CUDA프로그래밍 따라하기

유영천

https://megayuchi.com

tw:@dgtman

목표

- CUDA 프로그래밍의 한 문턱을 넘어간다.
  - 보통 소개와 다이어그램만 보고 끄덕거리다 끝난다.
  - 실제로 프로그램을 짜보자.

CUDA 기초

CUDA device 생성

데이터 전송

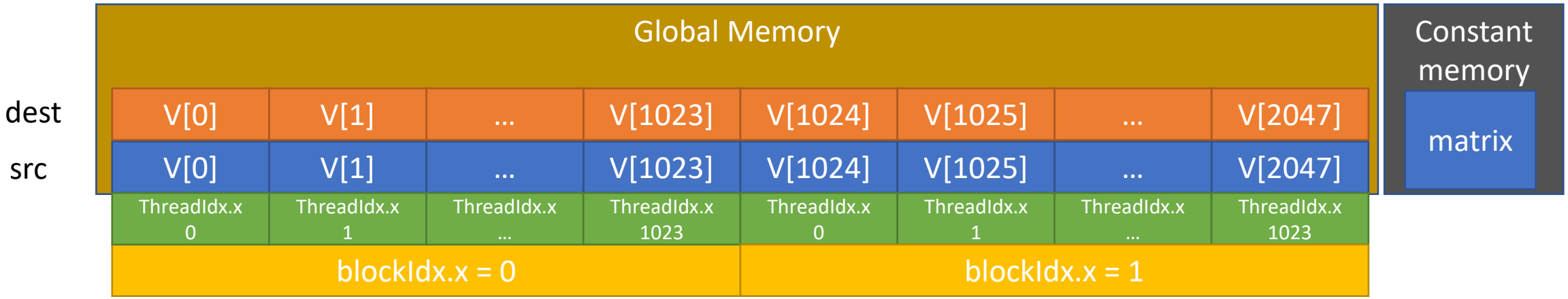
- cudaMemcpy()
- cudaMemcpyToSymbolAsync()

CUDA kernel

- 실제 호출되는 GPU함수. CUDA에서의 main()함수.
- __global__로 지정

예제 설명

- Float4배열 x matrix4x4
- 적합한 예제는 아니지만….


Global Memory
Constant memory
matrix
dest
src
V[0]
V[1]
...
V[1023]
V[1024]
V[1025]
...
V[2047]
V[0]
V[1]
...
V[1023]
V[1024]
V[1025]
...
V[2047]
ThreadIdx.x 0
ThreadIdx.x 1
ThreadIdx.x ...
ThreadIdx.x 1023
ThreadIdx.x 0
ThreadIdx.x 1
ThreadIdx.x ...
ThreadIdx.x 1023
blockIdx.x = 0
blockIdx.x = 1

__syncthreads()

- Block내의 모든 스레드들이 __synchthreads()에 도달할때까지 블럭킹 됨.
- Shared memory를 사용하려면 거의 항상 따라옴

Shared memory

- 작업용 고속 메모리.
- __shared__로 지정
- SM당 64KB의 캐시 메모리를 48KB shared memory + 16KB L1 캐시로 사용.
- 잘 쓰면 성능 up. 적합하지 않은 상황에서 사용하면 성능 down.
- 이걸 꼭 사용하려고 너무 강박관념을 가질 필요는 없다.

CPU multi thread – CUDA stream

CUDA 응용

- 성공사례
  - CUDA Lightmap Baking
  - CUDA Raytracing을 이용한 오브젝트 가시성 테스트
  - Image Filter
- 실패사례
  - CUDA 길찾기
  - 서버에서의 충돌처리 – 득보다 실이 크다.
  - 기타 등등

CUDA적용의 결정 기준(지극히 개인적)

- CPU 멀티 스레드 코드 대비 2x 이상 빠르지 않을 경우
  - 망했다. CPU코드 최적화에 힘쓴다.
- CPU 멀티 스레드 코드 대비 2x 이상 4x미만으로 빠를 경우
  - 만족스럽진 않지만 사용하는 편이 이득이 있는지 생각해본다.
- CPU 멀티 스레드 코드 대비 4x 이상 빠를 경우
  - CUDA코드 사용을 적극 권장한다.

디버깅