초저지연 저장장치를 위한 적응형 폴링 선택 기법

천명준○, 한상욱, 김지홍
서울대학교 컴퓨터공학부
{mjchun, shanehahn, jihong}@davinci.snu.ac.kr

An Adaptive Polling Selection Technique for Ultra-Low Latency Storage Systems

Myoungjun Chun○, Sangwook Shane Hahn, Jihong Kim
Department of Computer Science and Engineering, Seoul National University

요 약

최근 저장장치 분야의 비약적인 기술 발전으로 기존 저장장치들보다 매우 빠른 응답시간을 제공하는 초저지연 저장장치(Z-SSD, Optane SSD)들이 등장하였다. 이러한 10us 내외의 읽기 응답시간을 가진 저장장치에서는 I/O 과정 중 인터럽트를 처리할 때 발생하는 문맥 교환 비용이 새로운 병목이 될 수 있어, 인터럽트가 아닌 폴링 혹은 하이브리드 폴링 방식을 사용하는 것이 유리하다고 알려져 있다. 본 논문에서는 전체 시스템 CPU 이용률이 높은 상황에서 폴링과 하이브리드 폴링의 잦은 CPU 점유가 읽기 요청에 대한 꼬리 응답시간을 지연시키는 문제가 있음을 밝히고, 전체 시스템 CPU 이용률을 관찰하여 가장 효율적인 처리 방법을 선택함으로써 이를 개선한 적응형 폴링 선택 기법을 소개한다.

1. 서 론

전통적으로 운영체제는 I/O 요청을 처리하기 위해 비동기적으로 동작하는 인터럽트 방식을 사용해 왔다. 인터럽트 방식에서는 사용자 프로세스(User Process)가 디바이스 드라이버(Device driver)로 I/O 요청을 보낸 뒤 Sleep 상태로 인터럽트 핸들러(Interrupt Handler)의 신호를 기다린다. 저장장치에서 I/O 처리가 끝나면 인터럽트 핸들러가 사용자 프로세스로 신호를 보내게 되고 I/O 요청이 끝나게 된다. 인터럽트 방식은 저장장치가 I/O를 처리하는 동안 다른 프로세스가 CPU를 점유할 수 있지만, I/O가 끝날 때 문맥 교환(Context Switch) 비용이 든다.

한편, 최근 저장장치 시장에 소개된 Z-NAND 기반 Samsung Z-SSD[1]나 3D XPoint 기반 Intel Optane SSD[2]들은 각각 12us, 10us의 획기적으로 빠른 읽기 응답시간을 제공하며, 차세대 저장장치의 응답 시간은 이보다 더욱 감소할 것으로 기대된다.

이러한 빠른 응답시간을 제공하는 저장장치에서는 인터럽트 방식의 문맥 교환 비용이 새로운 병목 지점이 될 수 있어, 동기적으로 I/O가 완료되었는지 저장장치의 상태를 주기적으로 검사하는 폴링(Polling) 방식을 사용하는 연구가 기 제안되었다[3]. 하지만 폴링 방식 역시 상태를 검사하는 과정에서 프로세스가 CPU를 계속 점유하여 CPU 이용률이 높아지는 문제가 있다.

하이브리드 폴링(Hybrid Polling)은 프로세스가 타이머를 사용해 일정 시간을 Sleep 한 뒤 깨어나 폴링함으로써 폴링의 빠른 응답속도와 인터럽트의 낮은 CPU 이용률을 모두 취하기 위해 제안된 방식[4]이며, 최신 버전의 리눅스 커널에는 이미 구현되어 있다.

본 연구에서는 먼저 폴링과 하이브리드 폴링 방식을 실제 Z-SSD에 적용한 뒤 여러 시스템 CPU 이용률 상황에서의 읽기 응답시간을 평가하였다. 그 결과, 폴링과 하이브리드 폴링의 꼬리 응답시간이 예상보다 크게 지연되는 문제가 각기 다른 시스템 CPU 이용률 상황에서 관찰되었다. 폴링 방식은 응답시간이 가장 빠를 것으로 기대되었지만, 시스템의 CPU 이용률이 높은 상황에서 매우 긴 꼬리 응답시간을 보였다. 하이브리드 폴링 방식은 폴링 방식과 비슷한 응답시간을 가지면서도 CPU 이용률이 낮을 것으로 기대되었지만, 중간 정도의 시스템 CPU 이용률에서 인터럽트에 근접한 느린 꼬리 응답시간을 보였다.

본 논문에서는 먼저 꼬리 응답시간이 크게 지연되는 문제가 시스템 CPU 이용률에 따른 프로세스 스케줄링과 밀접한 연관이 있음을 밝히고, 폴링과 하이브리드 폴링 방식이 상호 보완적일 수 있다는 고찰을 통해, 현재 시스템 CPU 이용률을 관찰하여 현재 상황에서 가장 알맞은 I/O 완료 통지방식을 선택하는 적응형 폴링 선택 기법인 APS(Adaptive Polling Selection)를 제안한다.

논문의 구성은 다음과 같다. 먼저 2장에서는 리눅스 커널에 구현된 폴링, 하이브리드 폴링 방식을 분석한다. 3장에서는 여러 시스템 CPU 이용률 상황에서 인터럽트, 폴링, 하이브리드 폴링 I/O 통지방식을 평가한 결과와 한계점을 보인다. 4장에서는 제안하는 적응형 폴링 선택 기법의 동작 원리에 대해 설명하고 성능 개선에 대해 평가한 결과를 보인 후, 5장에서 결론을 맺는다.

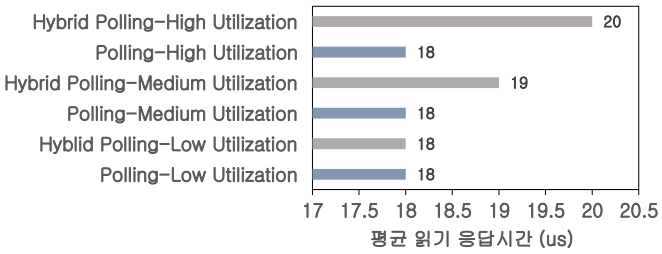
2. 폴링, 하이브리드 폴링

리눅스 커널에서 폴링과 하이브리드 폴링은 블록 계층에 구현되어 있으며 블록 I/O이고 `IOCB_HIPRI` 플래그가 설정된 Direct I/O에 한해 시도된다. 표 1은 본 장에서 살펴볼 주요

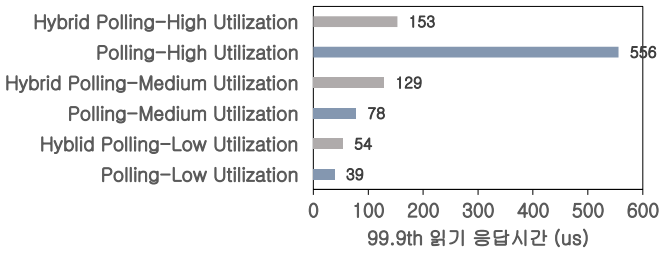
표 1. 폴링 구현 구성 함수

함수 이름	역할
blk_mq_poll	스핀하며 저장장치의 I/O 완료 여부 확인
blk_mq_poll_hybrid_sleep	지정된 시간만큼 프로세스를 Sleep 상태로 변경
blk_mq_poll_nsec	프로세스가 Sleep 할 시간 결정

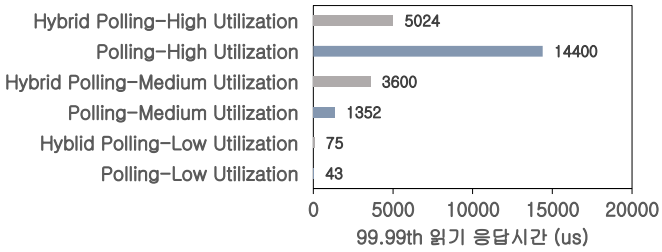
구성 함수의 이름과 역할을 보여준다. 먼저 blk_mq_poll 함수는 폴링이 수행될 때 가장 먼저 호출되며 NVMe 드라이버 계층의 nvme_poll 함수를 통해 저장장치의 I/O 처리가 끝났는지 반복하며 확인한다. 이 과정에서 만약 프로세스가 주어진 타임 퀀텀(Time Quantum)을 모두 사용할 경우 선점될 수 있다. blk_mq_poll_hybrid_sleep 함수는 blk_mq_poll 함수의 시작 부분에서 하이브리드 폴링을 사용할 경우에 호출되며, Hrtimer를 사용하여 프로세스를 지정한 시간동안 Sleep 상태로 바꾸는 역할을 한다. Sleep 상태가 된 프로세스는 지정한 시간이 끝나면 문맥 교환을 통해 blk_mq_poll 함수로 돌아가 폴링을 시작한다. blk_mq_poll_nsec 함수는 하이브리드 폴링을 사용할 경우에만 blk_mq_poll_hybrid_polling 함수의 시작 부분에서 호출되며, 프로세스가 Sleep할 시간을 결정한다. 이는 과거의 I/O 완료 시간을 통해 현재 I/O 요청의 완료 시간을 예측함으로써 이루어진다. 기본적으로 설정되어 있는 Sleep 시간은 최근 16개 I/O 완료 시간의 평균을 2로 나눈 값이다. 하이브리드 폴링의 예측이 잘못되어 오랜 시간을 프로세서가 폴링할 경우도 역시 폴링과 동일하게 선점될 수 있다.



(a) 평균 읽기 응답시간



(b) 99.9 퍼센타일 읽기 응답시간



(c) 99.99 퍼센타일 읽기 응답시간

그림 2. 시스템 CPU 이용률에 따른 읽기 응답시간

3. I/O 완료 통지방식별 응답시간 평가

본 실험의 목표는 폴링, 하이브리드 폴링 방식이 idle 상태가 아닌 여러 시스템 CPU 이용률에서 기대한 응답시간 분포를 보이는지 확인하고, 그렇지 않다면 원인을 분석하여 연구의 동기를 얻는 것이다. 모든 실험은 Intel i7-4790 3.6Ghz 8-스레드 CPU, PC3-10600 16GB 메인 메모리, 리눅스 커널 4.14.36-stable 환경에서 Samsung SZ985 Z-NAND SSD를 사용하여 진행하였다. 응답시간 분포를 측정하기 위한 저장장치 벤치마크로 FIO[5]를 사용하였으며 4개의 쓰레드, 4의 큐 깊이 설정으로 10GB 파일에 대한 4k 임의 읽기 요청 응답시간을 측정하였다. 여러 시스템 CPU 이용률 상황을 만들기 위한 시나리오로는, 그래프 마이닝 응용인 Pegasus[6]로 그래프 마이닝을 수행하여 시스템 전체 CPU 이용률이 다양하게 변화하도록 하였다.

그림 2는 그래프 마이닝 응용의 수를 늘려 가며 측정한 폴링과 하이브리드 폴링 방식의 읽기 응답시간 분포를 보여준다. 그래프 마이닝 응용의 수가 적은 Low Utilization 상황에서는 폴링과 하이브리드 방식이 비슷한 응답시간을 보여주며 99.9, 99.99 퍼센타일의 응답시간 또한 큰 차이가 없다. 하지만 Medium Utilization 상황으로 가면 하이브리드 폴링의 꼬리 응답시간이 폴링에 비해 급격하게 증가하게 된다. 하이브리드 폴링 방식의 함수 별 수행시간을 추가로 수집한 결과, 표 1의 함수 중 blk_mq_poll_hybrid_sleep 함수 수행에서 지연이 생겨 꼬리 응답시간의 대부분을 차지하는 것을 관찰하였다. 함수 내에서 타이머가 만료되면 프로세스가 바로 Sleep 상태에서 깨어나야 하는데, 스케줄링 되지 못해 깨어나는 시간이 늦어지는 것이 지연의 원인으로 보인다.

폴링 방식은 Low, Medium Utilization 상황에선 하이브리드 폴링보다 빠른 응답시간을 보이지만, High Utilization 상황처럼 CPU를 점유하기 힘든 상황에서는 하이브리드 폴링보다 3배 가량 긴 14400us의 꼬리 응답시간을 보인다. 폴링을 사용할 때의 FIO 벤치마크의 CPU 이용률이 Low Utilization 상황에선 96.29%, Medium Utilization 상황에선 92.98%이지만 High Utilization 에서는 68.27%에 불과한 것을 볼 때, 폴링을 하던 프로세스가 선점된 뒤 다시 잡기까지 오랜 지연이 생겨 꼬리 응답시간이 길게 지연되었다고 볼 수 있다. 하이브리드 폴링보다 더 긴 지연이 생기는 이유는 폴링 프로세스가 CPU를 오랜 시간동안 점유하였기 때문에, 선점될 경우 프로세스의 우선순위가 더 낮아져 스케줄링 되기까지 더 오랜 시간이 걸리기 때문으로 보인다.

결론적으로, 시스템의 CPU 이용률이 여유가 있는 상황에서는 폴링 방식이 상대적 우위를 보였으며, CPU 이용률이 높아 폴링을 하는 프로세스가 선점될 수 있는 상황에서는 하이브리드 폴링이 우위를 보였다.

4. 적응형 폴링 선택(Adaptive Polling Selection) 기법

4.1 동작 원리

적응형 폴링 선택 기법은 현재 시스템 CPU 이용률에 따라 폴링과 하이브리드 폴링 방식의 효율성이 변화하는 것에 착안하여, 시스템 CPU 이용률을 예측한 뒤 현재 상황에서 가장 적합하다고 판단된 I/O 통지방식을 사용하는 기법이다.

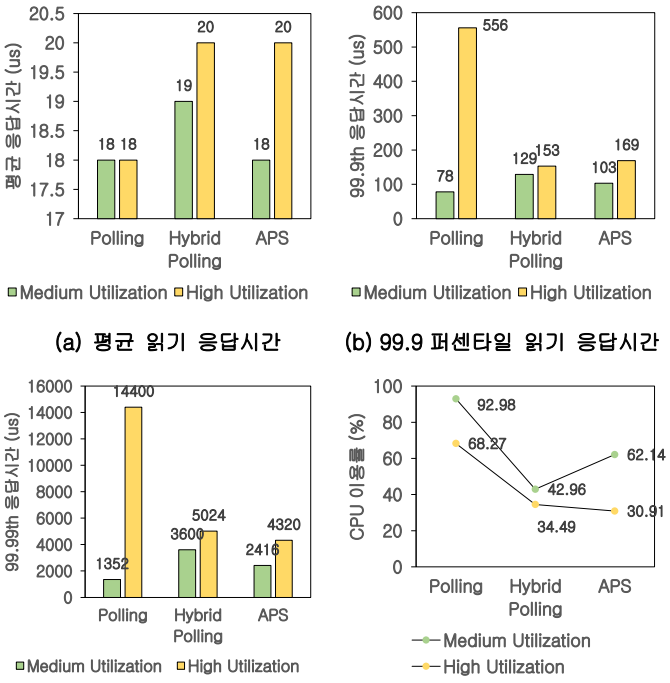
이를 위해, 새로운 I/O 요청이 들어오면 블록 계층에서 폴링과 하이브리드 폴링 중 어떠한 방식이 적합할지 올바르게 예측하여야 한다. 만약 새로운 I/O 요청이 들어온 시점의 시스템 CPU 이용률만을 가지고 판단할 경우 일시적으로 CPU 이용률이 변화하는 시점에는 예측의 정확도가 떨어질 수 있다. 따라서 최근 16개 I/O 요청이 완료된 시점의 CPU 이용률을 추가적으로 수집해 두고, 새로운 I/O 요청이 들어오면 현재 CPU 이용률과 과거 CPU 이용률을 종합하여 현재 사용할 수 있는 CPU 이용률의 마진(Margin)을 계산한다. 계산된 마진 수치가 높다면 폴링을 사용해도 무방하다고 판단하여 폴링을 사용하고, 마진 수치가 중간이라면 적당한 수치(기본 구현에서는 최근 16개의 I/O 완료 시간 평균을 2로 나눈 값)로 하이브리드 폴링을 사용한다. 마진 수치가 낮아 CPU 이용률에 여유가 없다면, 하이브리드 폴링을 사용하되 최대한 많은 시간동안 프로세스를 Sleep시켜 인터럽트에 가깝게 동작하도록 한다.

4.2 기법 평가

본 실험에서는 앞 2장에서 꼬리 응답시간이 크게 지연되었던 Medium Utilization, High Utilization 상황에서 적응형 폴링 선택 기법을 통해 평균, 99.9 퍼센타일, 99.99 퍼센타일의 읽기 응답시간이 단축된 정도를 평가한다. 추가로, 읽기 요청을 보내는 응용인 FIO 벤치마크의 CPU 이용률을 측정하여 기법이 의도한 대로 잘 동작하는지 확인하였다.

그림 3의 (a)는 시스템 CPU 이용률에 따른 각 방식의 평균 읽기 응답시간을, (b)는 99.9 퍼센타일 읽기 응답 시간을, (c)는 99.99 퍼센타일 읽기 응답 시간을 나타낸다. 적응형 폴링 선택(APS) 기법은 시스템 CPU 이용률이 낮아 마진이 클 때는 폴링에 가까운 응답시간을 달성했으며, 시스템 CPU 이용률이 높아 마진이 얼마 없는 상황에서는 하이브리드 폴링에 가깝거나 소폭 개선된 응답시간을 나타내는 것을 확인하였다. 구체적으로, 99.9 퍼센타일의 꼬리 응답시간이 Medium Utilization 상황에서 하이브리드 폴링에 비해 20.15% 감소하였으며 High Utilization 상황에서 폴링에 비해 69.23% 감소하였다. 99.99 퍼센타일의 꼬리 응답시간의 개선은 Medium Utilization 상황에서 32.89%, High Utilization 상황에서 70.00% 감소하여 더 큰 향상폭을 보였다.

그림 3의 (d)는 읽기 요청을 발생시키는 FIO 벤치마크의 CPU 이용률을 보이고 있다. 마진이 높은 Medium Utilization 상황에서 적응형 폴링 선택 기법은 CPU를 더 사용하여도 된다는 판단 하에 동작하므로, 하이브리드 폴링 보다는 높고 폴링보다는 낮은 CPU 이용률을 보이면서도 폴링에 가까운 꼬리 응답시간을 달성하였다. 마진이 얼마 없는 High Utilization 상황에서는 CPU를 하이브리드 폴링보다도 적게 사용하는 인터럽트에 가까운 방식으로 동작하면서도, 하이브리드 폴링보다 99.99 퍼센타일 꼬리 응답시간을 소폭 개선할 수 있음을 확인하였다.



(a) 평균 읽기 응답시간 (b) 99.9 퍼센타일 읽기 응답시간

(c) 99.99 퍼센타일 읽기 응답시간 (d) CPU 이용률

그림 3. 적응형 폴링 선택(APS) 기법 평가

5. 결 론

본 논문에서는 초저지연 저장장치에서 인터럽트 처리의 비용을 줄이기 위해 사용되는 폴링과 하이브리드 폴링 방식이 시스템 CPU 이용률의 변화에 따라 긴 읽기 꼬리 응답시간을 만드는 문제가 있음을 지적하고 이를 개선한 적응형 폴링 선택 기법을 제안하였다. 제안한 기법은 시스템 CPU 이용률을 예측하여 I/O 요청이 들어온 시점에 가장 효율적일 것으로 기대되는 I/O 완료 통지방법을 선택적으로 적용함으로써, 폴링과 하이브리드 폴링의 장점을 모두 취할 수 있음을 실제 저장장치에서의 평가를 통해 확인하였다.

감사의 글

이 연구를 위해 연구장비를 지원하고 공간을 제공한 서울대학교 컴퓨터연구소에 감사드립니다. 이 논문은 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2018R1A2B6006878). (교신저자: 김지홍)

참 고 문 헌

[1] Samsung Z-SSD SZ985. https://www.samsung.com/us/labs/pdfs/-collateral/Samsung_Z-NAND_Technology_Brief_v5.pdf.

[2] Intel Optane SSD 900P. https://www.intel.com/content/www/us/en/-products-memory-storage/solid-state-drives/gaming-enthusiast-ssds/optane-900p-series.html.

[3] J. Yang et al., “When Poll is Better than Interrupt,” in Proceedings of the USENIX Conference on File and Storage Technologies, Vol. 12, 2012.

[4] Damien Le Moal., “I/O Latency Optimization with Polling,” in the Valut-Linux Storage and Filesystems Conference, 2012.

[5] Flexible I/O Tester. https://github.com/axboe/fio.