# 과제 **#2**

2022-29677 한주희

## 1 Theoretical Peak Performance of CPU

(a) Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz

(b) 2

(c) base clock frequency는 2.10GHz, boost clock frequency는 3.20GHz이다. 워크로드에 따라 clock frequency를 다르게 하여 CPU를 작동시키기 위해 두 종류가 존재한다. 무거운 워크로드를 처리하는 경우 boost clock frequency로 작동한다.

(d) physical 코어는 16개, logical 코어는 32개이다. floating-point 연산에서 스레드가 functional unit을 공유하기 때문에 실제 성능을 계산할 때는 physical 코어 개수를 사용하는 것이 더 정확하다.

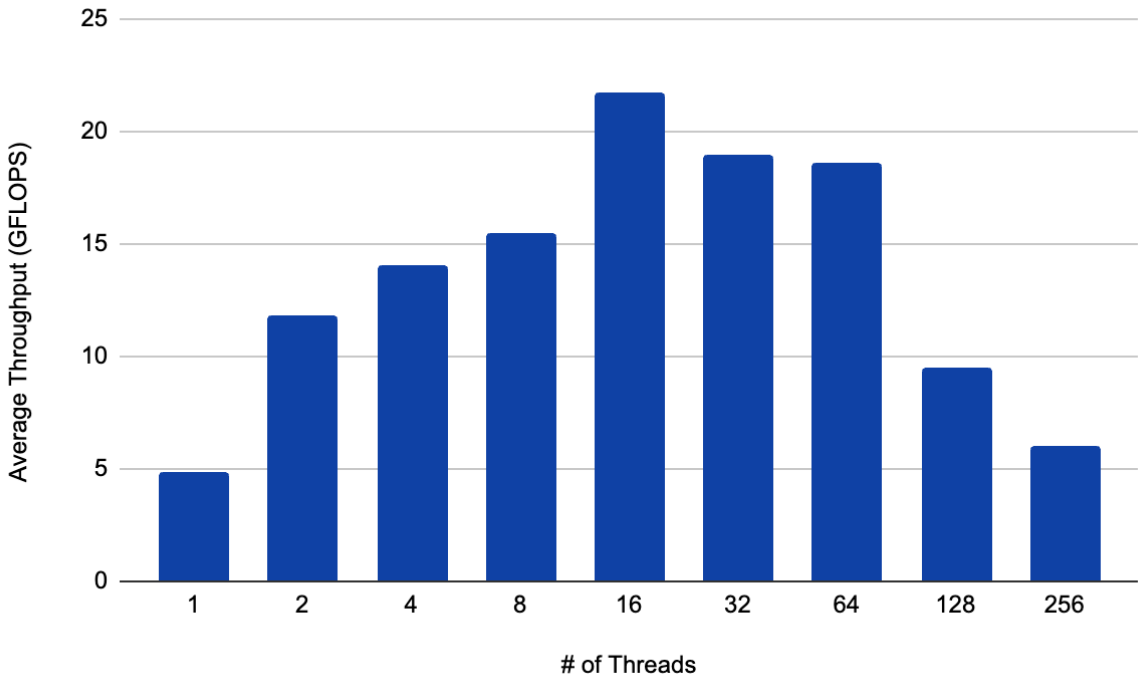
(e) 각 코어는 FP32 연산을 16개 수행할 수 있다.

(f) $R_{peak}$ = CPU 개수 × CPU 하나의 physical 코어 개수 × base clock frequency × (FLOPs / cycle)
= 2 × 16 × 2.10GHz × 64 = 4300.8 GFLOPS

## 2 Matrix Multiplication using Pthread

- 행렬곱에서 스레드는 각기 다른 행을 계산하도록 행 단위로 작업을 분할하였다. 스레드가 계산해야 할 행의 범위는 `rows_per_thread`로 계산되며, 각 스레드는 `row_start`와 `row_end`를 통해 자신이 담당하는 행 부분을 처리하게 하였다.
- 행렬곱의 성능 개선을 위해 tiling 방식을 사용하였으며, 중첩된 for문의 순서를 `ikj`로 변경하여 메모리 접근을 연속적으로 처리하도록 최적화하였다. (i는 A의 행, k는 A의 열과 B의 행, j는 B의 열 인덱스이다.)
- 다음은 스레드를 1, 2, 4, 8, 16, 32, 64, 128, 256개로 늘리면서 행렬곱 성능을 측정한 결과이다. 성능 측정을 위해 반복 횟수는 3으로 설정하고, 평균 성능을 계산하였다. 성능은 일정한 추세로 증가하지 않는다. 스레드가 16개일 때 최대 성능을 보이고 점차 감소하여 256개에서는 성능이 급격히 저하되었다. 이는 작업을 너무 많은 스레드로 나누면, 각 스레드가 수행하는 작업의 양이 적어져 효율이 떨어지고, 스레드를 시작하고 종료하는 오버헤드가 증가하며, 자원을 공유하는 데 필요한 오버헤드가 커지면서 성능이 저하되기 때문이다.

| # of Threads | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 |
|---|---|---|---|---|---|---|---|---|---|
| Average Throughput (GFLOPS) | 4.9 | 11.8 | 14.1 | 15.5 | 21.7 | 19.0 | 18.6 | 9.5 | 6.0 |




- 0.5% 정도이다. 현재 행렬곱 구현에서 tiling 방식을 사용하여 블록의 크기를 고정된 64로 설정하였다. 최적의 성능을 얻기 위해 더 적합한 블록 크기를 찾아 적용하면, 성능을 더욱 향상시킬 수 있다.