Events and Streams

Dmitri Nesteruk
@dnesteruk

pluralsight
hardcore developer training

Overview

- Events
- Event API
- Event example
- Pinned memory
- Streams
- Stream API
- Example (single stream)
- Example (multiple streams)

Events

- **How to measure performance?**
- **Use OS timers**
  - Too much noise
- **Use profiler**
  - Times only kernel duration + other invocations
- **CUDA Events**
  - Event = timestamp
  - Timestamp recorded on the GPU
  - Invoked from the CPU side

Event API

- **`cudaEvent_t`**
  - The event handle
- **`cudaEventCreate(&e)`**
  - Creates the event
- **`cudaEventRecord(e, 0)`**
  - Records the event, i.e. timestamp
  - Second param is the *stream* to which to record
- **`cudaEventSynchronize(e)`**
  - CPU and GPU are async, can be doing things in parallel
  - `cudaEventSynchronize()` blocks all instruction processing until the GPU has reached the event
- **`cudaEventElapsedTime(&f, start, stop)`**
  - Computes elapsed time (msec) between `start` and `stop`, stored as `float`

Pinned Memory

- **CPU memory is *pageable***
  - Can be swapped to disk
- **Pinned (page-locked) stays in place**
- **Performance advantage when copying to/from GPU**
- **Use `cudaHostAlloc()` instead of `malloc()` or `new`**
- **Use `cudaFreeHost()` to deallocate**
- **Cannot be swapped out**
  - Must have enough
  - Proactively deallocate

Streams

- **Remember `cudaEventRecord(event, stream)`?**
- **A CUDA *stream* is a queue of GPU operations**
  - Kernel launch
  - Memory copy
- **Streams allow a form of task-based parallelism**
  - Performance improvement
- **To leverage streams you need device overlap support**
  - GPU_OVERLAP

Stream API

- **`cudaStream_t`**
- **`cudaStreamCreate(&stream)`**
- **`kernel<<<blocks,threads,shared,`*`stream`*`>>>`**
- **`cudaMemcpyAsync()`**
  - Must use pinned memory!
- **`stream parameter`**
- **`cudaStreamSynchronize(stream)`**

Summary

- **CUDA events let you time your code on the GPU**
- **Pinned memory speeds up data transfers to/from device**
- **CUDA streams allow you to queue up operations asynchronously**
  - Lets you do *different* things in parallel on the GPU
  - Use of pinned memory is required