

# 과제 #05

확장형 고성능 컴퓨팅 (2021년도 2학기, M1522.006700, M3239.002300)

Due: 2021년 11월 22일(월) 23시 59분

## 1 Warming up (15점)

- (a) (5점) 실습 서버에 장착된 GPU는 무엇인가? (a00 노드 혹은 shpc partition을 기준으로 한다.)
- (b) (5점) 실습 서버에 장착된 GPU의 single-precision floating-point 연산 성능은 검색해보면 16.31 TFLOPS 라고 나온다. 어떻게 계산된 값인지 식을 풀어서 적어라.
- (c) (5점) 실습 서버에 장착된 GPU는 CPU와 PCIe 3.0 x16 인터페이스로 연결되어 있으며 이론상 통신 성능은 15.75 GB/s이다. CPU에서 GPU로  $8192 \times 8192$  float 행렬을 전송하는데 이론상 몇 초가 걸리겠는가?

## 2 Matrix Multiplication with OpenCL (70점)

$A(M \times K) \times B(K \times N) = C(M \times N)$  행렬 곱셈을 수행하는 예시 프로그램이 주어진다. 다음은 실행 예시이다.

```
$ make
$ make performance
(...)
Result: VALID
Reference time: 11.882559 sec
Reference throughput: 92.531553 GFLOPS
Your Avg. time: 5.428968 sec
Your Avg. throughput: 202.526838 GFLOPS
(...)
```

이번 과제의 목적은 OpenCL 프로그래밍 모델로 행렬 곱셈 구현을 수정하여 성능을 높이는 것이다. 주의사항은 다음과 같다.

- `mat_mul.c`와 `kernel.cl.c`에 행렬 곱셈을 구현하고, `Makefile`에서 컴파일 옵션을 수정할 수 있다. 그 외의 파일은 제출하지 않으며 뼈대 코드를 사용하여 채점된다.
- OpenCL을 사용하여 병렬화 한다. 아직 배우지 않은 병렬화 방식(*e.g.*, CUDA)은 사용을 금한다.
- 한 노드에 GPU가 4개 장착되어 있는데, 그 중 한개만 사용한다. 또한, 하나의 노드만 사용하도록 한다. 두 노드 이상 혹은 두개 이상의 GPU를 사용하도록 코드를 수정하여 제출하는 경우 감점될 수 있다.
- `mat_mul_init` 함수에서는 계산에 앞서 하고 싶은 초기화(*e.g.*, 메모리 할당)를 할 수 있다. 이 부분은 실행 시간에 포함되지 않는다.
- 계산량을 줄이는 알고리즘(*e.g.*, Strassen, Winograd)을 사용하는 것은 허용되지 않는다. 그 외의 최적화는 대부분 허용되나, 애매할 경우 조교에게 문의하면 된다.
- 사용하는 노드 개수 이외에 `Makefile`의 옵션은 자유롭게 변경가능하다. 채점은 제출한 `Makefile` 옵션을 기준으로 진행된다.

- (a) (15점) 어떻게 병렬화 하였는지 설명하시오.
- (b) (15점) 자유롭게 결과를 분석하시오. (초기화, CPU에서 GPU으로의 데이터 전송, 행렬곱 계산, GPU에서 CPU로의 데이터 전송에 걸리는 시간은 각각 얼마인지?, 행렬곱 성능이 이론 성능 대비 몇%인지?, 데이터 전송 속도는 이론 성능 대비 몇% 인지? 등)
- (c) (20점) 다양한  $M$ (128의 배수),  $N$ (128의 배수),  $K$ (128의 배수)에 대해 정확한 답이 나와야 한다. `-v` 옵션으로 답이 맞는지 확인할 수 있으며, `make validation`로 몇가지 케이스를 테스트할 수 있다. 채점시에는 다른 케이스를 사용한다.
- (d) (20점)  $M = N = K = 8192$  기준으로 128 GFLOPS 이상은 10점, 256 GFLOPS 이상은 20점을 부여한다. `make performance`로 제출 전 성능을 테스트 해볼 수 있다.

### 3 Multi-GPU (15점)

자유롭게 애플리케이션을 정하여 여러 개의 GPU를 사용해서 병렬화 해보자. GPU를 1, 2, 3, 4개 사용했을때의 성능 차이를 위주로 결과를 분석하시오.

(Hint: 마땅히 떠오르는 애플리케이션이 없으면 길이  $N$ 인 벡터 2개를 더하는 vector addition이 무난하다.)

- 두개 이상의 GPU를 사용하고자 할 경우 SLURM 작업 실행 명령에 `--gres=gpu:<필요한 GPU 개수>` 를 주도록 한다. 실제 사용 예시는 `mat_mul` 디렉토리의 `Makefile`을 참고한다.

### 4 Submission Instruction

- `mat_mul` 디렉토리, `multi_gpu` 디렉토리, `report.pdf`를 한 파일로 압축하여 ETL에 제출한다.
- `mat_mul` 디렉토리에는 `mat_mul.c`, `kernel.cl.c`, `Makefile`을 넣는다. 다른 파일은 있더라도 제외하고 채점된다.
- `multi_gpu` 디렉토리에는 구현한 파일들을 모두 넣는다. 실제로 실험을 했는지 확인하기 위한 용도이며, 조교가 실행해 보지는 않을 것이다.
- 첨부 파일명은 계정이름\_HW05.zip으로 한다. (e.g., shpc000\_HW05.zip)
- 채점은 프로그램에 의해 기계적으로 처리되므로 위 사항을 지키지 않은 경우 누락되거나 불이익을 받을 수 있다.
- Grace day를 사용하려면 본인이 과제를 제출한 날에 조교에게 메일(`shpc21@aces.snu.ac.kr`)로 알려야 한다. 메일 없이 제출만 한 경우 다음 과제를 위해 아낀 것으로 판단, 미제출 처리된다. 또한, grace day 사용 시에도 과제 제출은 이메일이 아닌 ETL을 통해 해야한다.