**Assignment 2**

(GPU 컴퓨팅)

담당교수: 공영호 교수님

학과: 컴퓨터정보공학부

학년: 3

학번: 2018202074

이름: 김상우

#코드 작성

과제를 들어가기 앞서, 과제 조건에 대하여 살펴보았습니다.

* Vertor addition code의 size는 5이다.
* 해당 코드는 d[i] = a[i] + b[i] +c[i]라는 동작을 한다.
* a,b,c array는 랜덤한 value로 초기화 되어야 한다.

즉, 랜덤하게 초기화된 길이가 5인 배열 a,b,c를 gpu환경에서 더해 CPU에서 출력해주는 것을 목표로 하면 될 것입니다. 위 정보를 바탕으로 다음과 같이 코드를 진행하였습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Google colab에서 해당 코드를 Assignment2cuda.cu라는 코드로 인식하기 위해 위에 %%writefile Assignment2cuda.cu로 기술하였고 아래 필요한 헤더들을 기술하였다.

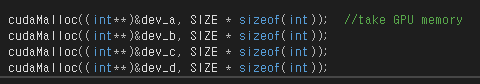
또한 d[i]=a[i]+b[i]+c[i]를 수행하는 함수를 CPU에서 GPU로 명령하기 위해 \_\_global\_\_을 추가하고, threadidx.x와 d[i]=a[i]+b[i]+c[i];를 이용해 thread의 숫자를 받아 각 숫자에 대해 vector addition이 동시에 진행될 수 있게 하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Array의 size가 5이므로 이후 사용을 위해 따로 SIZE를 5로 선언해주고 해당 길이로 a,b,c를 선언하고 0으로 초기화해주었다. 이후 해당 배열들을 random한 값을 넣어주기 위해 rand함수를 이용하였다. 이때, 합한 값에 대해 overflow를 막기 위해 난수 생성 후 4의 값으로 나누어 주었다.

이후 덧셈 결과를 담을 SIZE(5) 길이의 배열, d를 선언하고 위 배열들의 값을 GPU에서 받거나 저장할 dev\_a, dev\_b, dev\_c, dev\_d 또한 선언해주었다.



cudaMalloc을 이용, dev\_a, dev\_b, dev\_c, dev\_d에 int가 SIZE(5)만큼 들어갈 만큼 GPU메모리를 할당시켰다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이후 cudaMemcpy를 이용, GPU의 메모리를 할당받은 dev\_a, dev\_b, dev\_c에 CPU내의 a,b,c값들을 복사해준다. 이때 HOST에서 Device로 값을 int가 5개 들어가는 만큼 복사하므로 3번째 인사는 sizeof(int) \* SIZE, 4번쨰 인자는 cudaMemcpyHostToDevice가 들어가게 된다.

이후 addKernel을 이용 위의 addKernel에 대해 i가 0부터 4까지의 경우에 대해 진행한다. 결과적으로 dev\_a,dev\_b,dev\_c 배열의 i번째 요소를 dev\_d의 i번쨰에 저장하게 될 것이다.

이후 그렇게 값을 받은 dev\_d의 값을 cudaMemcpy를 이용 cpu의 d배열에 복사한다. Device에서 Host로 값을 int가 5개 들어가는 만큼 복사하므로 3번째 인사는 sizeof(int) \* SIZE, 4번쨰 인자는 cudaMemcpyDeviceToHost가 들어가게 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이후 a,b,c,d의 배열의 각 요소를 출력한다. 예상이 맞다면 난수의 값들인 a[i],b[i],c[i]의 요소들이 더해진 값이 d[i]에서 출력될 것이다.

이후 dev\_a, dev\_b, dev\_c, dev\_d에 할당된 GPU 메모리를 cudaFree로 해제하고 return 0로 코드를 종료한다.

#Google Colab

텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명

위에서 작성된 코드를 적용하였다. 맨 위에 적힌 코드에 의해 Assignment2cuda.cu가 생기게 된다.

테이블이(가) 표시된 사진

자동 생성된 설명

이후 nvcc -o를 이용 cuda capability를 sm\_35 버전으로 컴파일 할 수 있도록 실행파일을 생성하며 이때 이름을 Assignment2cuda로 지정해주었다.

이렇게 생긴 실행파일을 !./Assignment2cuda를 통해 실행하자, 난수 3개와 이들을 더한 값이 맞게 출력되는 것을 확인할 수 있다.

#고찰

GPU 컴퓨팅 Assignment2을 진행하면서 google colab에서 vector addition을 진행해볼 수 있었습니다. 이러한 과정을 통해 GPU가 CPU에 대해 어떠한 이점을 갖는지 알 수 있었습니다.(병행 처리에 있어 thread id를 사용하는 만큼 반복문을 쓰는 것보다 빠르게 처리된다는 장점이 있다.) 그 때문에 딥러닝 같은 병행처리가 많이 진행되는 것에 대해 GPU가 활용되는 이유도 알 수 있었습니다. 또한 \_\_global\_\_을 사용해보면서 Host 에서 device로 함수를 전달하는 방법에 대해 알 수 있었습니다.