

1

1-1 Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz

1-2 two sockets

1-3 2100/3200 (mhz) 평소에는 전력의 효율성과 안정성을 위해 낮게 작동하지만 특정 로드가 많이 요구될때에만 사용

1-4 16/32 물리/논리적 코어수 물리코어수를 기반으로 해야합니다. 논리 코어는 쓰레딩을 이용하여 코어가 쉬는 사이에 다른일을 시킬수 있지만 이상적인 상황에서는 물리코어가 언제나 일하는 상황을 이길수 없습니다.

1-5 512비트가 가능하므로  $512/32 = 16$ 개가 가능합니다.

1-6  $16 * \text{clock} * \text{socket} * \text{core} = 16 * 3.2 * 2 * 16 = 1075.4 \text{ Gflops}$

만약 FMA를 활용하여 애드와 곱을 둘다 동시에 한다고하면 2150.8이 됩니다.

2

2-1

M,N,K를 적당한 길이 간격으로 쪼개어 각 인접한 아이들끼리 우선적으로 연산되게 하여 캐시의 최적화를 달성하게 하였습니다.

각 쓰레드는 랭크에 따라 할당 받은 쓰레드 만큼만

추가적으로 AVX512를 사용하여 16개씩 작업을 할당하여 효율을 올렸습니다.

2-2

1~256 까지 10씩 올렸을때입니다.

19.54

96.55

99.64

93.97

100.91

149.65

110

64

115

45

57

77

129

30  
31  
34  
  
36  
39  
44  
49  
53  
61  
70  
83  
103  
181

으로 되는데, 볼수 있듯이 단순하게 증가만 하지 않는다. 또한 특정지점에서 급격하게 높아지는 지점이 존재하는데 이는 코어수로 정확히 나누어 질 때이다. 따라서 두가지를 알 수 있다

- 1) 단순하게 늘리기만한다고 해서 작업이 균등하게 분배되지 않는다 이처럼 작업량이 딱딱 나누어 떨어지게 된다면 코어수 또한 중요한 요소이다.
- 2) 코어수를 넘어간다면 코어배수로 늘린다하여도 드라마틱한 증가는 없다 코어수가 이 경우에는 고려해야할 중요한 요소이며 코어수 \*2 이후로는 늘리기 힘들어보인다

2-3

271 로 피크대비 27퍼센트 정도이다. 캐쉬로컬리티를 훨씬 올릴 여지가 많아 보입니다. 따라서 타일 사이즈를 더 잘게 나누어 할당하거나 더 적절한 값을 찾아야하며, M,N,K별로 다른 타일 사이즈를 적용하거나 미리 캐쉬에 로드시킬 방법이 있으면 좋을것 같습니다.