TLB 분석 보고서

20191562 김성훈

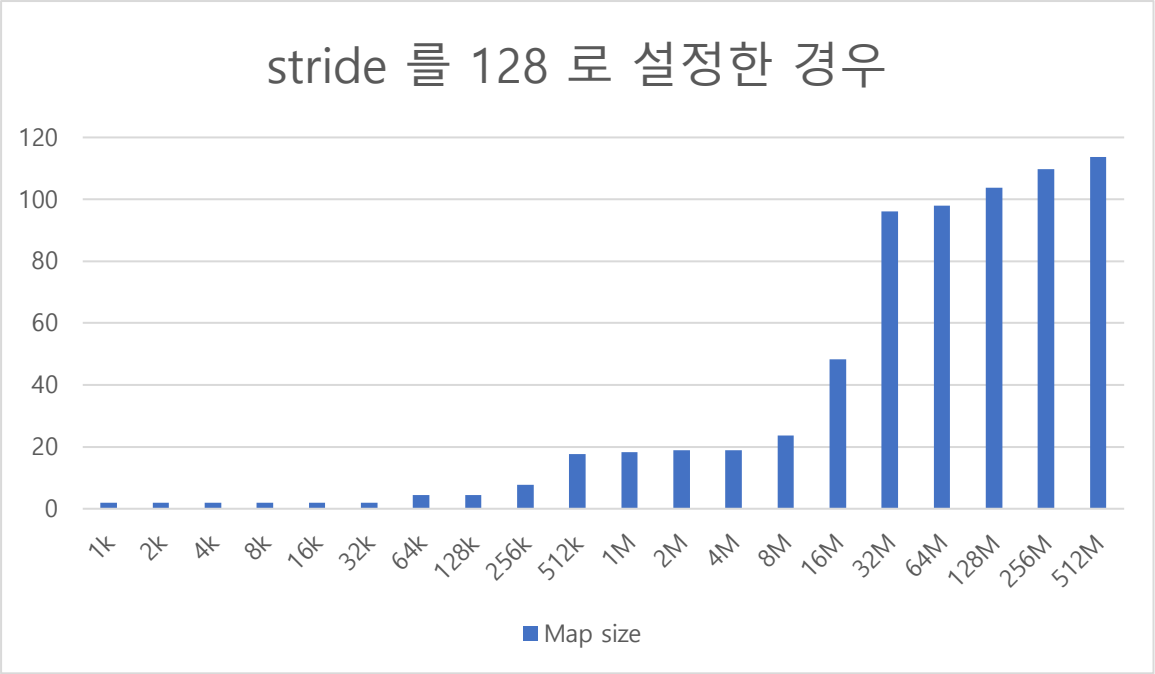
실행 환경

Ubuntu 18.04.4 LT

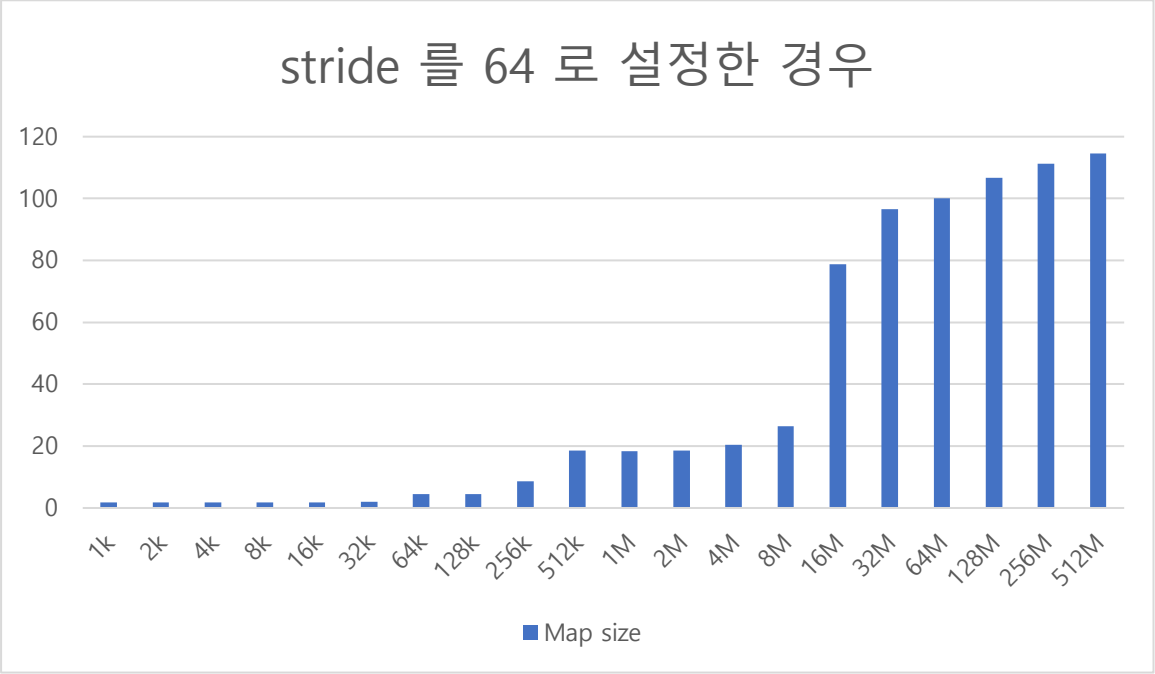
Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz, 싱글코어

CPU 캐시 크기 - 30720 KB

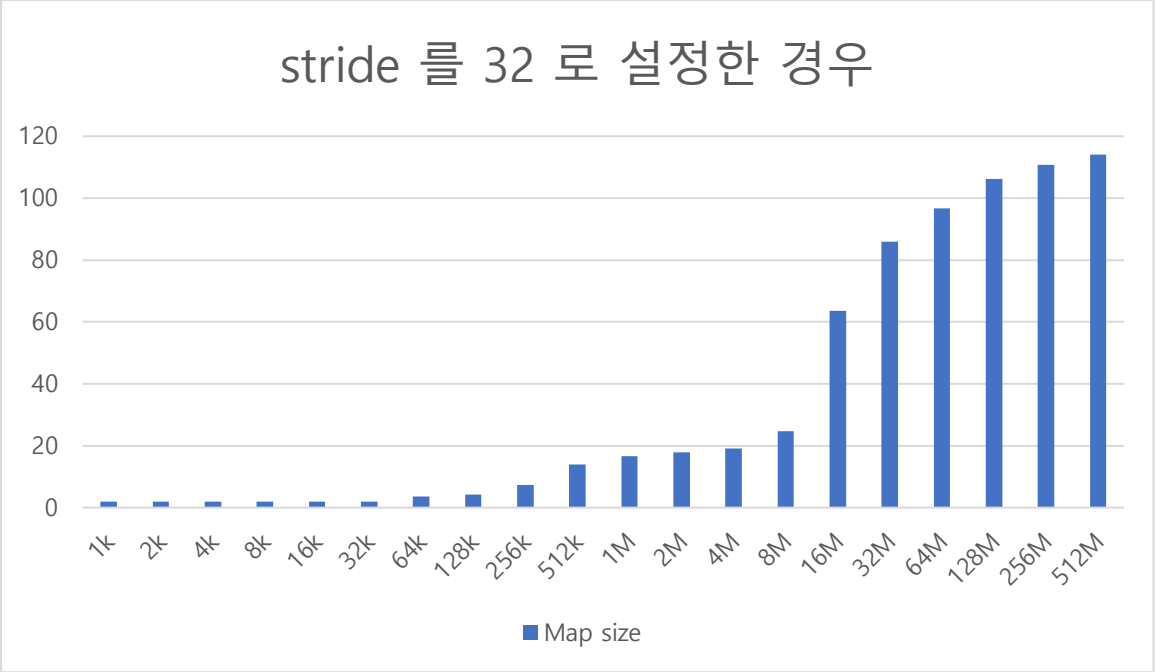
TLB 의 hierarchy 나 cache allotment 같은 것들은 결과 분석할 때 검증을 하는 과정에서 살펴보겠다.



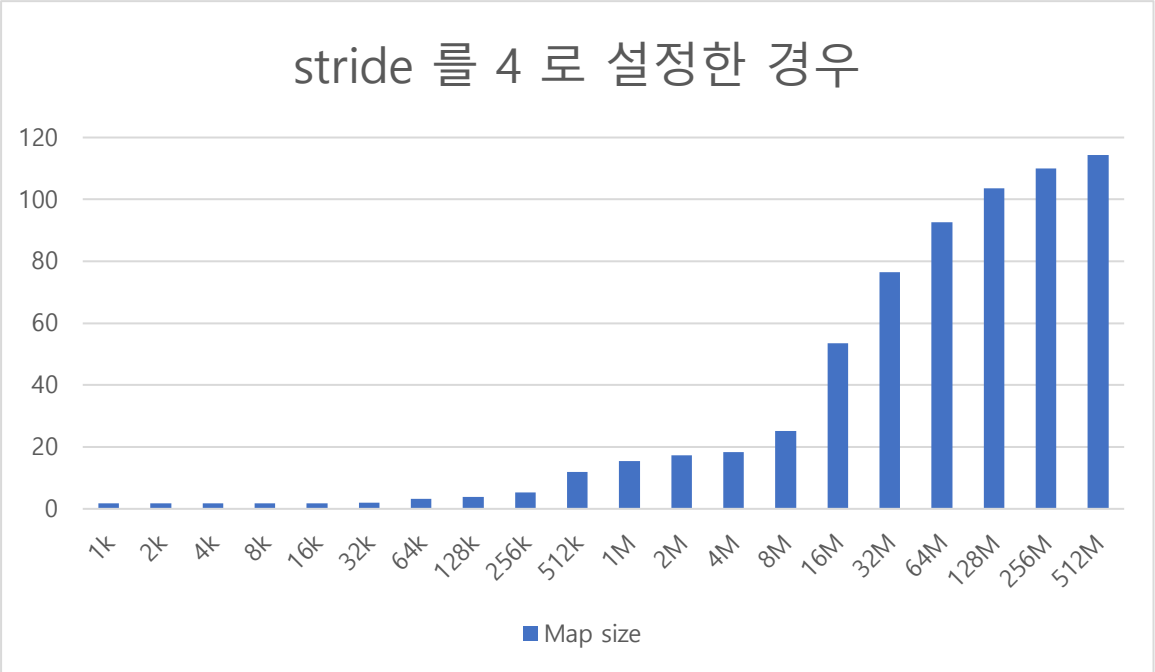
실행 코드: for i in 1k 2k 4k 8k 16k 32k 64k 128k 256k 512k 1M 2M 4M 8M 16M 32M 64M 128M 256M 512M ; do echo -n "$i, "; ./test-tlb -r $i 128; done



실행 코드: for i in 1k 2k 4k 8k 16k 32k 64k 128k 256k 512k 1M 2M 4M 8M 16M 32M 64M 128M 256M 512M ; do echo -n "$i, "; ./test-tlb -r $i 64; done



실행 코드: for i in 1k 2k 4k 8k 16k 32k 64k 128k 256k 512k 1M 2M 4M 8M 16M 32M 64M 128M 256M 512M ; do echo -n "$i, "; ./test-tlb -r $i 32; done



실행 코드: for i in 1k 2k 4k 8k 16k 32k 64k 128k 256k 512k 1M 2M 4M 8M 16M 32M 64M 128M 256M 512M ; do echo -n "$i, "; ./test-tlb -r $i 4; done

결과 분석

1. 1. stride 분석

128, 64, 32, 4 로 설정하고 다양한 Memory map 의 크기에 대해서 테스트 해봤다.
stride 는 cache line size 가 어느정도 인지 추측할 수 있게 해주는 옵션이다.
64 와 128 은 성능에 차이가 거의 없고 64 미만으로 설정한 경우만 64 이상으로 설정한 경우들보다 성능에 차이가 생기게 된다. 이 결과를 통해 cache line size 가 64 byte 에 근접하다고 추측할 수 있다. 결과는 아래에서 검증해보겠다.

1. 2. Cache line size 검증

cpuid 명령어를 이용해 각 cache level 에서 사용하는 cache line size 를 조회하면 모두 0x63 으로 나오게된다. (e.g. system coherency line size = 0x3f (63))

2. 1. Memory map size 분석

Memory map size 는 최소한 몇 단계의 TLB 가 있는지 알 수 있도록 도와주는 옵션이다.
또한 모든 테스트가 비슷한 용량 구간에서 급격한 변화를 보인다.
- 256KB -> 512KB
- 8MB -> 16MB

virtual machine 의 하드웨어의 한계로 인해 더 큰 사이즈는 분석할 수 없지만 최소한 2-Level-TLB 가 있는 것을 알 수 있다.

2. 2. Memory map size 검증

1-Level-TLB 는 4K pages, 4-way, 64 entries 로 구성돼서 4KB * 64 = 256KB 만큼 데이터를 캐싱해서 사용하는 것을 알 수 있다. 따라서 256KB 까지는 첫 데이터를 cache 로 올리는 것을 제외하고는 거의 모든 데이터가 hit 된다.

2-Level-TLB 는 4K pages, 8-way, 1024 entries 로 구성돼서 4KB * 1024 = 4MB 만큼 데이터를 캐싱해서 사용하는 것을 알 수 있다. 따라서 4MB 까지는 첫 데이터를 cache 로 올리는 것을 제외하고는 거의 모든 데이터가 hit 된다.

하지만 256KB -> 512KB, 4MB -> 8MB 의 차이가 크지 않은데 어느정도는 운으로 인해 이미 저장된 페이지를 사용할 수 있기 때문이다.