Оценим перспективу использования потоков в данном примере и то, какое ускорение мы можем получить.
- (используем профилировщик NVVP)
- Выводы?

CUDA STREAMS + RANDOM MEMORY ACCESS

- Рассмотрим другую задачу - random memory access
- `cd streams_random_access`
- Что и зачем делает ядро `random_access_kernel`?
- Какими особенностями оно обладает?
- Оптимизируйте копирования данных в данной задаче при помощи потоков CUDA.
- Произведите графическую профилировку полученной программы при помощи локальной версии NVVP.

MULTI GPU + VECTOR ADD

- Откройте наш первый пример с векторным сложением
- Измените его так, чтобы сложение производилось сразу на двух GPU
- Получилось ли ускорение в два раза? Почему?