**고급소프트웨어실습1 11주차 과제**

컴퓨터공학 20172141 김미소

► Visual Studio 2019, CUDA 11.1을 사용하였습니다.

**[실습1]**

**1. GPU**

실험은 원격으로 연결하여 진행하였으며 원격 컴퓨터의 GPU는

NVIDIA GeForce GTX 1660 Ti이다.

**2. CUDA Compute Capability**

7.5

**3. CUDA Compute Capability가 제공하는 성능 및 스펙**

아키텍쳐: Turing

NVIDIA CUDA® 코어: 1536

부스트 클럭: 1770 MHz

프레임 버퍼: 6 GB GDDR6

메모리 속도: 12 Gbps

**[실습2]**

**1. 어떻게 하면 가급적 정확한 시간을 측정할 수 있을지 생각해볼 것**

텍스트이(가) 표시된 사진

자동 생성된 설명

Kernel 함수를 호출하기 전에 시간 측정을 시작하고 호출이 끝난 후 시간 측정을 끝내어 정확하게 시간을 측정할 수 있다.

**2. 이 CUDA 프로그램에 대하여 블럭 크기를 다양하게 변화시켜 가면서 시간을 측정한 후, 그 결과를 보고서에 테이블로 요약하라.**

|  |  |  |  |
| --- | --- | --- | --- |
| BLOCK\_SIZE | CPU(ms) | GPU(ms) | CPU/GPU |
| 16 | 34.305000 | 0.089472 | 383.41 |
| 32 | 0.094336 | 363.64 |
| 8 | 0.088224 | 388.83 |
| 32 이상 | Error |  |

**- 블럭 크기에 따른 수행 시간 변화가 있는지 확인하고 자신이 발견한 사항을 보고서에 기술하라.**Release mode로 수행한 결과, C/C++와 CUDA의 수행시간에서는 굉장히 큰 차이를 보였으나 CUDA 간 수행시간의 차이는 크게 두드러지지는 않았다. 그러나 CPU/GPU를 계산해보았을 때 블록 크기가 8일 때, 388.83배로 제일 높았다.

**- 한 쓰레드의 계산 시간을 길게 한 후, 블럭 크기에 따른 수행시간 변화를 분석하라.**kernel 함수 내에서 while loop를 통해 100번 계산하게 한 결과

|  |  |  |  |
| --- | --- | --- | --- |
| BLOCK\_SIZE | CPU(ms) | GPU(ms) | CPU/GPU |
| 16 | 33.658566 | 4.276224 | 7.87 |
| 32 | 3.648064 | 9.23 |
| 8 | 4.964352 | 6.78 |
| 32 이상 | Error |  |

Release mode로 수행한 결과, C/C++와 CUDA의 수행시간에서는 굉장히 큰 차이를 보였다. BLOCK\_SIZE 32일 때 CPU/GPU가 9.23으로, 다른 블록 사이즈보다 수행시간이 빨랐다.

**[실습3]**

**1. 메인 메모리에서 두 배열 데이터를 읽어들여 ⃗yi = M⃗xi에 해당하는 벡터 ⃗yi (i =0,1,···,n)를 구한 후, 그 결과를 메인 메모리에 32n개의 원소를 가지는 배열 공간에 순차적으로 저장 해주는 프로그램을 작성하라.**

텍스트이(가) 표시된 사진

자동 생성된 설명

**2. 이 문제를 해결해주는 CUDA 커널 프로그램을 작성한 후 가급적 정확하게 GPU 수행 시간을 측정하라.**

텍스트, 모니터, 스크린샷, 화면이(가) 표시된 사진

자동 생성된 설명

**- 쓰레드 블록의 크기를 변화시켜가면서 수행 시간 관점에서 CPU 방법과 비교 분석한 후, 그 결과를 보고서 기술하라.**

N=65536 정도로 놓고 실험한 결과,

|  |  |  |  |
| --- | --- | --- | --- |
| BLOCK\_SIZE | CPU(ms) | GPU(ms) | CPU/GPU |
| 32 | 41.877158 | 0.079712 | 525.36 |
| 16 | 0.081920 | 511.2 |
| 8 | 0.081184 | 515.83 |
| 64 | 0.102784 | 407.43 |
| 128 | 0.104640 | 400.2 |

Release mode로 수행한 결과, C/C++와 CUDA의 수행시간에서는 굉장히 큰 차이를 보였고 CUDA 간 수행시간에서는 블록 크기가 32일 때 CPU/GPU가 525.36으로 가장 빠른 수행시간을 보였다.

**[숙제1]**

**1. 첫 번째 두 배열 X0와 X1에는 자신이 구한 두 개의 실근 x0과 x1이 저장되어 있는데 반드시 x0 ≤ x1 조건을 만족시키도록 저장이 되어야 한다. 다음 두 번째 두 배열 FX0와 FX1에는 각각 x0과 x1을 대응하는 이차 방정식에 대입하여 계산한 함수 값을 저장해주어야 한다.**

텍스트, 모니터, 스크린샷, 닫기이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

**2. N\_EQUATIONS 값에 대하여 자신의 CUDA 프로그램이 가장 빠른 속도를 보이는 블럭의 크기를 실험적으로 결정한 후 그 값을 보고서에 기술하라.위의 C/C++ 함수와 자신의 CUDA 커널 프로그램의 속도를 가급적 정확히 측정하여 그 결과를 보고서에서 비교 분석하라.**N\_EQUATIONS = 1 << 20 즉, 1048576을 놓고 실험하였다. 참고로 dimBlock(BLOCK\_SIZE, 1);이다. 시간 측정 방법은 실습 방법과 같은 방법을 사용하였다.

|  |  |  |  |
| --- | --- | --- | --- |
| BLOCK\_SIZE | CPU(ms) | GPU(ms) | CPU/GPU |
| 32 | 6.89044 | 0.126976 | 54.27 |
| 16 | 0.185760 | 37.09 |
| 8 | 0.319360 | 21.58 |
| 64 | 0.125920 | 54.72 |
| 128 | 0.123808 | 55.65 |

Release mode로 수행한 결과, C/C++와 CUDA의 수행시간에서는 큰 차이를 보였고 CUDA 간 수행시간에서는 차이가 월등하게 나는 것은 아니지만 블록 크기가 128일 때 CPU/GPU가 55.65로 가장 빠른 수행시간을 보였다.

**[숙제2]**

**1. 입력 파일에서 n 값과 n개의 양의 정수 값을 배열 x[]로 읽어 들여 각 x[i]에 대한 피보나치 수 Fx[i]를 계산하여 출력 배열 y[i]에 저장을 한 후, 배열 y[]를 출력 파일 y.binary에 저장을 해주는 프로그램을 작성하라.**

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

**2. 먼저 본 수업 제공 코드에 기반을 둔 CPU 코드를 작성한 후 처리 속도를 가급적 신빙성 있게 측정하라.**

**다음 이에 대응하는 CUDA 프로그램을 작성한 후 다양한 크기의 블록에 대하여 속도를 측정한 후, CPU 기반 코드에 비해 얼마나 성능이 향상이 되는지 분석하라.위의 CPU 코드와 CUDA 코드로 실험한 내용을 자신의 분석 결과와 함께 보고서에 명확히 기술하라.**

N = 1 << 26 즉, 67108864으로 놓고 실험한 결과

|  |  |  |  |
| --- | --- | --- | --- |
| BLOCK\_SIZE | CPU(ms) | GPU(ms) | CPU/GPU |
| 32 | 1067.521801 | 6.588608 | 162.03 |
| 16 | 12.130720 | 88.00 |
| 8 | 22.714720 | 47.00 |
| 64 | 4.462880 | 239.2 |
| 128 | 4.493248 | 237.5 |

Release mode로 수행한 결과, C/C++와 CUDA의 수행시간에서는 굉장히 큰 차이를 보였고 CUDA 간 수행시간에서는 블록 크기가 64일 때 CPU/GPU가 239.2로 가장 빠른 수행시간을 보였다. 다만 블록 크기가 128일때와는 큰 차이를 보이진 않았다.

**[숙제3]**

**1. 먼저 for 문장을 사용하여 순차적으로 n번의 동일한 계산을 반복적으로 수행하는 C/C++ 함수를 작성한 후 CPU 수행 시간을 측정하라.**

**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.**

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

만약 왼쪽이나 오른쪽에 충분한 개수의 원소가 없을 경우 합이 가능한 원소들까지만 더한다고 하였으므로 index가 0보다 크고 N보다 작을 경우에만 값을 더하면 된다. 따라서 그렇게 구현하였다.

N\_SIZE는 1 << 26 즉, 67108864이고 NF\_SIZE는 1 << 6 즉, 64이다. 이를 바탕으로 실험을 진행한 결과로

|  |  |  |  |
| --- | --- | --- | --- |
| BLOCK\_SIZE | CPU(ms) | GPU(ms) | CPU/GPU |
| 64 | 21339.0604 | 243.740 | 87.55 |
| 32 | 434.722 | 49.09 |
| 16 | 875.527 | 24.37 |
| 128 | 234.966 | 90.82 |
| 256 | 248.033 | 86.03 |

Release mode로 수행한 결과, C/C++와 CUDA의 수행시간에서는 굉장히 큰 차이를 보였다. 블록 크기가 줄어들수록 수행시간이 줄어드는 것을 확인하여 블록 크기가 8일때를 따로 체크해보지 않았다. 그 대신 블록크기가 256일 때의 수행시간을 체크해보았다. 그러나 블록 크기가 128일 때보다 오히려 더 느려진 것을 확인할 수 있었다. 즉, CUDA 간 수행시간에서는 블록 크기가 128일 때 CPU/GPU가 90.82로 가장 빠른 수행시간을 보였다.