고급소프트웨어 실습 11주차 과제

20171693 조병화

**실습 1**

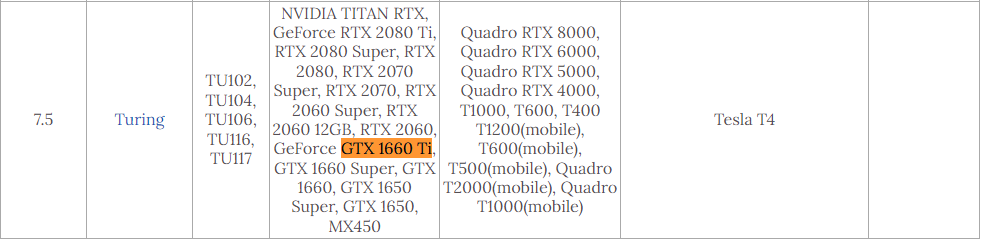
1. **사용하는 컴퓨터에 장착된 GPU의 기종**

실습을 진행한 컴퓨터에 장착된 GPU의 기종은 NVIDIA GeForce GTX 1660 Ti

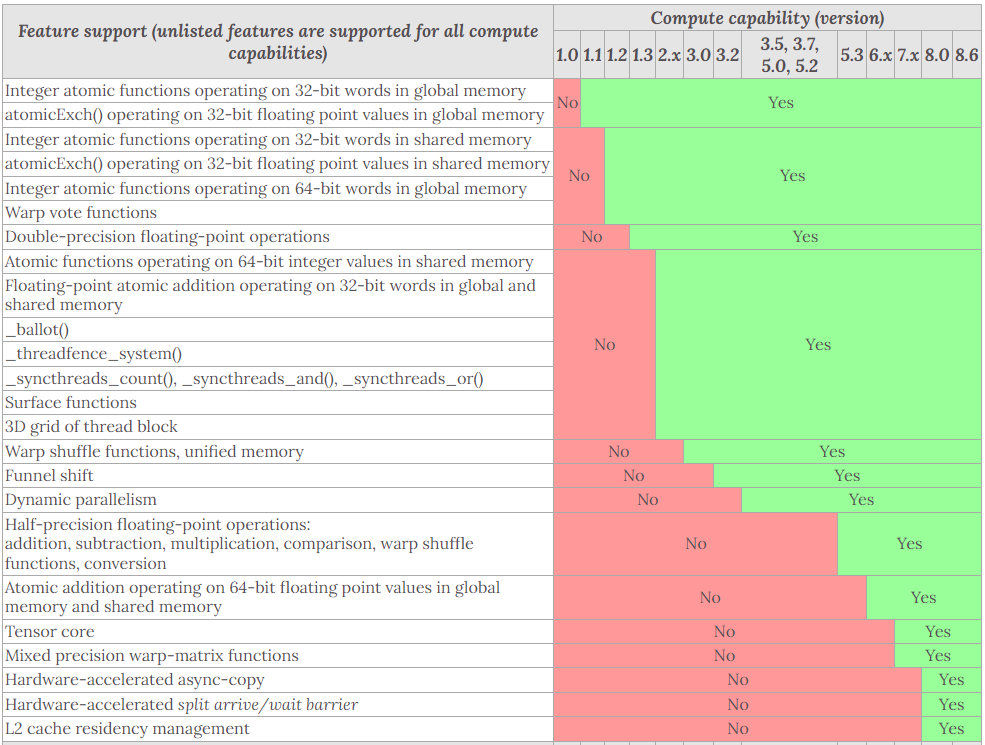
1. **현재 설치된 CUDA 시스템의 Compute Capability**

그리고 이 GPU의 compute capability는 7.5 였다





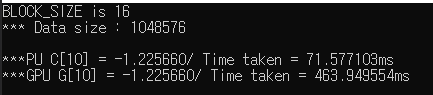
1. **현재 CUDA Compute Capability가 제공하는 각종 성능 및 스펙**

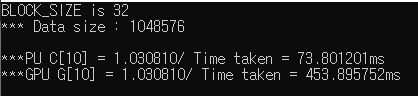


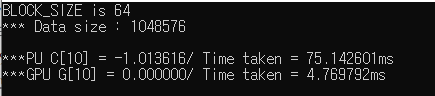
다음과 같은 기능들을 제공하는점을 알 수 있었다. 출처 : https://www.wikiwand.com/en/CUDA

**실습 2**

**다음 이 CUDA 프로그램에 대하여 블럭 크기를 다양하게 변화 시켜가면 서 시간을 측정한 후, 그 결과를 보고서에 테이블로 요약하라.**







블록의 크기를 바꿔가며 연산해본 결과, BLOCK SIZE가 64일 때 두 시간이 다른 것을 통해 error가 발생하였음을 확인할 수 있었다. 다음의 결과는 각 gpu의 계산 과정에 의미가 없는 loop들으 돌려 쓰레드의 계산시간을 의도적으로 길게 바꾼것이다.

|  |  |  |
| --- | --- | --- |
| BLOCKSIZE | CPU Time | GPU Time |
| 16 | 71.577103ms | 463.949554ms |
| 32 | 73.801201ms | 453.895752ms |
| 64 | 75.142601ms | 4.769792ms |

**실습3**

**(i) 먼저 for 문장을 사용하여 순차적으로 n번의 행렬-벡터 곱셈을 수행해주는 C/C++ 함수를 작성한 후 CPU 수행 시간을 측정하라. 시간 측정 방법은 이 전 실습 시간에 사용한 방법을 사용하고, 가급적 정확한 시간 측정을 위하 여 여러 번 반복적으로 함수를 수행시킨 후 평균값을 취할 것. (ii) 다음 이 문제를 해결해주는 CUDA 커널 프로그램을 작성한 후 가급적 정 확하게 GPU 수행 시간을 측정하라. 앞의 문제에서와 같이 쓰레드 블럭의 크기를 변화시켜가면서 수행 시간 관점에서 CPU 방법과 비교 분석한 후, 그 결과를 보고서 기술하라.**

위 문제를 해결하기 위해 cpu, gpu 각각에 대하여 5번씩의 테스트를 돌려 평균을 계산해보았다.







다음은 5 케이스의 평균으로는 다음과 같이 나왔다. (Block Size는 16으로 했을때의 결과이다.)

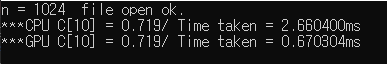
Cpu : 2.7436

Gpu : 0.6609

다음은 Block Size를 변경하면서 수행시간을 측정해 본 것이다.

블록사이즈가 32일 때

블록사이즈가 64일 때

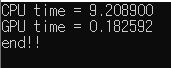
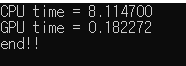
블록사이즈가 128일 때

결과로, 블록만큼 작업을 분할 처리하는 GPU의 속도가 미세히 더 빨라지는 것으로 확인하였다.

또한, Gpu와 Cpu를 비교해보니, Gpu가 더 빨라짐을 확인할 수 있었다.

**과제 1**

우선 실험결과를 여러 번 반복하여 계산된 속도는 다음과 같다.



이를 통해 구해진 수행시간의 평균은 다음과 같다. (Block size = 16일 때)

CPU 수행시간 : 8.5119

GPU 수행시간 : 0.122048

다음은 block size가 8, 32, 64 일 때에 각각 수행해본 것이다.

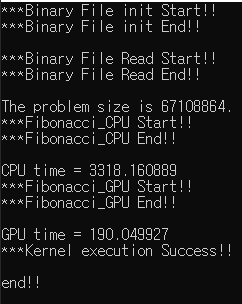
32일 때, 64일 때.

 8 일 때

이를 통해 Block size를 더 줄였을 때는 시간이 매우 증가하고, 16에서 더 증가시켰을 때에는 시간이 조금씩 더 증가한 것을 확인할 수 있었다. 따라서 16근방에서 가장 좋은 계산결과를 얻었음을 알 수 있었다.

**과제 2**

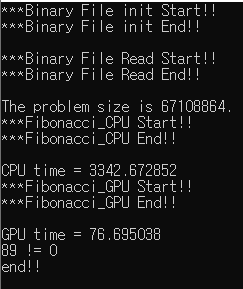
**Block size = 16**



**Block size = 32**

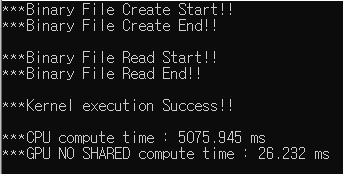


Block size = 48



블록사이즈에 따라 연산시간이 눈에 띄게 감소하였음을 확인할 수 있었다. CPU의 실행시간과 비교해도 확연히 GPU의 속도가 더 빨랐음을 확인할 수 있었다. 32 Block Size일 때 가장 시간이 빨랐다. . 블록의 수는 작업을 분할하는 기준이므로 많은 수를 지정할수록 분할되는 작업이 많아지는 걸 의미하기 때문에 블록의 수가 늘어날수록 실행 속도가 빨라진 것으로 보인다.

**과제 3**

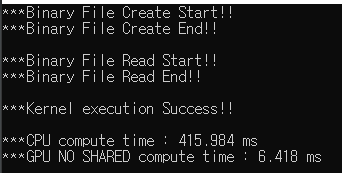


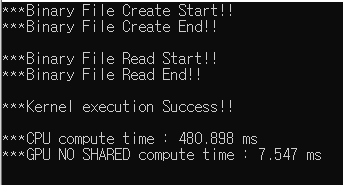
NF가 64 일 때의 수행결과는 다음과 같다. 이를 여러 번 수행하여 평균을 구해본 결과는 다음과 같았다.

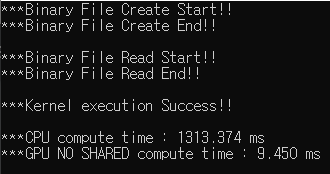
CPU: 5050.135 ms

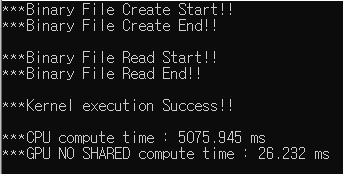
GPU: 26.330 ms

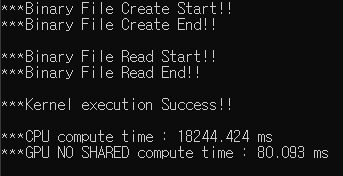
다음으로는 N의 값을 유지한 상황에서 NF의 값을 변경시켜가며 진행하였고, 그 결과는 다음과 같다.

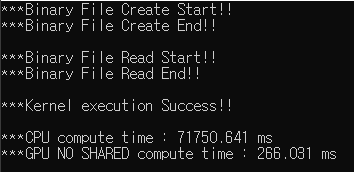
 1 일 때

4일 때

16일 때

64일 때

256일 때

1024 일 때

이를 통해, NF를 증가시킬수록 실행시간이 증가함을 알 수 있었다. 이는 NF 가 커질수록 합을 구하는 계산이 많아져 시간이 증가했다.

NF에 따라 GPU의 수행 수행시간이 어떻게 증가하였는지 경향성을 분석해 본 결과, NF의 값이 작았을 때에는 그 수행시간이 조금씩 증가한 반면, NF의 값이 커져감에 따라 NF에 비례하여 선형적으로 증가하였음을 확인할 수 있었다.

마지막으로 GPU의 결과를 Cude\_HW3\_output.bin 파일에 저장해주었다. 이 fwrite 과정에서 시간이 더 증가하였다.