고급소프트웨어실습1 11주차 과제 20161565 권기윤

실습문제2

1. 블럭 크기에 따른 수행 시간 변화가 있는지 확인하고 자신이 발견한 사항을 보고서에 기술하라

* 텍스트이(가) 표시된 사진

  자동 생성된 설명
* 텍스트이(가) 표시된 사진

  자동 생성된 설명텍스트이(가) 표시된 사진

  자동 생성된 설명
* 텍스트이(가) 표시된 사진

  자동 생성된 설명
* 텍스트이(가) 표시된 사진

  자동 생성된 설명
* Data Size가 32^5일때부터 overflow가 발생하여 제대로 된 값이 나오지 않는 것을 확인 할 수 있다. Data size를 증가시키면서 CPU time은 exponential 하게 증가하지만, GPU는 CPU보다 증가폭도 적고, 더 빠른 시간에 연산을 수행한다는 것을 확인할 수 있다.

실습문제 3

이 문제를 해결해주는 CUDA 커널 프로그램을 작성한 후 가급적 정확하게 GPU 수행 시 간을 측정하라. 앞의 문제와 같이 쓰레드 블록의 크기를 변화시켜가면서 수행 시간 관점 에서 CPU 방법과 비교 분석한 후, 그 결과를 보고서 기술하라.

* 실제 블록 크기는 BLOCK\_SIZE x BLOCK\_SIZE로, BLOCK\_SIZE는 블록의 한 변의 크기를 나타낸다. 각 데이터 크기에 대하여 쓰레드 블록의 크기를 변화시켜 실험한 결과는 아래와 같다.
* 
* 
* 
* 
* <block size = 32, n = 1 ~ 32786>
* 
* <block size = 1, n = 1024>
* 
* <block size = 16, n = 1024>
* 
* <block size = 32, n = 1024>
* 위 결과로 볼 수 있듯이, 쓰레드 블록의 크기가 작아진다면 성능 하락이 발생하고, 쓰레드 블록의 크기를 적절하게 설정하였을 때 가장 높은 성능을 보였다. 위 실험에서 가장 좋은 성능을 보인 block size는 16이다. 또한 모든 실험에서 CPU로 실행한 결과보다 압도적인 속도 차이를 보여주었다.
* 쓰레드 블록의 크기를 32보다 크게 설정하지 않은 이유는 ELEM\_PER\_VECTOR=32로 설정되어있기 때문에 32보다 크게되면 Mat\_Vec\_Multiply\_kernel() 함수에서 row와 col의 주어진 계산구조를 모두 바꾸어 구성해야 한다. 따라서 구조 수정 없이 실험가능한 block size의 최대 크기는 32이다.

숙제 1

(ii)조교가 지정한 N\_EQUATIONS 값에 대하여 자신의 CUDA 프로그램이 가장 빠른 속도를 보이는 블록의 크기를 실험적으로 결정한 후 그 값을 보고서에 기술하라.

<N\_EQUATIONS = 2^26 고정>

텍스트이(가) 표시된 사진

자동 생성된 설명







텍스트이(가) 표시된 사진

자동 생성된 설명







실제 block size는 코드에서 지정한 으로 BLOCK\_SIZE 블록의 한 변의 크기를 나타낸다. 따로 지정한 N\_EQUATIONS 값이 따로 없기 때문에 int형에서 최대 양수 크기인 2^26으로 설정하여 실험하였다. 각 블록 크기 별로 실험한 결과는 위와 같다.

위의 결과를 통해 스레드 블록의 크기가 너무 작아지거나 너무 커진다면 성능 하락이 발생하는 것을 확인 할 수 있다. 블록크기 2^5에서 가장 좋은 성능을 보임을 확인할 수 있다. 또한 GPU 중에서 가장 낮은 성능을 보여도 CPU실행결에 비해 압도적인 성능차이를 보여주었다.

숙제2

(i) 먼저 본 수업 제공 코드에 기반을 둔 CPU 코드를 작성한 후 처리 속도를 가급적 신빙성 있게 측정하라

(ii) 다음 이에 대응하는 CUDA 프로그램를 작성한 후 다양한 크기의 블럭에 대하여 속도를 측정한 후, CPU 기반 코드에 비해 얼마나 성능이 향상이 되는지 분석하라

(iii) 위의 CPU 코드와 CUDA 코드로 실험한 내용을 자신의 분석 결과와 함께 보고서에 명확히 기술하라

* 과제2의 각 BLOCK\_SIZE의 크기에 대해 실험한 결과는 아래와 같다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

|  |  |  |
| --- | --- | --- |
| BLOCK\_SIZE | CPU | GPU |
| 16 | 2467.733643 | 10.920960 |
| 32 | 2362.372803 | 5.411840 |
| 64 | 2567.562012 | 4.959232 |
| 256 | 2379.741211 | 4.960256 |
| 1024 | 2295.614746 | 5.178368 |

숙제1과 마찬가지로, 쓰레드 블록의 크기가 너무 작거나 너무 크다면 성능의 하락이 발생함을 알 수 있다. 또한, BLOCK\_SIZE = 2^6에서 가장 높은 성능을 보임을 확인할 수 있다. 따라서 적절한 크기의 쓰레드 블록 크기를 설정해야 함을 확인 할 수 있다.

숙제3

(i) 먼저 for 문장을 사용하여 순차적으로 n번의 동일한 계산을 반복적으로 수 행하는 C/C++ 함수를 작성한 후 CPU 수행 시간을 측정하라. 시간 측정 방 법은 이전 실습 시간에 사용한 방법을 사용하고, 가급적 정확한 시간 측정을 위하여 여러 번 반복적으로 함수를 수행시킨 후 평균값을 취하라.

(ii) 다음 이 문제를 해결해주는 CUDA 커널 프로그램을 작성한 후 가급적 정확하게 GPU 수행 시간을 측정하라. 여러 n과 nf 값에 대하여 쓰레드 블럭의 크기를 변화시켜가면서 수행 시간 관점에서 CPU 방법과 비교 분석한 후, 그 결과를 보고서 기술하라.

* 첫번째 실험은 nf값을 고정하고 블록 크기를 변화해가며 실험하였다. 블록의 크기는 BLOCK\_WIDTH \* BLOCK\_HEIGHT이다.

텍스트이(가) 표시된 사진

자동 생성된 설명





텍스트이(가) 표시된 사진

자동 생성된 설명

실험 결과 위 숙제 1,2의 실험과 같이 쓰레드의 크기가 너무 작거나 크다면 성능의 하락이 발생함을 확인 할 수 있었다. 또한 적절한 크기의 쓰레드 블록 설정시에 가장 좋은 성능을 보임을 확인할 수 있다. 실험결과 NF\_SIZE가 64로 고정되었을 때 BLOCK\_SIZE는 256 ~ 1024에서 가장 높은 성능을 보인다.

* 두번째 실험은 블록 크기를 고정하고 nf값을 변화해가며 실험하였다. 블록의 크기는 BLOCK\_WIDTH \* BLOCK\_HEIGHT이다.

(1)

(2)

(3)

(4)

(5)

(6)

NF\_SIZE가 증가하면서 CPU compute time은 NF\_SIZE = 256부터 기하급수적으로 증가하지만, 그에 비해 GPU compute time은 별 변화가 없음을 확인 할 수 있다. 이러한 이유는 CPU사용시 NF\_SIZE의 증가가 유미한 연산횟수의 증가로 이어지지만, GPU 사용시에는, 이미 분할된 thread에서 연산을 수행하고, 각각의 thread마다 증가하는 연산수는 크지 않고, 각 thread를 분할하고, synchronize하는데 시간이 더 걸리기 때문에 별 차이가 없는 것으로 예상된다.

