

# 과제 #04

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

Due: 2021년 11월 15일(월) 23시 59분

## 1 MPI Test (20점)

- (a) (5점) 실습 서버의 노드들은 InfiniBand라는 네트워크 기술을 이용하여 연결되어 있으며, 각 노드에는 100Gbit/s의 통신 성능을 지원하는 ConnectX-5 네트워크 카드가 장착되어 있다. 한 노드에서 다른 노드로 1GByte 데이터를 전송하려면 이론상 몇 초가 걸리겠는가?  
(Hint: Bit와 Byte를 혼동하지 않도록 주의하자. 단방향 통신만 고려한다.)
- (b) (15점) 두 노드 사이에 1GB 데이터를 10번 주고받는 프로그램 pingpong을 구현하라. pingpong 디렉토리에 예시 프로그램이 주어진다. 다음은 실행 예시이다.

```
$ make
$ make performance
(...)
[rank 0] Ready to communicate...
[rank 1] Ready to communicate...
Elapsed time: 0.000005 sec
(...)
```

A노드에서 B노드로 데이터를 보내고 이후 B 노드에서 받은 데이터를 A 노드로 보내는 것을 1번이라고 친다. 전송한 데이터양을 시간으로 나누면 전송 속도를 얻을 수 있다. 전송 속도가 몇 Gb/s로 측정되는가? 이론값에 근접한가?

## 2 Matrix Multiplication with MPI (80점)

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

```
$ make
$ ./main -h
(... help message for all supported options ...)
$ make performance
(...)
[rank 0] Initializing...
[rank 0] Initializing done!
[rank 1] Initializing...
[rank 1] Initializing done!
(...)
Validating...
Result: VALID
Reference time: 9.942481 sec
Reference throughput: 110.587248 GFLOPS
```

Your Avg. time: 977.052038 sec  
Your Avg. throughput: 1.125335 GFLOPS  
(...)

이번 과제의 목적은 `mat_mul.c`에 있는 행렬 곱셈 구현을 수정하여 성능을 높이는 것이다. 주의사항은 다음과 같다.

- `mat_mul.c`에 행렬 곱셈을 구현하고, `Makefile`에서 컴파일 옵션을 수정할 수 있다. 그 외의 파일은 제출하지 않으며 뼈대 코드를 사용하여 채점된다.
- MPI와 지금까지 배운 병렬화 방법(*e.g.*, OpenMP)을 이용하여 병렬화 한다. 아직 배우지 않은 병렬화 방식(*e.g.*, CUDA)은 사용을 금한다.
- `mat_mul_init` 함수에서는 계산에 앞서 하고 싶은 초기화(*e.g.*, 메모리 할당)를 할 수 있다. 이 부분은 실행 시간에 포함되지 않는다.
- Rank가 0인 프로세스에만 입력 행렬 A, B가 주어짐에 주의한다. 다른 프로세스에는 입력 행렬이 할당조차 되지 않으므로 A, B 행렬을 접근하는 경우 오류가 발생한다. 그러므로 rank가 0이 아닌 프로세스에서는 적절히 메모리를 할당하고 rank가 0인 프로세스로부터 행렬의 일부 또는 전체를 전달받아야 한다.
- 마찬가지로, 출력 행렬 C는 rank가 0인 프로세스에만 할당되며 계산 결과를 rank가 0이 아닌 프로세스로부터 전달받아야 한다. `-v` 옵션을 통한 결과 검증 또한 rank가 0인 프로세스에서 이루어진다.
- 계산량을 줄이는 알고리즘(*e.g.*, Strassen, Winograd)을 사용하는 것은 허용되지 않는다. 그 외의 최적화는 대부분 허용되나, 애매할 경우 조교에게 문의하면 된다.
- `Makefile`의 옵션은 자유롭게 변경하되 사용할 노드 갯수를 정하는 옵션인 `--nodes`의 값은 2를 넘기지 않도록 한다. 최대 사용할 수 있는 노드 갯수는 2개로 고정한다. 그 외에 `--ntasks-per-node` 옵션을 조절하여 프로세스 수를 늘려도 되고, 통신 관련 옵션을 수정해도 된다. 채점은 제출한 `Makefile` 옵션을 기준으로 진행된다.

- (20점) 어떻게 병렬화 하였는지 설명하시오.
- (20점) 자유롭게 결과를 분석하시오. (노드 1개를 사용했을 때와 노드 2개를 사용했을 때의 성능차는 얼마나 되는가? 등)
- (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.
- (20점) 노드 2개,  $M = N = K = 8192$  기준으로 64 GFLOPS 이상이 나와야 한다. `make performance`로 제출 전 성능을 테스트 해볼 수 있다.

### 3 Submission Instruction

- `pingpong` 디렉토리, `mat_mul` 디렉토리, `report.pdf`를 한 파일로 압축하여 ETL에 제출한다.
- `pingpong` 디렉토리에는 `pingpong.c`만 넣는다. 실제로 실험을 했는지 확인하기 위한 용도이며, 조교가 실행해 보지는 않을 것이다.
- `mat_mul` 디렉토리에는 `mat_mul.c`와 `Makefile`을 넣는다. 다른 파일은 있더라도 제외하고 채점된다.
- 첨부 파일명은 계정이름 `HW04.zip`으로 한다. (*e.g.*, `shpc000.HW04.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.