멀티코어 컴퓨팅 숙제 5

2013-11392 김지현

CPU, GPU를 어떻게 고를지는 아래 코드의 인자를 바꾸면 된다.

cl\_device\_id device;

clGetDeviceIDs(platform, **CL\_DEVICE\_TYPE\_GPU**, 1, &device, NULL);

clGetDeviceIDs(platform, **CL\_DEVICE\_TYPE\_CPU**, 1, &device, NULL);

# 문제 1

10,000\*10,000 행렬곱을 기준으로, 성능은 아래와 같았다.

CPU : 80.847초

GPU : 10.322초

3만 이상의 크기가 입력될경우, GPU는 한번에 할당할 수 있는 버퍼의 크기가 제한되어있기 때문에 실행을 여러번으로 나눠야만 했다.

Local work size를 16으로 고정시킨 채 Global work size를 바꿨을때, 그리고 Global work size를 [10,000, 10,000]으로 고정시키고 Local work size를 바꿨을때 GPU에서의 결과는 아래와 같았다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Global work size | (1,000, 1000) | (5,000, 5000) | (10,000, 10,000) | (20,000, 20,000) |
| 성능 | 0.825554초 | 2.16576초 | 10.322초 | 250.233초 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Local work size | (1, 1) | (8, 8) | 자동 | (16, 16) |
| 성능 | 30.322초 | 18.232초 | 18.232초 | 10.322초 |

성능이 Global work size의 제곱에 비례하여 증가하는것을 볼 수 있고, Local work size의 경우 커질수록 성능이 향상되는것 을 볼 수 있었고, (16, 16) 이상 크게 키울 수 없었다.

일정 크기 이상의 계산은 GPU에서 한번에 할당할 수 있는 메모리 양이 제한되어있기 때문에, 행렬의 좌변과 우변을 각각 가로, 세로로 잘라서 각 부분을 전부 따로 순차적으로 계산해야 한다.

# 문제 2

문제에 조어진 조건 (데이터 수: 1,048,576, centroid 수: 16, 1024회 이터레이션) 아래에서 CPU와 GPU 각각에 실험했을때 결과는 아래와 같았다.

CPU : 20.901148233초

GPU : 14.474954328초