Начало

Инициализация:
n = 1000000
size = n * sizeof float

Выделение памяти
на CPU: h_arr

Заполнение массива
h_arr i = i + 1.0

Выделение памяти
на GPU: d_arr

Установка конфигурации:
blockSize = 256
numBlocks = n +
blockSize - 1 /
blockSize

Копирование данных
CPU → GPU

Создание событий
start, stop

Запись события start

Запуск ядра
multiplyGlobal

Ядро multiplyGlobal:
для каждого потока

idx = blockIdx.x *
blockDim.x +
threadIdx.x

idx < n?

arr idx *= multiplier

Конец ядра

Запись события stop

Вычисление времени
выполнения

Вывод времени
глобальной памяти

Копирование данных
CPU → GPU повторно

Запись события start

Запуск ядра
multiplyShared

Ядро multiplyShared:
для каждого потока

idx = blockIdx.x *
blockDim.x +
threadIdx.x

idx < n?

Копирование в
разделяемую память:
shared_data
threadIdx.x = arr idx

Синхронизация потоков
__syncthreads

shared_data threadIdx.x
*= multiplier

Синхронизация потоков
__syncthreads

Копирование обратно:
arr idx = shared_data
threadIdx.x

Конец ядра

Запись события stop

Вычисление времени
выполнения

Вывод времени
разделяемой памяти

Копирование результата
GPU → CPU

Освобождение памяти GPU

Освобождение памяти CPU

Конец