CUDA(Compute Unified Device Architecture)

nvcc : NVIDIA CUDA Compiler

msvc : Microsoft Visual C++ Compiler

\* General CUDA Program Scenario

1. Serial code(host)

- serial execution : read data

- prepare parallel execution

- copy data from RAM to VRAM

1. Kernel code(device)

- parallel processing

-read/write data from VRAM to VRAM

-CUDA do not have any direct i/o feature

1. Serial code(host)

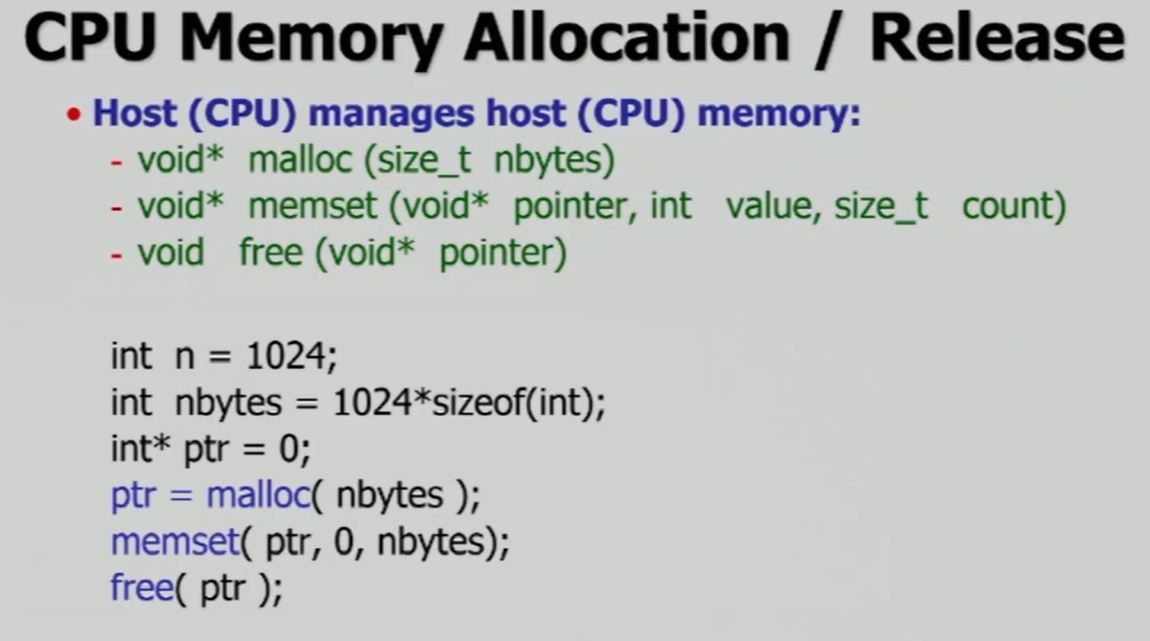
- copy data from VRAM to RAM

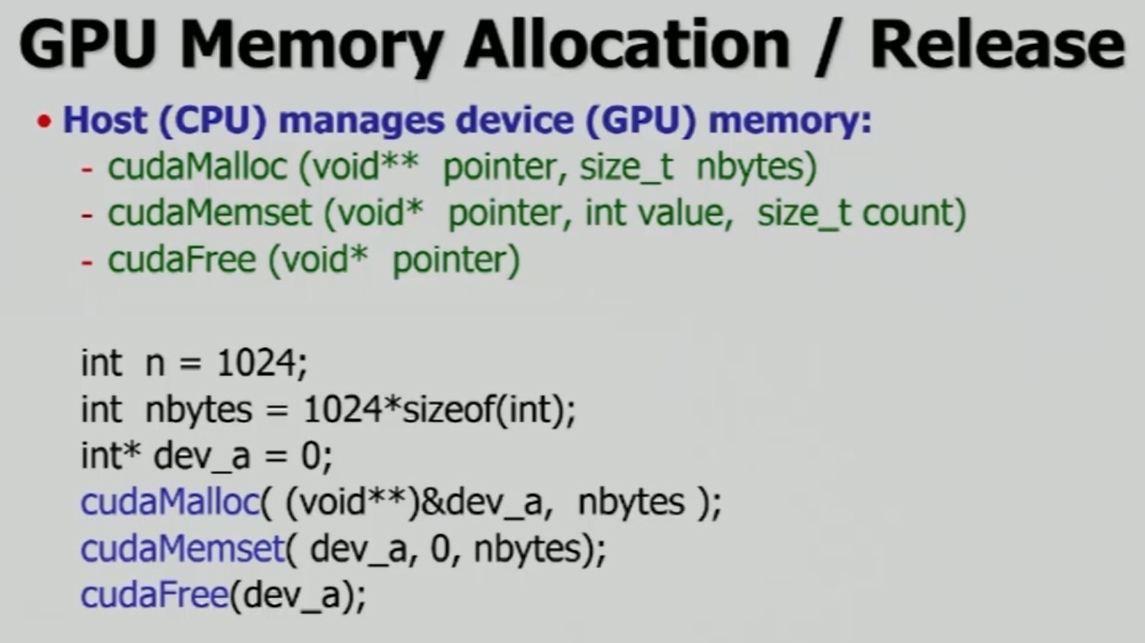
- serial execution : print data

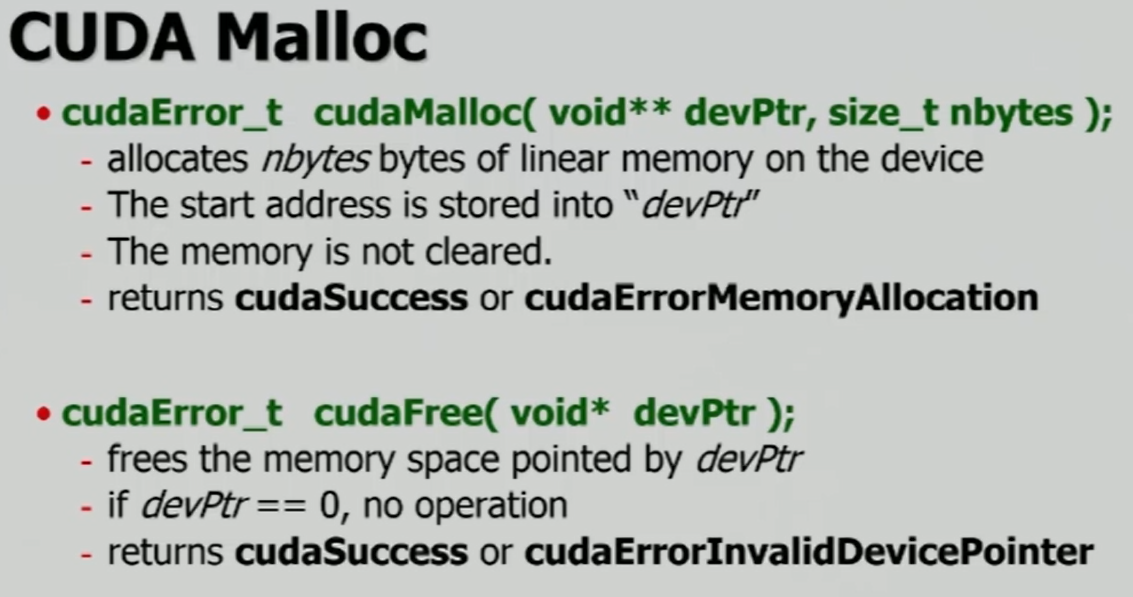
\* GPU cores share the “global memory”

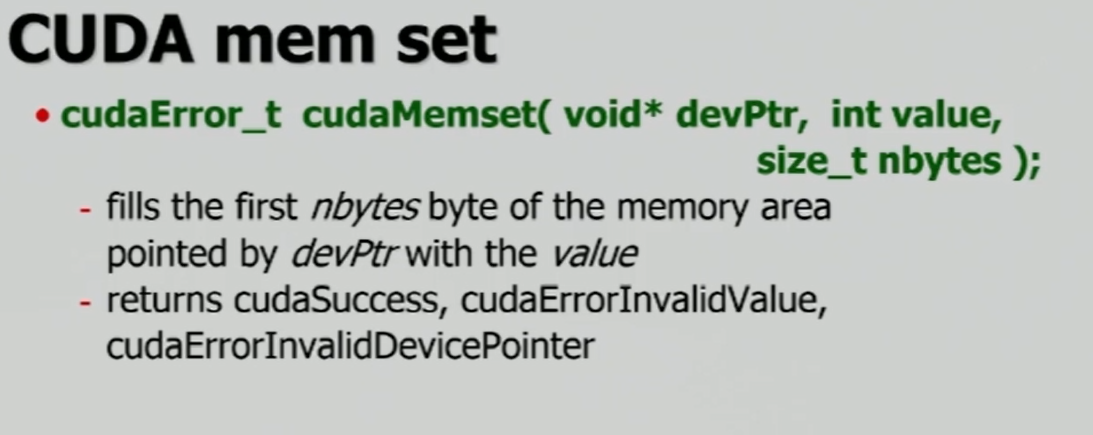
- later, we will introduce the CUDA memory hierarchy.

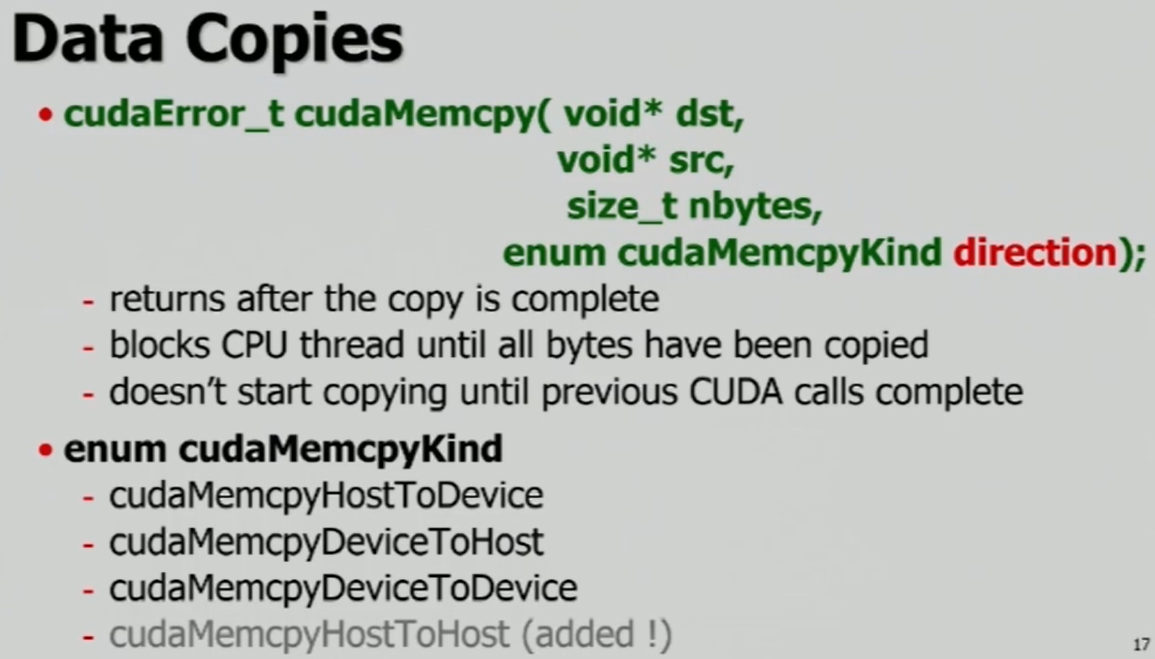
-at this time, we will use only the CUDA global memory

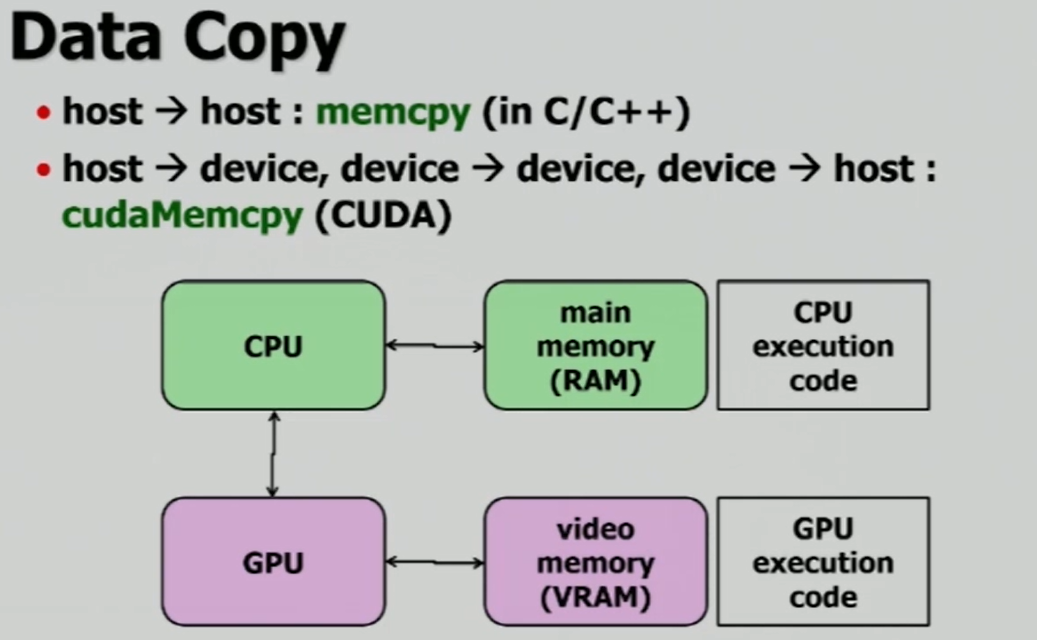


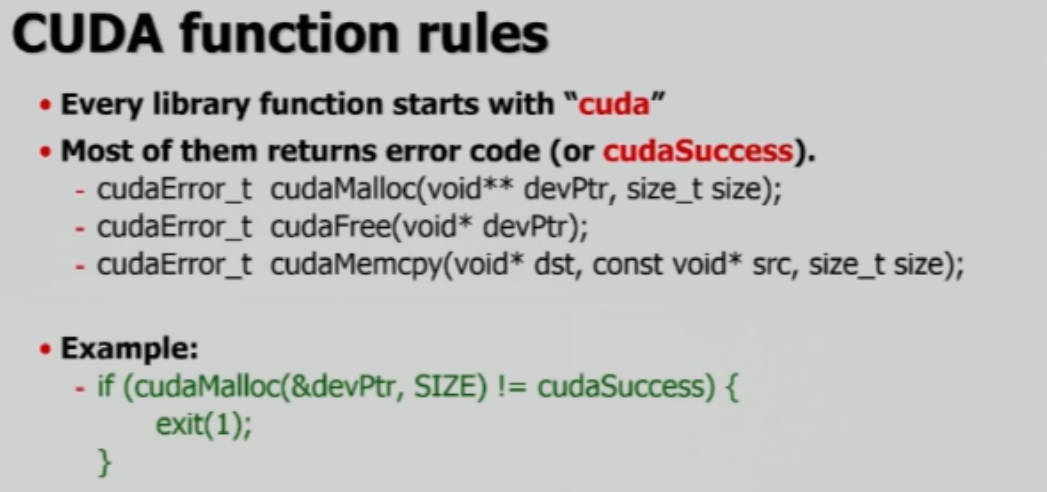




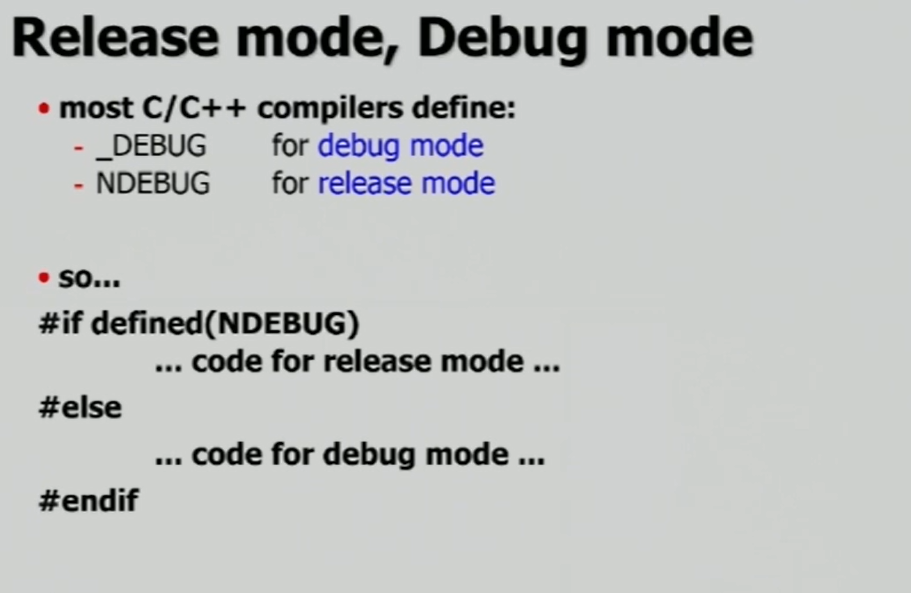








모든 cuda 함수는 cudaError\_t를 리턴한다.



Process and Thread

컴퓨터 프로세스 : 컴퓨터 프로그램(하드나 디바이스에 저장되어 있는) 메인 메모리로 옮겨와서 실행 시켰을 때 이를 프로세스라 한다.

- 내부적으로 프로세스는 프로그램 코드와 스테이터스 데이터로 이루어 져 있다.

- 하나의 프로세스 안에 여러개의 프로그램 코드를 시행 시키기 위해 스레드를 사용한다.

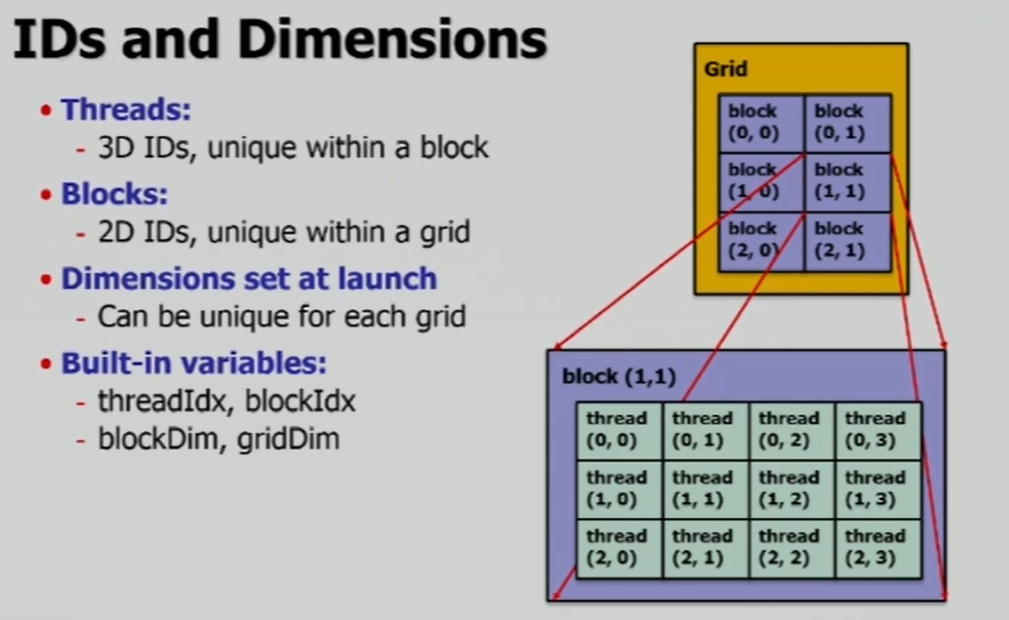
스레드 : 운영체제나 실행 할수 있는 플래폼상의 스케줄 상의 다룰 수 있는 최소한의 단위 이다.

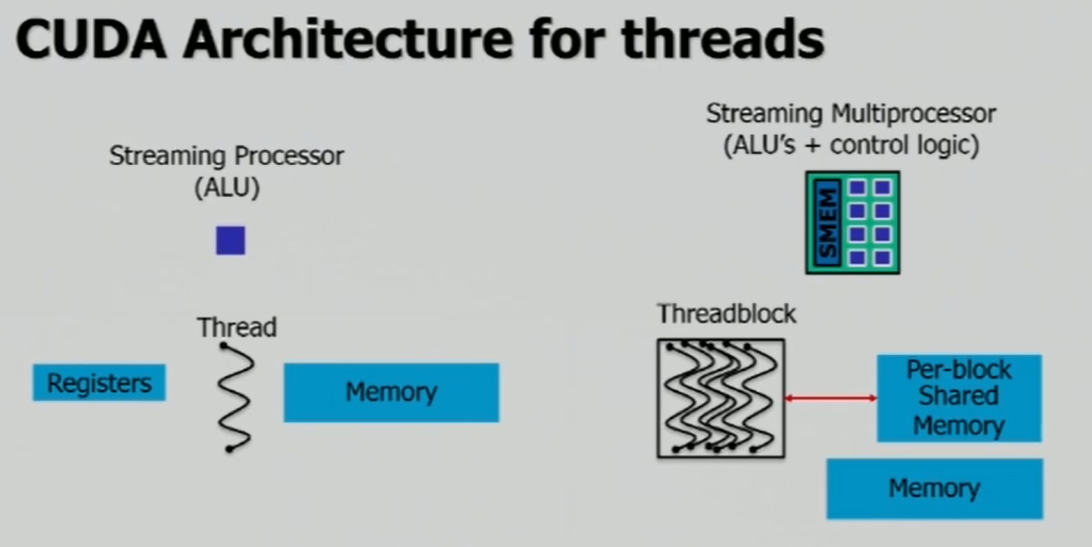
- 하나의 스레드는 하나의 코어에서 실행.

Launches are hierarchical : Threads -> blocks -> grids

- Threads are grouped into blocks

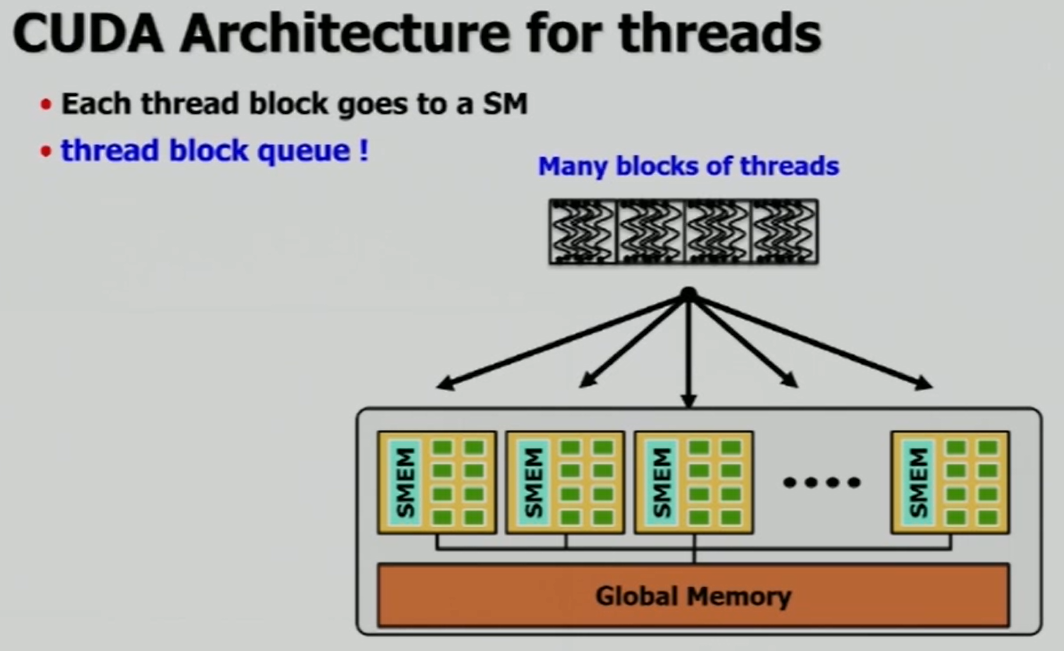
- Blocks are grouped into grids





Sp가 하나의 코어라 할수 있고 하나의 스레드를 수행할 수 있다.

Sm는 여러개의 스레드를 동시에 수행 할수 있다.



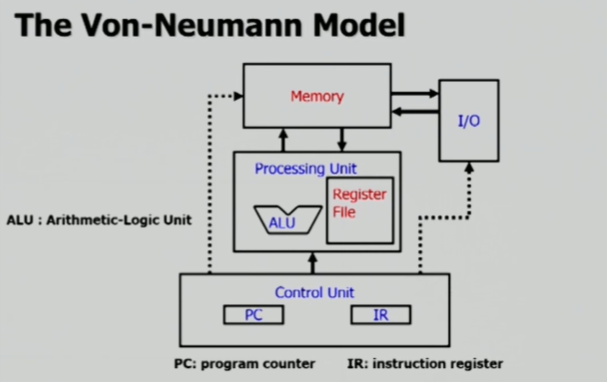
Block size 1 to 1024 concurrent threads

Divided into warps

1warps = 32 threads

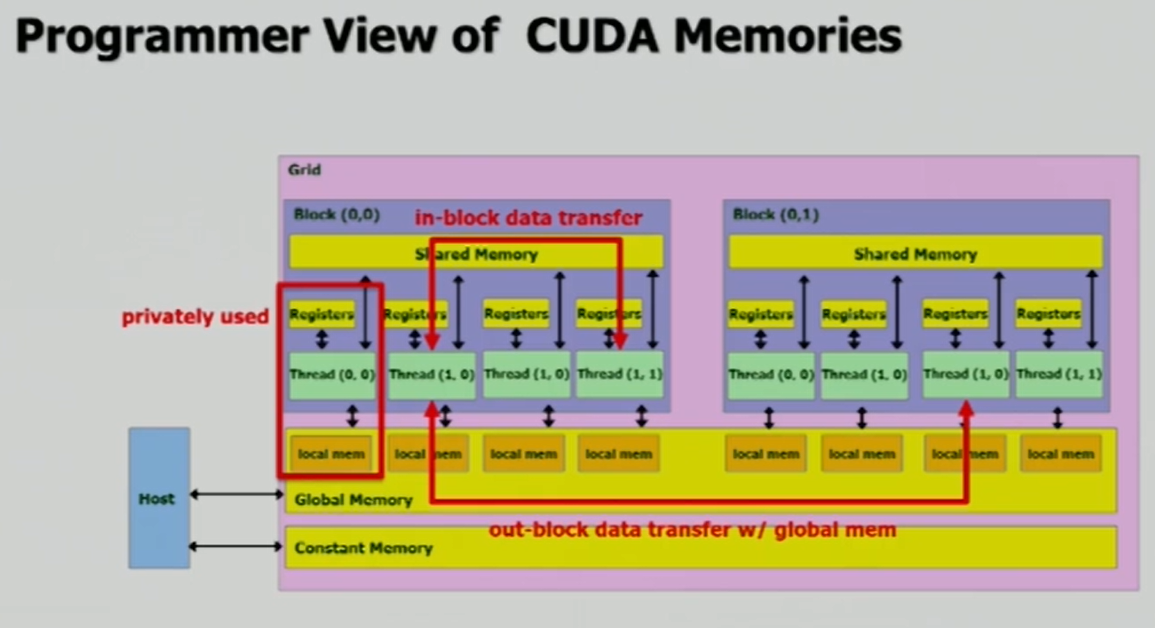
1SM = 32cores -> parallel execution

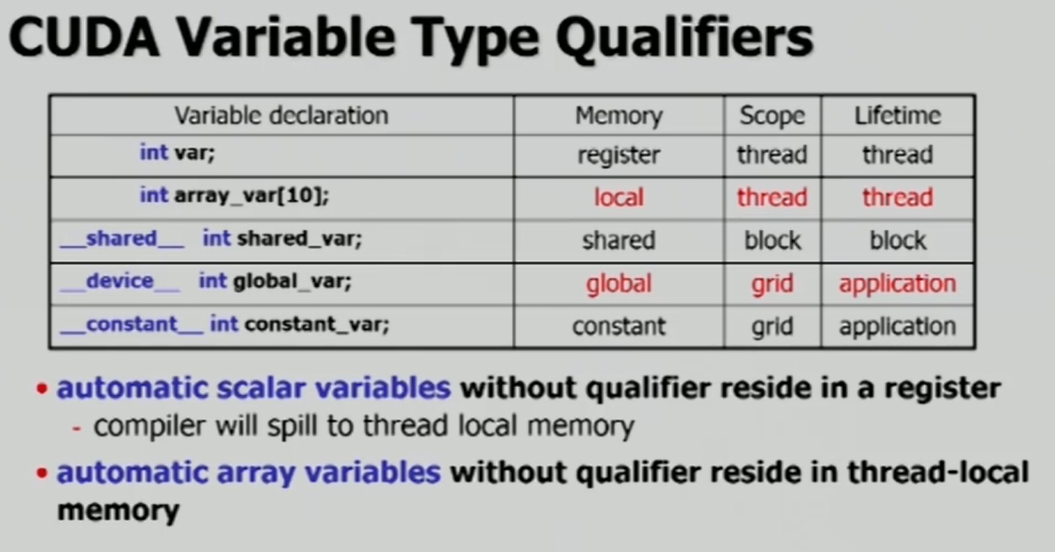
폰노이만 모델

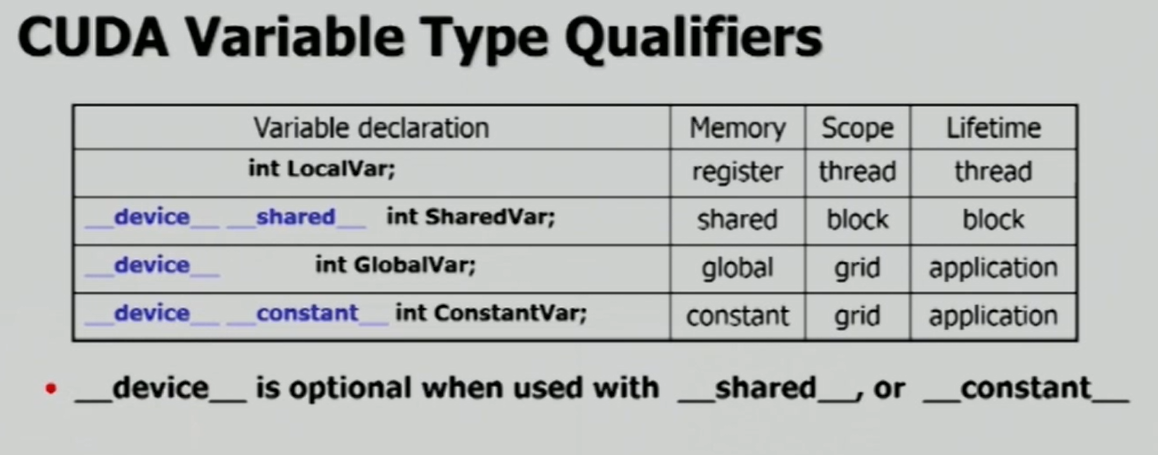


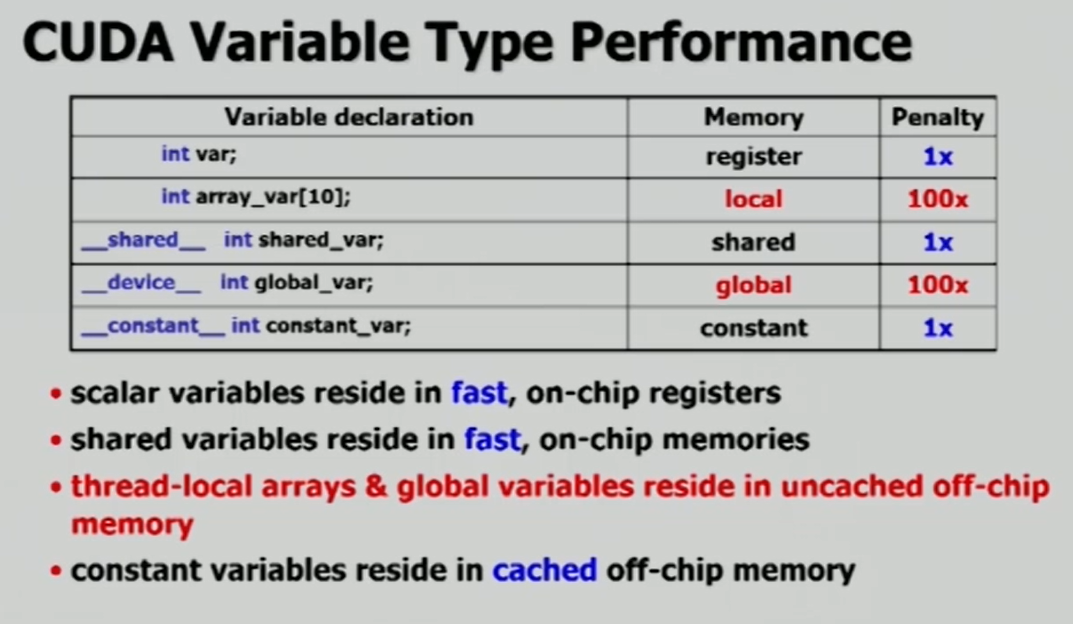
메모리 ( 코드, 데이터 ... )

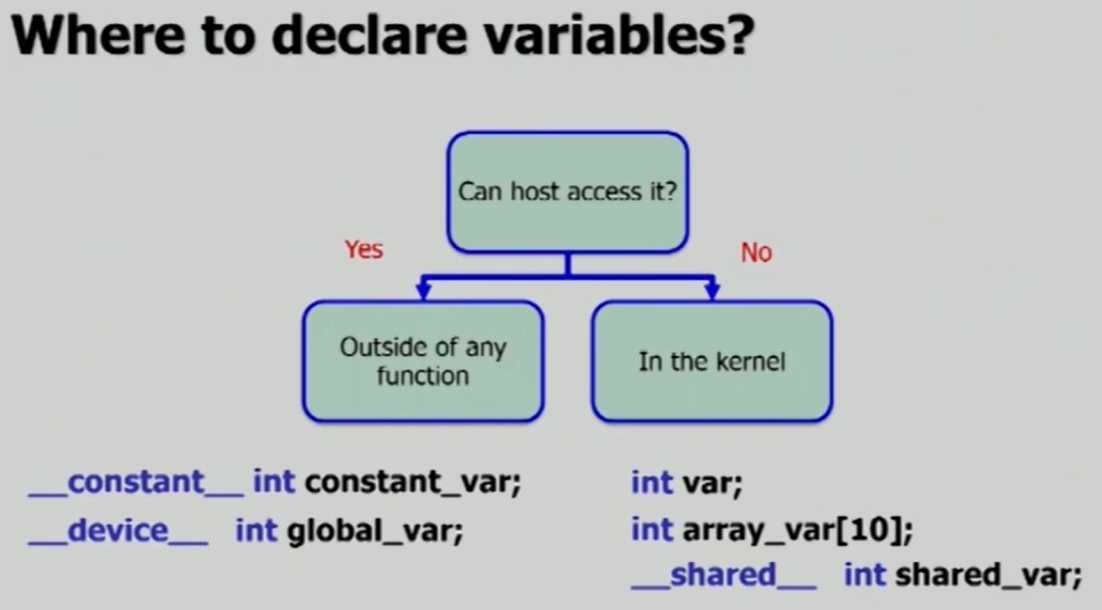
문제점 : 버스에서 병목현상

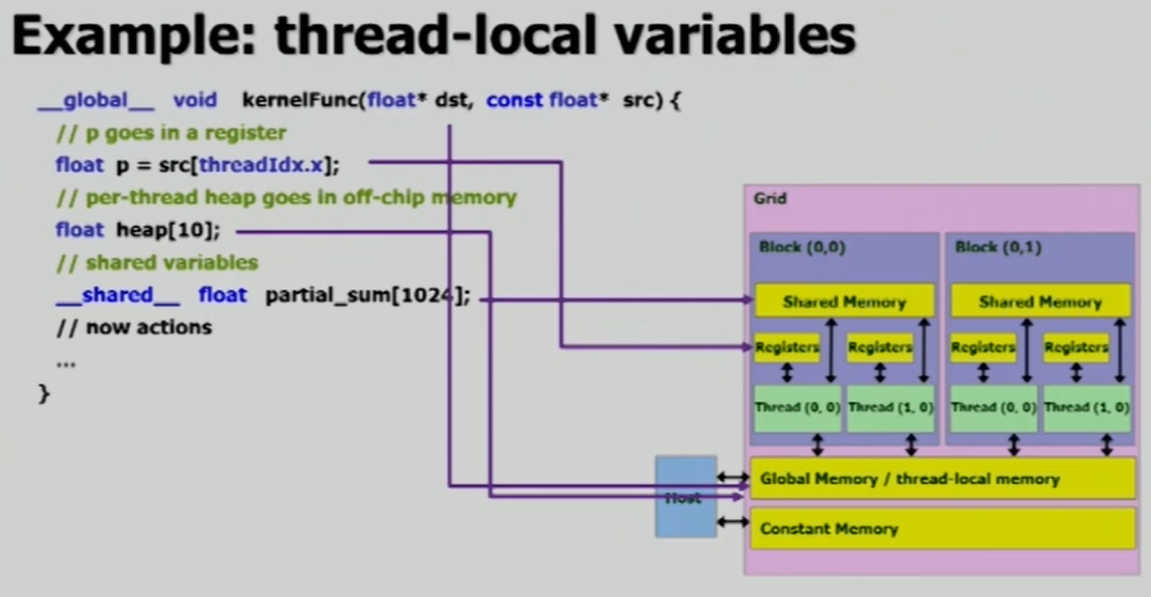


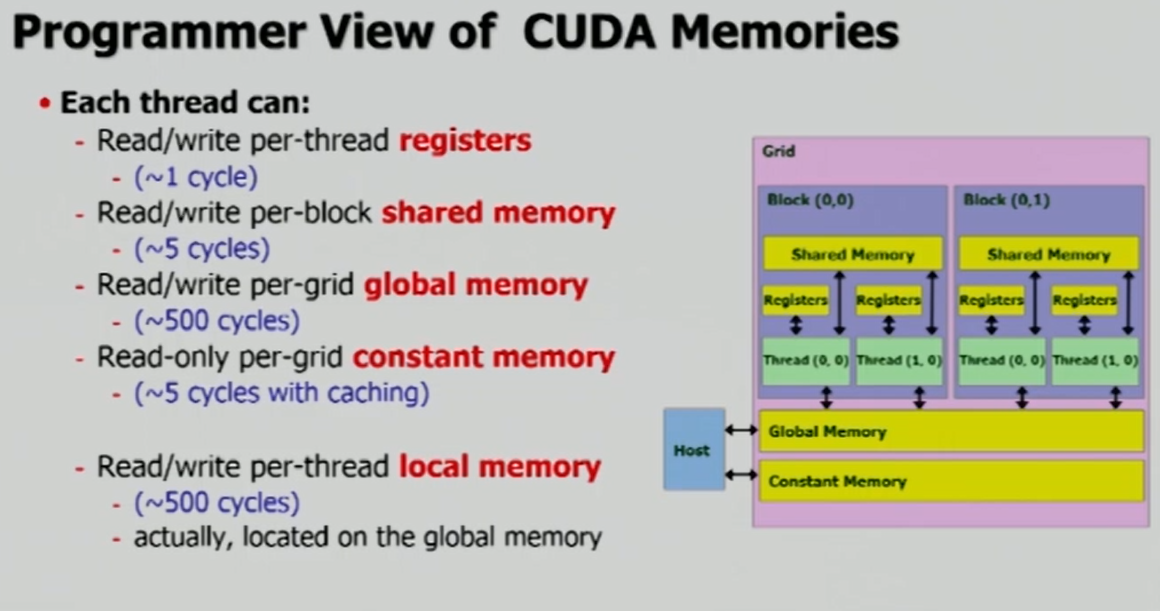












<https://wonjayk.tistory.com/105>

