

1. 서론

본 보고서에서는 OpenCL을 통하여 다수의 GPU를 사용하여 matrix multiplication을 병렬화를 이용하여 수행하고 그 방법과 matrix multiplication을 실행해본 결과와 결과에 대한 분석을 하였다.

2. 병렬화 방법

지난 과제처럼 tiling 기법을 사용하였다. 타일 크기는 16×16 을 사용하였다. GPU를 device 배열을 통해서 4개를 읽어 와서 사용하였다. 각 GPU당 계산은 행렬 A를 row로 나누어서 같은 크기만큼 주고 B는 주어진 B를 사용하였다. kernel.cl의 경우 N,M,K를 받아서 $N \times M$ 행렬과 $M \times K$ 행렬을 곱해서 $N \times K$ 행렬을 출력하도록 (나누어서 넣어주므로 첫 행렬의 세로 길이와 두 번째 행렬의 가로 길이가 다름) 지난 과제의 kernel.cl을 수정하였다. 추가적으로 Makefile에서 -lOpenCL을 flag에 추가하여 OpenCL을 사용할 수 있게 수정하였다.

3. 실행 결과

1) N=M

```
Problem size: 2048 x 2048 x 2048
Initializing ... done!
Calculating ... done!
Validation off.
Elapsed time: 0.127 sec (126.712 ms)
```

```
Problem size: 4096 x 4096 x 4096
Initializing ... done!
Calculating ... done!
Validation off.
Elapsed time: 0.167 sec (166.656 ms)
```

```
Problem size: 8192 x 8192 x 8192
Initializing ... done!
Calculating ... done!
Validation off.
Elapsed time: 0.713 sec (712.586 ms)
```

```
Problem size: 16384 x 16384 x 16384

Initializing ... done!
Calculating ... done!
Validation off.

Elapsed time: 4.715 sec (4714.776 ms)
```

차례로 2048, 4096, 8192, 16384 개의 행과 열을 가진 두 matrix의 곱 연산에 걸린 시간이다.

2) $N > M$

```
Problem size: 8192 x 8192 x 2048

Initializing ... done!
Calculating ... done!
Validation off.

Elapsed time: 0.188 sec (187.921 ms)
```

```
Problem size: 16384 x 16384 x 4096

Initializing ... done!
Calculating ... done!
Validation off.

Elapsed time: 1.144 sec (1143.519 ms)
```

3) $N < M$

```
Problem size: 2048 x 2048 x 8192

Initializing ... done!
Calculating ... done!
Validation off.

Elapsed time: 0.250 sec (249.646 ms)
```

```
Problem size: 4096 x 4096 x 16384

Initializing ... done!
Calculating ... done!
Validation off.

Elapsed time: 0.455 sec (455.244 ms)
```

validation은 큰 수의 경우 오래 걸려서 하지 않았다. 작은 수인 704(64*11) * 448(64*7) 과 448*704, 192*320, 320*192 에 관해서 수행한 결과 맞는 것을 확인했다.

4. 결과 분석

1) GPU 1개를 사용했을 때와 시간 비교

```
Problem size: 8192 x 8192 x 8192

Initializing ... done!
Calculating ... done!
Validation off.

Elapsed time: 1.866 sec (1865.870 ms)
```

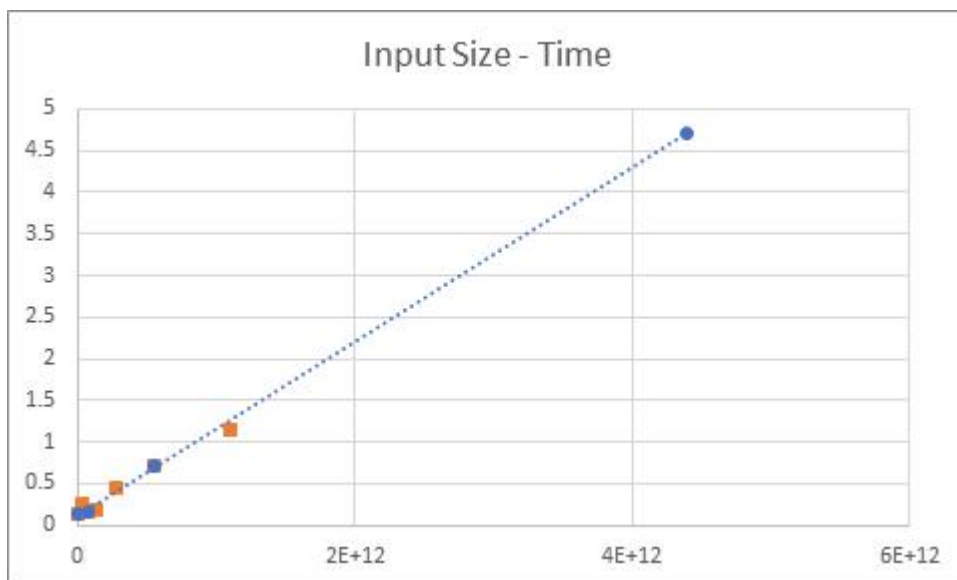
```
Problem size: 8192 x 8192 x 8192

Initializing ... done!
Calculating ... done!
Validation off.

Elapsed time: 0.713 sec (712.586 ms)
```

GPU 4개를 사용한 경우가 약 2.5배 빠르다. GPU 개수가 4배이나 4배 빠르지 않은 이유는 kernel 4개의 buffer를 연결해주는 시간과 context 한 개에서 작동하므로 context switching하는데 걸리는 시간 등이 영향을 준다 생각한다.

2) N,M에 따른 수행시간 분석



그래프는 Input Size(연산 수 = $N \times N \times M$)와 Elapsed Time을 두 축으로 하여 그렸다. 파란 점선과 파란 점들은 $N=M$ 인 경우의 결과이다. 이 점선을 기준으로 위쪽에 위치한 점들은 $N < M$ 이고 아래쪽은 $N > M$ 인 경우이다. 이로 미루어봤을 때 OpenCL은 가로(row) 방향 load에 대한 연산 속도가 세로(column) 방향 load에 대한 연산 속도보다 느리다고 추론할 수 있다.