**고급소프트웨어실습1 - 11주차 과제**

CUDA 프로그래밍의 기초 1

|  |  |  |
| --- | --- | --- |
| **분반** | **:** | 3 |
| **학번** | **:** | 20121635 |
| **이름** | **:** | 장종석 |

1. **실습 문제 1 – 먼저 조교가 설명하는 방식대로 자신의 PC에 CUDA 프로그래밍 시스템을 사용할 수 있는 환경을 구축하라. 다음 자신의 CUDA 환경을 이해하기 위하여 다음과 같은 내용을 파악한 후 보고서에 기술하여 제출하라.**
2. 자신이 사용하는 컴퓨터에 장착된 GPU의 기종.

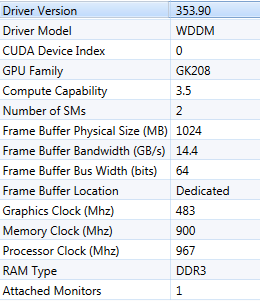


1. 현재 설치된 CUDA 시스템의 compute Capability

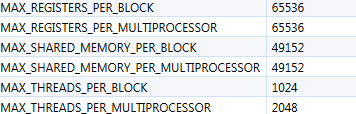
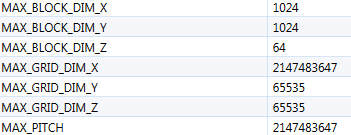


1. 현재 CUDA Compute Capability가 제공하는 각종 성능 및 스펙.

CUDA Core의 개수 : 384개, 메모리 사이즈 : 1024Mbyte



C:\Users\cspclab\Desktop\와프사이즈.PNG



1. **(실습 문제 1-2) 다음 CUDA 프로그램에 대하여 블록 크기를 다양하게 변화 시켜가면서 시간을 측정한 후, 그 결과를 보고서에 테이블로 요약하라.**

|  |  |  |
| --- | --- | --- |
| N의 크기 | CPU 수행시간(ms) | GPU수행 시간(ms) |
| 1,048,576 | 22.751162 | 1.034208 |
| 2,097,152 | 45.396667 | 2.079680 |
| 4,194,304 | 93.320000 | 4.107104 |
| 8,388,608 | 181.832092 | 8.637344 |
| 16,777,216 | 362.189240 | 17.050304 |
| 33,554,432 | 709.435852 | 33.856449 |

결과에 따라 N의 크기가 증가하면 수행 시간이 늘어남을 알 수 있다.   
현재 GPU의 크기는 메모리의 크기가 1GB이다. 이는 Byte으로써 수치적으로 계산하면 한 변수당 가질 수 있는 변수의 개수는 다음과 같다. 이는 동일한 개수의 변수가 3개 있는 것과 각 변수가 float 형이므로 4Byte의 크기를 갖는다는 점에서 계산하면 위와 같은 수식을 얻을 수 있다. 실제로도 개의 개수로는 메모리에 각 변수들을 저장 할 수 있었지만 개의 개수는 메모리에 저장 할 수 없었다.

|  |  |  |
| --- | --- | --- |
| BLOCK\_SIZE | CPU(ms) | GPU(ms) |
| 1 | 22.870401 | 73.604065 |
| 2 | 22.630112 | 19.847872 |
| 4 | 22.598415 | 5.519296 |
| 8 | 22.269676 | 1.553344 |
| 16 | 22.367483 | 1.043456 |
| 32 | 22.672373 | 1.064960 |

위의 데이터는, N의 크기가 1,048,576 일때 측정한 데이터 이다. 위의 데이터를 보게되면 C함수를 이용한 경우, BLOCK\_SIZE 에 무관하게 비슷한 시간을 나타내는 것을 알 수 있으며, GPU를 이용한 경우, BLOCK\_SIZE 가 16일 때 가장 좋은 효율을 내는 것으로 확인 되었다. 또한 이 경우, 약 20배 정도의 성능 이득을 보았다.

1. **(실습 문제 1-3) 다음 이 문제를 해결해주는 CUDA 커널 프로그램을 작성한 후 가급적 정확하게 GPU 수행 시간을 측정하라. 앞의 문제에서와 같이 쓰레드 블록의 크기를 변화시켜가면서 수행 시간 관점에서 CPU 방법과 비교 분석한 후, 그 결과를 보고서에 기술하라.**

. 실습 시간에 사용한 코드에 몇 가지 연산을 추가한 후 시간을 측정하였다. 사용한 표본(N)의 크기는 1,048,576()개 이며, 표에 기재한 수행 시간은 5회 측정한 수행 시간의 평균값이다.

|  |  |  |
| --- | --- | --- |
| BLOCK\_SIZE | CPU(ms) | GPU(ms) |
| 8 | 675.727966 | 1220.742188 |
| 16 | 681.713440 | 1225.402344 |
| 32 | 694.512512 | 1262.627808 |

위의 결과를 보게 되면 BLOCK\_SIZSE의 값에 따른, 속도의 연관성이 없는 것으로 나타났다. 실험을 위하여, BLOCK\_SIZE가 1, 2, 4 일 때도 측정을 해보았지만, 정상적인 결과가 나오지 않았다. 이를 확인해본 결과, BLOCK\_SIZE가 1,2, 4인경우, grid Y의 최대값인 2^16이 넘어가기 때문에 정상적인 값이 도출되지 않음을 알 수 있었다. 이와 비슷하게 BLOCK\_SIZE 의 값이 64가 되었을 경우, BLOCK 의 사이즈가 맞지 않아서 값을 구하지 못하였다.

1. **(숙제 1) 이제 조교가 지정한 N\_EQUATIONS 값에 대하여 자신의 CUDA 프로그램이 가장 빠른 속도를 보이는 블록의 크기를 실험적으로 결정한 후 그 값을 보고서에 기술하라.  
   위의 C함수와 자신의 CUDA 커널 프로그램의 속도를 가급적 정확히 측정하여 그 결과를 보고에서 비교 분석하여라**

|  |  |  |
| --- | --- | --- |
| BLOCK\_SIZE | CPU(ms) | GPU(ms) |
| 1 | 22.329145 | 50.335617 |
| 2 | 20.415581 | 14.718432 |
| 4 | 20.926046 | 4.598592 |
| 8 | 21.468811 | 2.609152 |
| 16 | 21.135847 | 2.243392 |
| 32 | 20.917894 | 2.206976 |

위 표의 경우 BLOCK\_SIZE 의 값을 변화 시키면서, 측정한 결과이다. 표의 결과를 보게 되면, CPU연산의 경우 BLOCK\_SIZE 에 상관없이 대체로 일정한 값을 나타내는 것을 알 수 있다. 이에 반해 GPU 연산의 경우 BLOCK\_SIZE의 연산에 영향을 받는데, BLOCK\_SIZE 의 값이 32 일 때 가장 빠른 것으로 측정이 되었다. 따라서, BLOCK\_SIZE값이 32 인 경우, 5차례 실험을 진행하였다.

|  |  |  |
| --- | --- | --- |
|  | CPU(ms) | GPU(ms) |
| 1차 | 20.749149 | 2.937824 |
| 2차 | 23.192196 | 2.531616 |
| 3차 | 21.615520 | 2.419072 |
| 4차 | 21.357721 | 2.399136 |
| 5차 | 21.416588 | 2.259968 |
| 평균 | 21.6662348 | 2.509523 |

위 표의 경우 BLOCK\_SIZE 를 32로 하여 5차례 비교를 한 후 평균을 낸 값이다. CPU와 C PU 모두 차수마다 어느정도 오차가 발생한 것을 확인할 수 있었다.

1. **(숙제 2) – 먼저 본 수업 제공 코드에 기반을 둔 CPU 코드를 작성한 후 처리 속도를 가급적 신빙성 있게 측정하라. (당연히 Visual Studio에서 Release mode를 사용할 것.)**

C함수와 CUDA 커널프로그램 속도를 비교해보도록 하자.

|  |  |  |
| --- | --- | --- |
| BLOCK\_SIZE | CPU(ms) | GPU(ms) |
| 2 | 2986.323975 | 1773.334351 |
| 4 | 3007.726074 | 478.942230 |
| 8 | 2980.628906 | 128.797638 |
| 16 | 2996.549316 | 72.652832 |
| 32 | 2989.113281 | 79.260605 |

위 표의 경우 3차례의 실험을 통한 평균값을 기록하였다. 실습1-3의 경우 N의 값이 67108864 (2^13)으로 되어있다. 이 경우도 위에서와 마찬가지로, 이 또한 BLOCK의 SIZE가 1일 경우, 정상적인 값을 구하지 못하는 것을 확인하였고, 해당 이유는 위와 동일하게 GRID MAX y의 값이 초과되었기 때문으로 생각된다. 위의 실험 또한 BLOCK\_SIZE 에 64를 입력하게 되면, 문제가 발생하는 것을 알 수 있었는데, 이는 전체 BLOCK 사이즈가 초과되기 때문이다.  
전체적인 성능을 비교해보게 되면, C함수를 사용할 경우에는 BLOCK\_SIZE에 상관없이 비슷한 시간대로 계산을 한 것을 확인할 수 있으며, GPU 를 사용한 경우에는, BLOCK\_SIZE 값이 16인경우 최대의 효율을 나타냄을 알 수 있었다  
BLOCK\_SIZE 가 16 일때, CPU와 GPU의 속도를 비교하게 되면, GPU가 대략 42배 정도 효율이 좋은 것으로 나타나고 있다.