

KVM 기반 VM 환경에서의 디스크 I/O 성능 분석: 볼륨 구성 및 캐시 정책에 따른 정량적 평가

차호현¹, 신주용², 남재현³¹단국대학교 모바일시스템공학과 학부생²단국대학교 인공지능융합학과 석사과정³단국대학교 컴퓨터공학과 교수

outcider112@dankook.ac.kr, juyongshin@dankook.ac.kr, namjh@dankook.ac.kr

Disk I/O Performance in KVM-based VMs: A Quantitative Study of Volume and Cache Configurations

Hohyeon Cha¹, Juyong Shin², Jachyun Nam³¹Dept. of Mobile Systems Engineering, Dankook University (Undergraduate Student)²Dept. of AI-based Convergence, Dankook University (Graduate Student)³Dept. of Computer Engineering, Dankook University (Professor)

요 약

클라우드 환경에서는 다양한 사용자에게 컴퓨팅 자원을 효율적으로 제공하기 위해 가상화 기술이 폭넓게 활용되고 있다. 그러나 가상화 계층은 디스크 I/O 경로에 추가적인 오버헤드를 발생시켜, 서비스 성능 및 시스템 응답성 저하를 초래할 수 있다. 본 연구에서는 KVM 기반 VM 환경을 대상으로, 디스크 볼륨 구성(RAID, LVM-linear, LVM-stripe)과 페이지 캐시 사용 여부가 I/O 성능에 미치는 영향을 정량적으로 분석하였다. 분석 결과는 클라우드 인프라의 디스크 및 캐시 구성 전략 수립에 있어 실질적인 참고 기준으로 활용될 수 있을 것으로 기대된다.

1. 서론

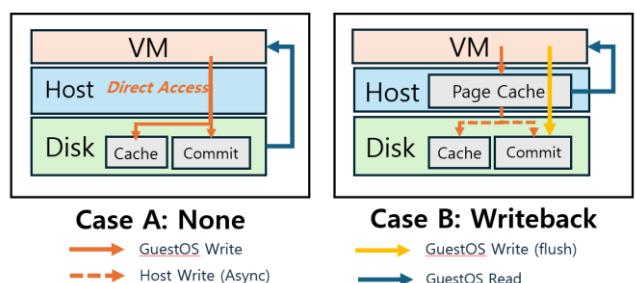
클라우드 컴퓨팅 환경에서는 가상화를 통해 다양한 사용자가 자원을 효율적으로 공유하고 활용할 수 있도록 지원한다. 특히 KVM을 기반으로 한 가상화는 고성능 서버 환경에서 널리 활용되며, 가상 머신(VM)은 물리적 디스크를 공유하는 구조로 구성된다. 그러나 가상화 계층은 I/O 경로에 추가적인 처리 단계를 도입함으로써 디스크 접근에 대한 성능 저하를 유발할 수 있다 [1]. 이러한 성능 저하는 클라우드 인프라의 효율성과 응답성을 저하시킬 수 있기 때문에, 물리 디스크 구성과 캐시 전략에 대한 정량적 이해가 필수적이다. 본 논문에서는 KVM 기반 VM 환경에서 디스크 구성 방식과 페이지 캐시 활용 여부에 따른 디스크 I/O 성능 차이를 실험적으로 분석하고, 효과적인 구성 전략을 도출하고자 한다.

2. 실험 환경 구성

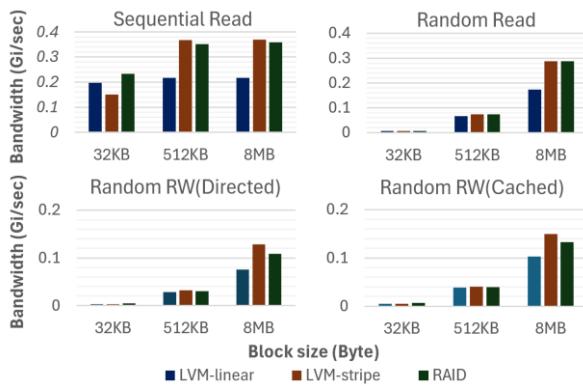
본 실험은 SATA3 인터페이스 기반의 하드 디스크(최대 처리량 220MB/s) 두 개를 이용하여 물리적 디스크 환경을 구성하였다. 디스크는 RAID 0 구성으로

해 각각 Intel VROC [2]과 LVM 기반 linear 및 stripe 방식 [3]을 이용하여 구성하였다. 실험을 위한 VM의 디스크 인터페이스로는 VirtIO SCSI single을 사용하였다.

실험 환경은 (그림 1)에서 볼 수 있듯이 페이지 캐시의 사용 여부에 따라 Case A 와 Case B 로 구분된다. Case A 는 페이지 캐시를 비활성화한 상태이며, VM의 모든 I/O 는 디스크에 직접 기록된다. 반면 Case B 는 Writeback 방식의 페이지 캐시를 활성화하여, I/O 요청이 호스트 메모리에 임시 저장된 후 비동기적으로 디스크에 반영된다. 성능 측정은 fio [4]를 이용하여 순차 읽기, 무작위 읽기, 무작위 읽기/쓰기(R:W = 50:50) 시나리오를 각각 수행하였다.



(그림 1) 페이지 캐시 활성화 여부에 따른 동작 방식



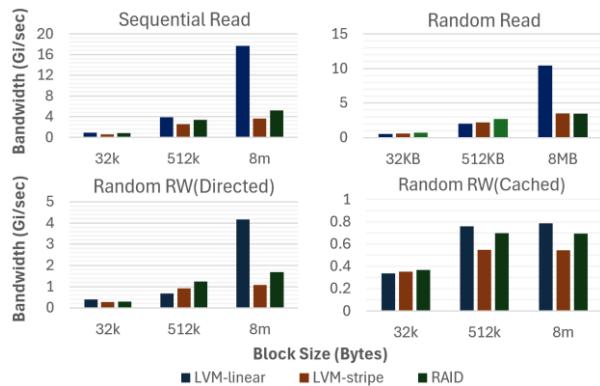
(그림 2) Case A: VM 부하 실험 (페이지 캐시 비활성화)

3. 실험 결과

Case A(페이지 캐시 비활성화 환경)에서 순차 읽기 성능은 (그림 2)에 나타나 있다. LVM-linear 구성은 블록 크기에 관계없이 평균 211MB/s의 일정한 처리량을 보였으며, LVM-stripe 구성은 32KB에서는 151MB/s로 LVM-linear 대비 약 28% 낮은 처리량을 보였으나, 블록 크기가 증가함에 따라 최대 368.5MB/s로 향상되며 SATA3 인터페이스의 이론적 최대치에 근접하였다. RAID 구성은 소블록 크기(32KB)에서도 233MB/s로 우수한 성능을 보였고, 이후 평균 355MB/s 수준의 안정적인 성능을 유지하였다.

무작위 읽기 및 무작위 읽기/쓰기 환경에서는 모든 구성에서 성능 저하가 발생하였다. 32KB 블록에서 무작위 읽기 성능은 약 68MB/s로, 순차 읽기 대비 평균 70% 이상 낮은 처리량을 기록하였다. 무작위 읽기/쓰기의 경우 35MB/s 수준으로 감소하였으며, 디스크 회전 지연과 대기열 처리 오버헤드의 영향을 받은 결과로 해석된다. 한편, 블록 크기를 8MB로 확대한 경우, LVM-stripe 구성은 Directed/Cached 모드에서 각각 128MB/s, 149MB/s의 상대적으로 높은 성능을 보였다.

Case B(Writeback 페이지 캐시 활성화 환경)에서는 모든 측정 항목에서 성능이 크게 향상되었다. 순차 읽기의 경우 LVM-linear 는 17.7GB/s로, 캐시 미사용 상태 대비 약 82 배 향상되었고, LVM-stripe 와 RAID 구성 역시 각각 약 870%, 460% 수준의 증가율을 보였다. 무작위 읽기에서도 각 구성은 Case A 대비 20 배 이상 처리량이 증가하였으며, 이는 캐시의 버퍼링 효과가 무작위 접근에서도 일정 수준의 성능 개선을 유도함을 시사한다. 반면, 무작위 읽기/쓰기의 경우 성능 하락이 다시 관측되었다. LVM-linear 는 786MB/s로, 직전 무작위 읽기 대비 약 93% 감소하였다. LVM-stripe 와 RAID 역시 각각 545MB/s, 694MB/s로 동일하게 급격한 감소세를 보였다. 이는 캐시 사용 시 커널의 디스크 플러시 타이밍에 따른 처리 지연과 메모리 자원 경쟁이 복합적으로 영향을 미친 결과로 분석된다.



(그림 3) Case B: VM 부하 실험 (페이지 캐시 활성화)

4. 분석 및 고찰

실험 결과, 디스크 구성 방식과 캐시 설정에 따라 I/O 처리량은 현저한 차이를 보였다. LVM-linear 구성은 데이터가 순차적으로 저장되어 캐시 적중률이 높았고, 이에 따라 캐시 사용 시 가장 큰 성능 향상을 나타냈다. 반면 LVM-stripe 와 RAW-RAID 는 데이터를 디스크에 분산 저장하므로, 캐시 미스율이 높아 처리량 증가폭이 제한적이었다. 또한 페이지 캐시 사용은 메모리 사용 증가를 야기하므로, 커널의 플러시 작업으로 인한 VM I/O 지연이 발생할 수 있다. 따라서, 캐시 전략은 워크로드 특성에 따라 조정되어야 한다.

5. 결론

본 연구는 KVM 기반 가상 머신 환경에서 디스크 구성 방식과 페이지 캐시 사용 여부가 I/O 성능에 미치는 영향을 정량적으로 분석하였다. 그 결과, 캐시 미사용 환경에서는 RAID 구성이 전반적으로 우수한 성능을 보였으며, 캐시 활성화 시에는 LVM-linear 구성이 가장 큰 성능 향상을 나타냈다. 반면, stripe 기반 구성은 캐시 활용 효율성과 시스템 자원 사용 측면에서 한계가 드러났다. 본 결과는 클라우드 환경에서 디스크 구성과 캐시 정책을 수립할 때 실질적인 참고 기준으로 활용될 수 있을 것으로 기대된다.

Acknowledgement

본 연구는 과학기술정보통신부 및 정보통신 기획 평가원의 학석사연계 ICT 핵심인재양성사업의 연구결과로 수행되었음(IITP-2025-RS-2023-00259867)

참고문헌

- [1] Diskeeper Corporation, “Virtualization and Disk Performance”, https://www.nist.gov/system/files/documents/2017/05/09/181_virtualization_disk_performance.pdf, 2017.
- [2] Intel, “Intel(R) Virtual RAID on CPU”, <https://www.intel.com/content/www/us/en/software/virtual-raid-on-cpu-vroc.html>, 2022.
- [3] Ubuntu, “About Logical Volume Management”, <https://documentation.ubuntu.com/server/explanation/storage/about-lvm/index.html>, 2025.
- [4] axboe, “fio”, <https://github.com/axboe/fio>, 2025.