

# 2022/06/23

## clflush 성능 vs. clflushopt 성능 vs. clwb 성능

### ▼ 실험 목적

- 각 flush instruction의 PM DAX write 성능 차이를 살펴보기 위함

### ▼ 실험 구성

- 1GB file에 4KB memcpy write 후 flush를 진행하는 sequential 20GB write 워크로드
- write 워크로드는 2번 core에 bind

### ▼ 실험 결과

- clflush instruction으로 flush 진행
  - 35.984930 s
- clflushopt instruction으로 flush 진행
  - 9.944753 s
- clwb instruction으로 flush 진행
  - 9.945553 s

### ▼ 실험 결과 분석

- clflushopt 및 clwb는 성능이 거의 동일했으며, clflush는 clflushopt 및 clwb에 비해 약 3.5배 정도 성능이 안좋음을 확인할 수 있었다.

### ▼ 추가 실험

- memcpy를 하는 단위를 4KB 뿐만 아니라 1KB, 2KB, 1MB, 2MB로 다르게 해서 실험을 진행
  - 1KB
    - clflush instruction으로 flush 진행
      - 33.749936 s
    - clflushopt instruction으로 flush 진행

- 8.985184 s
  - clwb instruction으로 flush 진행
    - 8.986358 s
- 2KB
  - clflush instruction으로 flush 진행
    - 34.708267 s
  - clflushopt instruction으로 flush 진행
    - 10.119052 s
  - clwb instruction으로 flush 진행
    - 10.214165 s
- 1MB
  - clflush instruction으로 flush 진행
    - 35.409210 s
  - clflushopt instruction으로 flush 진행
    - 6.322212 s
  - clwb instruction으로 flush 진행
    - 6.295060 s
- 2MB
  - clflush instruction으로 flush 진행
    - 34.999049 s
  - clflushopt instruction으로 flush 진행
    - 6.417146 s
  - clwb instruction으로 flush 진행
    - 6.426376 s

#### ▼ 추가 실험 결과 분석

- 한 번에 memcpy로 write를 진행하는 크기에 따라 어느 정도 성능 차이가 나기는 하지만, clflush로 flush를 하는 write task의 성능이 clflushopt 또는 clwb으로 flush를 하는 write task의 성능의 1/3 수준의 성능을 갖는다는 사실은 변하지 않았음

## 다수의 mcf task LLC noisy neighbor 실험

### ▼ 실험 목적

- PM DAX I/O write task로 인해 mcf의 성능이 낮아지게 되는데, 이 성능 저하의 폭이 약 6%로 그다지 크지 않았고, 해당 이유는 실험을 진행한 PM 서버의 LLC 크기가 44MB로 매우 크기 때문이라고 판단하였음
- 이에 따라 mcf task의 수를 늘려서 LLC 경합을 증가시켰을 때 더 큰 폭의 성능 저하가 발생할 것이라는 가정을 검증하기 위해 본 실험을 구상 및 실행함

### ▼ 실험 결과

- mcf task 1개 (core 2)
  - mcf test : 13.643775 s
  - mcf train : 77.325902 s
  - mcf ref : 532.873423 s
- mcf task 4개 (core 2, 4, 6, 8)
  - mcf test : 13.855256 s
  - mcf train : 79.840096 s
  - mcf ref : 547.470586 s
- mcf task 8개 (core 2, 4, 6, 8, 10, 12, 14, 16)
  - mcf test : 14.283490 s
  - mcf train : 81.483982 s
  - mcf ref : 560.771988 s
- mcf task 1개 (core 2) + DAX I/O(clflushopt instruction without sfence) task 8개 (core 4, 6, 8, 10, 12, 14, 16, 18)

- mcf test : 13.899757 s
- mcf train : 80.058192 s
- mcf ref : 564.545805 s
- mcf task 4개 (core 2, 4, 6, 8) + DAX I/O(clflushopt instruction without sfence)  
task 8개 (core 10, 12, 14, 16, 18, 20, 22, 24)
  - mcf test : 14.228387 s
  - mcf train : 83.421368 s
  - mcf ref : 580.737915 s
- mcf task 8개 (core 2, 4, 6, 8, 10, 12, 14, 16) + DAX I/O(clflushopt instruction without sfence) task 8개 (core 18, 20, 22, 24, 26, 28, 30, 32)
  - mcf test : 14.916985 s
  - mcf train : 87.212383 s
  - mcf ref : 595.730504 s

#### ▼ 실험 결과 분석

- ref 기준 성능 분석
  - mcf task 1개 : 5.61% 성능 저하
  - mcf task 4개 : 5.73% 성능 저하
  - mcf task 8개 : 5.87% 성능 저하
- mcf task 수가 늘어날수록 성능 저하 폭이 증가하긴 하나, 미미한 폭으로 증가함  
→ mcf task와 8개의 PM DAX I/O task들이 이미 44MB의 캐시를 충분히 다 활용하고 있었던 것이 아닌가 판단됨

## DRAM LLC noisy neighbor 실험

#### ▼ 실험 목적

- clflushopt 및 clwb는 ordering을 지키지 않기 때문에 더 자주 flush를 진행하게 되고, PM은 SSD와 다르게 속도가 빠르기 때문에 computation task의 성능저하가 두드러

지게 발생하게 됨. 따라서 PM보다 더 빠른 DRAM으로 실험을 진행할 경우 오히려 더 높은 성능저하가 보일 것으로 예상되며 이를 검증하기 위한 실험.

#### ▼ 실험 결과

- mcf task 1개 (core 2)
  - mcf test : 13.643775 s
  - mcf train : 77.325902 s
- mcf task 1개 (core 2) + DAX I/O(clflushopt instruction without sfence) task 1개 (core 4)
  - mcf test : 13.690814 s
  - mcf train : 77.550261 s
- mcf task 1개 (core 2) + DAX I/O(clflushopt instruction without sfence) task 4개 (core 4, 6, 8, 10)
  - mcf test : 13.711087 s
  - mcf train : 78.193414 s
- mcf task 1개 (core 2) + DAX I/O(clflushopt instruction without sfence) task 8개 (core 4, 6, 8, 10, 12, 14, 16, 18)
  - mcf test : 13.847239 s
  - mcf train : 79.248937 s

#### ▼ 실험 결과 분석

- PM에 write를 진행한 LLC noisy neighbor 실험과 결과가 거의 동일한 것을 통해 PM 이 속도가 빠르기 때문에 flush 과정에서 noisy neighbor가 된다는 가설은 틀린 것으로 판단됨
- 하지만 DRAM을 PM으로 에뮬레이트해서 실험을 진행하였기에 이 영향이 있을 수 있다고 판단되어 DRAM에 바로 memory mapping을 한 후 flush를 진행하는 방식의 추가 실험이 필요함