

병렬 컴퓨팅

초창기 컴퓨팅 장치 성능향상의 주요방법은 "클럭" 속도를 높이는 것.

80년대 1MHz → 2010년대 1GHz ~ 3GHz (약 1000배 상승)

클럭 ↑ : 근본적 H/W 문제로 클럭 속도를 높이는 것에 한계가 발생함. 1. 전압과 열 문제 2. 도커지스터 크로제

⇒ 1개의 코어가 아닌 여러코어를 탑재해서 성능향상 (멀티코어)

90년대 초. U와들은 개인컴퓨터를 위한 2D 디스플레이 가속기 구입.

ps OpenGL 등장 (실시간 그래픽스)

그래픽 성능을 측정하는 방법이나 사용성을 보.

90년대 중반 - 3D 그래픽스 이용자 수 (게임인기) + H/W 회사들의 그래픽 가속기 개발.

렌더링 : 2차원 아 3차원 장면을
사진이나 영상으로 만드는 작업.

초창기 GPU 컴퓨팅 : 프로그래밍이 가능한 파이프라인 탑재 GPU의 발매.

OpenGL + DirectX 를 이용하여 단순 렌더링 목적 이상으로 사용기를 위한.

초창기 방법 : 그래픽스 API를 통한 범용 계산 방법강구.

ex) 픽셀 생성시 원하는 data를 넣어서 결과로 받는다.

CUDA 아키텍처가 등장했지만, 여전히 OpenGL, DirectX를 이용.
그래픽스 문제로 위장해야 하는 문제 존재.

NVIDIA - CUDA Language, cudacompiler 개발

→ 범용 계산 가능해짐.

Host - CPU와 시스템메모리 Device - GPU와 리바이스 메모리

디바이스에서 수행되는 함수를 일반적으로 커널(kernel)이라고 한다.

```
*include <stdio.h>
```

```
-- global -- void kernel (void) {  
{
```

```
int main (void) {
```

```
kernel <<< 1, 1 >>>();
```

```
printf("Hello, world");
```

```
return 0;
```

```
}
```

1. 디바이스에서 작동하는 code임을 명시.

2. 호스트가 디바이스 코드를.

runtime system에 넘겨줄 인자, 디바이스 코드 전달 X
런타임이 디바이스 코드를 어떻게 해석할지 영향을 준다.

즉, 1. 커널에 매개변수 전달 가능. 2. 반대로 Host 전달 device에서 수행할 작업을 위한
디바이스 메모리 할당이 필요.

정리. `cudaMalloc()`으로 할당된 메모리 포인터를 디바이스에서 실행되는 함수로 전달 가능.

device에서 실행되는 코드에서 `cudaMalloc()`으로 할당된 메모리 포인터를 이용하여 일꾼들.

`cudaMalloc()`으로 할당된 메모리 포인터를 Host에서 실행되는 함수로 전달할 수 있다.

호스트에서 실행되는 코드에서 `cudaMalloc()`으로 할당된 메모리 포인터를 이용하여
메모리를 읽거나 쓸 수 없다.

```
int count;
```

```
cudaGetDeviceCount(&count);
```

→ cuda device 개수 알기.

`cudaDeviceProp prop` → cuda 디바이스 정보질의 가능.

* 다수의 gpu중에서 원하는 gpu를 선택하는법.

`cudaDeviceProp prop` 선언.

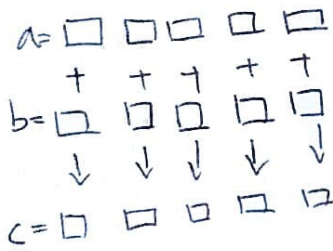
`prop. ~~` = 원하는 value

이후 `cudaChoseDevice` 함수 호출. 인자로 넘겨줌.

나온 device로 번호를 가지고 세팅한다.

CUDA 병렬 처리 이해를 위한 백서함.

a, b, c는 각각 array.



병렬블록개수: 블록숫자만큼 디바이스가 개를 세기.

적절한 매개변수의미 kernel <<<N, 1>>>

앞 인자만을 커널의 인자를 생성하고 병렬로 개시한다. 이때 병렬실행 각각을 블록(block)이라고한다.

몇번째 block인지는 block.Idx로 확인이 가능하다. (해당 값은 runtime이 할당).

왜 block.Idx가 아닌 block.Idx.x 인가? <2차원블록을 하나의 값으로 정의하게 해주기 때문이다.>
간단히 2차원 인덱싱이 1차원 인덱싱보다. 종종 편리하다.

병렬블록 인덱스 = blockIdx (크리드)

주의사항* 블록의 차원이 65,535를 초과X.

dim3 타입은 커널의 실행 크기를 명시하는데 사용되는 3차원 요소들 집합.