

SSD 캐시를 이용한 RAID의 에너지 절감 기법

(Energy Conservation of RAID by Exploiting SSD Cache)

이 호 정 [†] 김 은 삼 ^{**}
(Hyo J. Lee) (Eunsam Kim)

노 삼 혁 ^{***}
(Sam H. Noh)

요 약 최근 서버시스템 에너지 절감이 중요하게 대두되고 있다. 서버시스템에서 스토리지는 에너지 사용량이 매우 큰 서브시스템이지만, RAID와 같은 스트라이핑 기법과 하드디스크의 물리적 특성 때문에 실용적인 에너지 절감 기법 개발에 어려움이 따른다. 본 연구팀의 관찰에 따르면 데이터 센터에서 전체 데이터 셋의 크기에 비해 하루 또는 몇 시간과 같이 짧은 기간의 풋프린트는 매우 작다. 본 논문은 이러한 관찰에 기반하여 SSD를 그러한 풋프린트를 감당하기 위한 커다란 캐시로 활용하여 에너지를 절약하는 새로운 아키텍처와 기법을 제안한다. 제안된 기법을 Linux 환경에서 네 개의 하드디스크와 한 개의 SSD로 구성된 RAID5에 구현하고 전력 사용량 측정 장치를 이용해 실제 전력 소모량을 측정한 결과 널리 사용되는 서버 트레이스에서 최대 14%의 저장장치 에너지 절감 효과를 얻었다.

키워드 : SSD, 에너지, RAID

Abstract Energy conservation of server systems has become important. Though storage subsystem is one of the biggest power consumers, development of energy

- 이 논문은 2007년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. ROA-2007-000-20071-0)
- 이 논문은 2009 한국컴퓨터종합학술대회에서 'SSD 캐시를 이용한 RAID의 에너지 절감 기법'의 제목으로 발표된 논문을 확장한 것임

[†] 학생회원 : 홍익대학교 컴퓨터공학과
hjlee@mail.hongik.ac.kr

^{**} 정 회 원 : 홍익대학교 컴퓨터공학과 교수
eskim@hongik.ac.kr

^{***} 종신회원 : 홍익대학교 컴퓨터공학과 교수
samhnoh@hongik.ac.kr

논문접수 : 2009년 8월 13일

심사완료 : 2009년 11월 9일

Copyright©2010 한국정보과학회: 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨터의 실제 및 레터 제16권 제2호(2010.2)

conservation techniques is challenging problem due to striping techniques like RAID and physical characteristics of hard disks. According to our observation, the size of footprint for a day or for hours is much smaller compared to the size of whole data set. In this paper, we describe a design of a novel architecture for RAID that uses an SSD as a large cache to conserve energy by holding such a footprint. We incorporate these approaches into a real implementation of a RAID 5 system that consists of four hard disks and an SSD in a Linux environment. Our preliminary results in actual performance measurements using the cello99 and SPC traces show that energy consumption is reduced by a maximum of 14%.

Key words : SSD, energy, RAID

1. 서 론

전통적으로 배터리로 기반의 모바일 환경에 비해 서버시스템의 전력소모는 부차적인 문제로 취급되어 왔다. 그러나 미국 환경보호국의 보고서에 따르면 2006년 미국의 데이터 센터들이 미국 전체 전력 사용량의 1.5%를 소모했으며 그 요구량이 해마다 12%씩 증가해 오고 있다[1]. 이러한 전기 사용량은 매달 지속적으로 발생하는 운용비용인 막대한 전기 요금으로 직결되며, 이는 사용자에게 설비를 추가로 갖추는 것 보다 심각한 비용 문제를 야기한다. 따라서 최근 서버시스템의 전력 소모를 낮추기 위한 연구가 활발하게 이뤄지고 있다.

스토리지는 서버시스템에서 에너지를 많이 소모하는 대표적인 서브시스템의 하나로 전체 전력의 약 30%를 소모한다. 그러나 스토리지의 전력 소모를 줄이는 기법에는 여러 가지 어려움이 따른다. 첫째로 현재 서버시스템의 스토리지의 대부분을 구성하는 하드디스크의 물리적 특성이 절전 기법의 개발에 장애가 된다. 하드디스크는 물리적 구동부가 있어 스핀 다운을 시키지 않는 한 기본적인 전력 소모가 크다. 또한 트랜지션 성능과 전력의 오버헤드가 크기 때문에 스핀 다운을 시키는 일도 어려움이 따른다. 둘째로 서버시스템과 같은 대용량 시스템의 스토리지에서 주로 사용되는 RAID의 스트라이핑 기법은 병렬처리를 통해 스토리지가 높은 성능을 제공할 수 있도록 해 주지만 동시에 로드가 낮을 때에도 모든 데이터 불륨에 요청을 발생시킨다. 이로 인해 RAID를 구성하는 하드디스크들은 로드가 낮을 때에도 충분한 유휴 시간을 가지지 못해 디스크의 회전이 멈추게 되는 스탠바이 모드로 들어갈 수 없어 절전 기법의 개발에 큰 장애가 된다.

최근 각광을 받고 있는 SSD는 디스크와 동일한 인터페이스를 가지지만 그 내부가 플래시 메모리로 이루어져 있다. SSD는 하드디스크에 비교해 빠른 속도를 제공

하며 전력 소모가 낮고 트랜지션 오버헤드가 거의 없는 매력적인 특성을 가지고 있다. 그러나 이러한 장점에도 불구하고 SSD를 주 저장장치로 사용하기에는 아직까지 용량이 충분하지 않고 용량 당 가격이 하드디스크에 비해 너무 비싸다. 따라서 SSD를 이용하여 의미 있는 절전 효과를 거두기 위해서는 주 저장장치를 SSD로 대체하는 것이 아니라 하드디스크와 함께 사용하면서 그 특성을 잘 살릴 수 있는 기법이 필요하다.

본 논문에서는 이러한 현재의 환경에서 SSD를 RAID의 커다란 캐시로 사용하여 성능 저하를 최소화 하면서 에너지를 절약할 수 있는 구조를 제안한다. 또한 선반입을 수행하여 SSD가 최대한의 요청을 수행하도록 하기 위한 기법에 대해 기본적인 실험 결과를 제공한다. 기존에 널리 쓰이는 미리 읽기 방식의 선반입은 이러한 구조에서 적절하지 않음을 확인할 수 있다. 이 결과와 분석을 통해 장치 특성을 고려할 때 선반입 기법이 일반적인 캐시 계층구조 상에서의 선반입과 어떻게 달라져야 하는지에 대해 논한다.

본 논문에서는 제안한 기법의 검증에 위해 Linux 환경에서 소프트웨어 RAID에 해당 기법을 구현했다. 널리 사용되는 서버시스템 트레이스인 HP Cello99[2]와 SPC Web Search[3] 트레이스를 수행시키고 전력 소모량 측정 장치를 이용해 실제 전력 소모량을 측정한 결과, 최대 14%의 저장장치 에너지 절감 효과를 얻었다.

본 논문은 다음과 같이 구성된다. 먼저 관련 연구와 SSD의 특성을 2절에서 설명하고 3장에서는 본 연구의 바탕을 이루는 관찰에 대해 설명한다. 4장에서는 본 연구에서 제안하는 구조와 기법에 대해서 기술하고 5장에서 실험 결과를 통해 성능을 분석한다. 끝으로 5장에서 결론을 맺고 차후 연구 과제에 대해 기술한다.

2. 배경

본 절에서는 스토리지 전력 소모를 감소하는 기법에 대한 관련 연구들을 살펴보고 본 논문에서 사용하는 하드웨어인 SSD의 특성을 기술한다.

2.1 관련 연구

스토리지의 전력소모를 절감하는 기법은 최근 몇 년간 다양한 관점에서 이뤄지고 있다. 이들은 크게 새로운 유형의 장치를 활용한 기법, 데이터 배치와 중복성을 이용한 기법, 캐시정책을 활용한 기법으로 나누어 볼 수 있다.

첫째, 새로운 유형의 장치를 활용한 기법은 주로 다수의 회전속도를 제공하는 멀티스피드디스크를 활용한 기법들이 많이 연구되었다. Gurumurthi 등과 Carrera 등은 동적으로 회전 속도를 바꿀 수 있는 멀티 스피드 디스크를 제안하고 이를 이용한 에너지를 절감 알고리즘을 함께 제시하였다[4,5]. Zhu 등은 두 가지 속도를 가

지는 디스크를 디스크 어레이에서 활용하여 에너지를 관리하는 기법을 제시하였다[6]. 그러나 멀티스피드 디스크가 기대와 달리 널리 사용되지 않아 이러한 연구들의 성과가 실제로 반영되기 어렵다. 본 논문은 이미 시장에서 널리 사용되고 있는 SSD를 이용한다는 점에서 보다 실용적인 방법이라 할 수 있다.

두 번째로 데이터 배치와 중복성을 활용해 디스크 어레이에서 디스크들을 가능한 오랫동안 스텐바이 상태로 보내고자 하는 기법들이 있다. 대표적인 기법들은 자주 요청되는 데이터들을 특정한 디스크(들)로 옮겨서 다른 디스크들을 가능한 오랫동안 스텐바이 모드에 있도록 한다[6-8].

세 번째로 캐시 정책을 변경하여 에너지 절감 효과를 얻고자 하는 기법이 있다. 하드웨어 디스크 어레이 컨트롤러에 있는 메모리에 중복적인 데이터가 저장된다. 이러한 정보를 에너지 효율성에 활용하기 위한 캐시 관리 기법[9-11]과 선반입 기법[12,13]들이 있다. 그러나 컨트롤러 내의 메모리 크기는 상당히 한정되어 있기 때문에 교체 정책이 매우 중요하고 어려운 문제가 된다. 뿐만 아니라 디스크 어레이의 메모리를 추가하거나 교체하는 것은 불가능하거나 매우 어렵다. 본 논문에서 제안하는 기법은 SSD를 활용함으로써 쉽게 교체 및 추가가 가능하다는 점에서 보다 실용적인 특성을 갖는다.

2.2 Solid State Disk

플래시 메모리는 빠른 성능과 에너지 효율성 때문에 임베디드 시스템, USB메모리 등에 널리 사용되어 왔고, 최근 디스크와 동일한 인터페이스를 가진 SSD형태로 출시되어 차세대 주 저장장치로 기대되고 있다. SSD는 NAND 플래시 메모리들을 어레이로 배치하여 병렬적인 접근이 가능하다. 기계식이 아닌 전자식 장치라는 특성과 병렬성과 인터리빙의 차용 때문에 SSD는 하드디스크보다 매우 빠른 랜덤 리드 성능을 제공하며, 그 밖의 오퍼레이션에 대해 하드디스크와 유사한 성능을 제공한다. SSD의 성능은 지속적으로 향상되고 있으며 당분간은 그 향상이 지속될 것으로 기대된다.

또한 SSD는 기계적인 구동부가 없어 하드디스크에 비해 적은 에너지를 소비한다. 그러나 SSD를 서버시스템의 주된 저장장치로 사용하기에는 아직 무리가 있다. 우선 SSD는 아직 그 용량 면에 있어서 서버 저장장치로 사용되기에 부족하다. SSD를 어레이로 구성하여 서버용 저장장치로 사용하는 방법을 고려할 수 있으나, SSD는 하드디스크에 비해 용량 당 비용이 커서 어레이를 구성하는 데 따르는 비용이 지나치게 커진다.

3. 관찰

본 절은 해당 기법을 본 연구를 진행하는 배경이 된 관찰과 분석들을 서술한다.

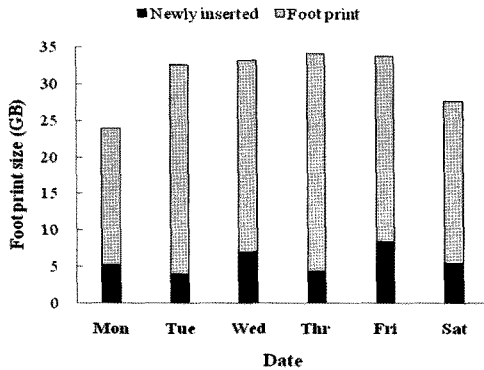


그림 1 HP Cello99의 일별 풋프린트

3.1 풋프린트

서버시스템의 워크로드 특성을 파악하기 위해 대표적인 서버시스템의 I/O 트레이스 중 하나인 HP Cello99를 분석하였다. 그림 1은 11월 29일 월요일부터 12월 4일 토요일까지의 트레이스를 분석한 결과이다. 그래프에서 X축은 날짜를, Y축은 전체 용량 대비 해당하는 날짜에 접근된 데이터 셋인 풋프린트의 크기를 보여준다.

그래프를 통해 실제로 하루 동안 접근되는 데이터의 셋은 전체 사용량의 0.2% 이내인 35GB 이내임을 확인할 수 있다. 또 한 가지 주목할 점은 바에서 검게 칠해진 부분이다. 이 부분은 전날 접근되지 않았지만 당일 접근된 영역을 표시한다. 이를 통해 일별 풋프린트 크기가 전체 용량에 비해 작을 뿐 아니라 그 변화가 비교적 완만함을 알 수 있다.

3.2 지역성

일반적으로 메모리 레벨의 트레이스는 매우 높은 시간적 공간적 지역성을 보인다. 그러나 본 연구에서 관심 대상으로 삼는 메모리를 거친 이후인 장치 레벨의 트레이스를 분석한 결과 다음과 같은 결과를 얻었다.

우선 특히 읽기 요청의 시간적 지역성이 메모리 레벨 트레이스에 비해 매우 낮음을 확인할 수 있었다. 이는 메모리에서 읽기 요청의 시간적 지역성의 상당 부분을 이미 흡수했기 때문에 나타나는 현상으로 생각할 수 있다. 특히 서버시스템의 경우 대용량의 메모리를 사용하기 때문에 일간 데이터를 분석했을 때 날짜에 따라 다르지만 대부분의 날짜에서 한번 요청된 읽기 요청이 재요청 되는 회수가 평균 2~3회에 머물렀다.

공간적 지역성은 상대적으로 그 특성이 상당 부분 유지된다. 완전히 연속하지 않아도 전체 볼륨 중에 좁은 영역이 상당히 자주 요청되는 것을 확인할 수 있었다. 이는 각 서버, 날짜마다 동시에 수행되는 어플리케이션들이 워킹 셋이 좁은 것에 기인한다.

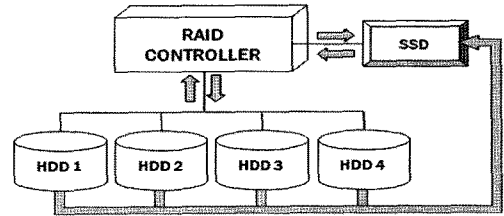


그림 2 SSD 캐시를 탑재한 RAID 컨트롤러

4. SSD 캐시

본 연구는 관찰 내용을 기반으로 SSD를 커다란 캐시로 활용하여 넓은 범위에서 유지되는 풋프린트를 최대한 SSD에 담아 하드디스크를 최대한 스탠바이 상태로 유지시킬 수 있는 시스템을 제안한다. 그림 2는 본 연구에서 제안하는 시스템의 개념적 구조를 보여준다. 일반적인 RAID구성에 SSD가 커다란 캐시로 추가되어 가능한 많은 서비스를 SSD가 수행할 수 있도록 한다.

컨트롤러가 SSD를 활용하는 방법은 다음과 같다. 먼저 읽기 요청이 발생하면 해당 내용이 SSD에 존재하는지 살펴보고 유효한 상태로 존재한다면 SSD에서 대신 서비스한다. 존재하지 않는다면 하드디스크에서 요청을 수행하며 읽은 내용을 SSD에 복사하여 추후 요청을 SSD가 서비스 할 수 있도록 한다. 쓰기 요청이 발생할 경우 해당 내용이 SSD에 이미 존재하는지 살펴보고 해당 내용이 이미 SSD에 존재한다면 이전 기록은 무효화시킨다. 그리고 사용할 수 있는 공간이 존재한다면 SSD에 해당 내용을 적어 하드디스크 요청을 방지한다.

만약 요청될 블록을 미리 SSD에 가져다 놓을 수 있다면 훨씬 큰 에너지 절감 효과와 성능 개선 효과를 얻을 수 있다. 따라서 이러한 기본적인 동작 외에, 읽기 요청이 발생할 때 선반입을 수행할 수 있도록 바꿀 수 있다.

5. 성능 측정

본 절에서는 본 연구에서 제안하는 기법의 성능에 대해 실험환경, 구현 내용과 그 결과를 통해 기술한다.

5.1 실험환경

많은 에너지 절감 기법에 대한 연구들이 시뮬레이터를 기반으로 두고 수행되고 있지만 스토리지의 에너지 절감 관련 연구에서는 시뮬레이터 수행결과와 실제 구현 결과가 일치하지 않는 경향이 있다. 따라서 본 논문에서는 실제 구현을 하고 실험을 수행하여 시스템이 소모하는 전력을 전력계측계를 이용해 측정하는 방식을 택했다.

본 논문에서 제안하는 기법은 소프트웨어 RAID에서도 하드웨어 RAID에서도 구현 가능한 기법이다. 본 논문에서는 Linux 커널에 들어있는 md드라이버를 이용한 소프트웨어 RAID에 본 기법을 구현하고 실험하였다.

RAID구성은 SCSI 인터페이스를 사용하는 것이 보다 일반적이고 더 높은 성능을 기대할 수 있지만, SCSI 기반의 SSD를 시장에서 구매하기 어렵기 때문에 SATA 인터페이스를 사용하여 환경을 구축하였다. 최근에는 SATA 인터페이스의 발전과 저렴한 가격 때문에 SATA 기반의 RAID 역시 널리 사용되는 추세이고 본 논문에서 제안하는 기법은 인터페이스에 구애 받지 않고 동일하게 사용될 수 있는 기법이기 때문에 다른 환경 구성에서도 동일한 결과를 보일 수 있을 것으로 기대한다.

본 논문의 모든 실험에서는 RAID를 구성하는 하드디스크로 Seagate사의 ST3500320AS(500GB) 네 개를 사용하였고, SSD 캐시로는 Mtron사의 MSD-SATA3035(32GB)를 사용하였다. 커널은 Linux 2.6.21.7을 사용하였다.

트레이스로는 앞서 설명한 HP Cello99 중 11월 29일부터 12월 5일까지 6일간과 SPC Web Search 트레이스를 활용하였다. SPC Web Search 트레이스는 SPC 벤치마크를 웹 검색에 맞추어 구성하여 수행한 결과를 수집한 트레이스로 어플리케이션 특성 상 읽기 요청이 매우 많은 특성을 갖는다.

트레이스를 수집한 환경과 실험 환경의 차이로 인해 트레이스를 정확하게 재생하는 데 어려움이 따르기 때문에 각 트레이스의 로드를 초당 100개의 요청을 재생하는 Low와 초당 200개의 요청을 재생하는 High의 두 가지 모드로 실험하였다.

5.2 구현

리눅스 소프트웨어 RAID는 md(multiple device) 드라이버에 구현되어 있으며 mdadm 어플리케이션으로 제어할 수 있다. SSD cache를 RAID구성에 포함시킬 수 있도록 하기 위해 mdadm을 수정했다. 그리고 md 디바이스 드라이버 코드를 수정하여 md 가 다음과 같은 기능을 추가하였다.

우선 md는 SSD에 들어있는 데이터를 관리하기 위해 해시 테이블을 유지한다. 원본 요청에 해당하는 디스크 번호와 디스크 섹터, 그리고 그에 해당하는 SSD 섹터 정보와 더티 여부를 알기 위한 상태정보가 해시 테이블에서 관리된다.

읽기 요청이 도착하면 해시 테이블에 해당 요청이 SSD에 존재하는지 살펴본다. 해당 요청이 SSD에 없으면 하드디스크를 통해 요청을 수행한다. 요청이 완료되면 사용자에게 응답하고 같은 내용을 SSD에 복사하고 SSD에 쓰기 연산이 끝날 때 해시 테이블을 업데이트 한다. 만일 해당 요청이 SSD에 있으면 하드디스크로 가지 않고 SSD에서 바로 서비스한다.

쓰기 요청이 도착하면 SSD의 공간이 충분할 경우 해당 요청은 SSD에 기록되고 해시 테이블의 더티 상태가 업데이트된다.

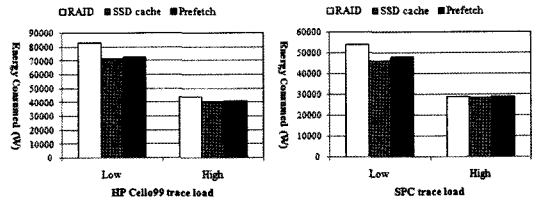


그림 3 로드별 에너지 사용량

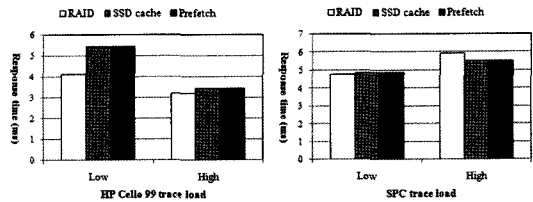


그림 4 로드별 응답 시간

공간이 부족하거나 더티한 공간이 SSD의 70%를 넘어가게 되면 클리너 쓰레드가 구동된다. 클리너 쓰레드는 무효상태의 블록들에 대해서는 단순히 클린 상태로 바꾸고 더티 상태의 블록들을 원래 써져야 할 하드디스크에 기록하고 클린 상태로 변경한다.

본 연구에서 관심을 두는 성능 지표는 에너지 소모량과 응답 시간이다. 에너지 소모를 측정하기 위해 실제 사용된 에너지를 측정할 수 있는 장치인 HPM-300A를 사용하였고, 응답시간을 측정하기 위해 각 요청이 소요한 시간을 gettimeofday함수를 이용해 측정하였다.

5.3 실험 결과

그림 3는 각 트레이스를 로드별로 수행했을 때 스토리지에 의해 소모된 에너지를 보여준다. SSD 캐시는 모든 경우에서 에너지 절감효과를 나타냈고 전체 실험을 통해 평균 9.65%가량의 에너지를 절감할 수 있었다. 그래프를 통해 로드가 낮을 때 에너지 절감 폭이 훨씬 큰 것을 확인할 수 있다. 로드가 높을 때 에너지 사용량이 로드가 낮을 때의 에너지 사용량보다 큰데 이것은 같은 분량의 로드를 더 짧은 시간동안 수행하기 때문에 기본 소모 전력이 줄어들기 때문이다. 또한 로드가 낮을 때 에너지 절감 폭이 로드가 높을 때의 에너지 절감 폭보다 큰 것을 확인할 수 있는데 이는 SSD가 요청을 대행함으로써 얻을 수 있는 유희시간이 더 길어지기 때문이다.

그림 4는 각 트레이스를 로드 별에 따라 수행했을 때 각 요청의 평균 응답 시간을 보여준다. HP Cello99의 경우 SSD 캐시를 사용했을 때, 선반입 여부에 관계없이 응답속도가 느려졌고 SPC의 경우 로드가 낮을 때는 응답속도가 느려지지만 로드가 높을 때는 오히려 응답속도가 빨라졌다. 그 원인은 각 트레이스의 특성과 SSD의 특성 모두에서 찾을 수 있다. SSD는 하드디스크보다

매우 빠른 랜덤 읽기 성능, 유사한 순차 읽기/쓰기 성능을 제공하지만 랜덤 쓰기 연산에 대해 매우 취약하다. 따라서 쓰기 요청이 주를 이루는 HP Cello99를 수행할 때는 성능 저하가 발생하고 읽기 요청이 주를 이루는 SPC를 수행할 때는 성능 이익을 얻을 수 있다. SPC수행에서 로드가 낮아질 때 소폭의 성능저하가 발생하는 이유는 하드디스크들이 유희상태로 전환되었다가 돌아오는 트랜지션 오버헤드에 의한 성능저하이다. 일반적으로 생각할 수 있는 것과 달리 HP Cello99의 경우에는 로드가 높을 때보다 낮을 때 응답시간이 긴데 이것은 쓰기 요청이 집중되면서 I/O 패스 상에서 이익을 얻을 수 있기 때문인 것으로 생각된다.

선반입을 잘 수행할 경우 좋은 효과를 얻을 수 있을 것으로 예상하였으나 실제 결과에서는 응답속도에 있어서도 이익이 거의 없을 뿐 아니라 에너지 절감 면에서는 선반입을 하지 않았을 때와 비교해 오히려 손해가 발생했다. 선반입 및 일반 요청의 시작과 종료시간을 관찰한 결과 다음의 두 가지 문제를 확인할 수 있었다.

첫째로 SSD로 일어나는 선반입 요청처리가 하드디스크에서 메모리로의 선반입처럼 빠르지 않았다. 이로 인해 많은 경우에 선반입 요청이 완료되기 이전에 실제 요청이 도착하여 하드디스크에 요청이 발생했다.

둘째로 선반입을 수행하기 위해 일어나는 추가적인 I/O 오버헤드이다. 모든 선반입 기법은 불필요한 I/O 발생이 오버헤드가 될 위험을 내포하는데, 선반입의 이익이 이미 감소한 상태에서 이러한 오버헤드가 발생함으로써 선반입의 좋은 영향을 넘어서게 된다.

실험 결과를 통해 이러한 구조에서는 미리 읽기같은 방법보다 접근 패턴을 활용해 보다 빨리 선반입을 구동할 수 있도록 하는 방식이 필요함을 유추할 수 있다.

6. 결론 및 향후 과제

본 연구에서는 SSD를 RAID의 캐시로 활용하여 에너지를 절감하는 기법을 제안하였다. 본 연구에서 제안하는 시스템을 활용할 경우 비교적 단순한 기법을 통해 최대 14%의 에너지 절감을 얻을 수 있음을 확인할 수 있다. 특히 읽기 요청이 많은 환경에서는 바쁠 때에 성능 이익을 함께 가져올 가능성을 확인할 수 있다. 본 연구에서 제안한 시스템은 기존 시스템에 쉽게 적용할 수 있을 뿐 아니라 많은 최적화 기법의 여지를 가지고 있다는 점에서 지속적인 연구 가치를 확인할 수 있다.

무엇보다 관찰결과와 실험결과를 통해 성능과 에너지 절감 폭에 있어 보다 큰 이익을 얻기 위해서는 장치 특성을 반영한 새로운 선반입 기법이 필요함을 확인할 수 있다. 따라서 본 연구팀은 이러한 기법의 개발을 향후 과제로 삼고 진행하고 있다.

참고 문헌

- [1] Report to Congress on Server and Data Center Energy Efficiency Public Law 109-431, <http://www.energystar.gov>, 2007.
- [2] HP Labs. Tools and Traces. <http://www.hpl.hp.com/research/ssp/software/>
- [3] Storage Performance Council. SPC-1 Specification. <http://www.storageperformance.org/specs>.
- [4] E. V. Carrera, E. Pinheiro, and R. Bianchini. Conserving Disk Energy in Network Servers. In *Proceedings of the 17th International Conference on Supercomputing*, 2003.
- [5] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke. DRPM: dynamic speed control for power management in server class disks. *SIGARCH Comput. Archit. News*, 31(2):169-181, 2003.
- [6] Q. Zhu, Z. Chen, L. Tan, Y. Zhou, K. Keeton, and J. Wilkes. Hibernator: helping disk arrays sleep through the winter. In *Proceedings of the Twentieth ACM Symposium on Operating Systems Principles (SOSP '05)*, pp.177-190, 2005.
- [7] C. Weddle, M. Oldham, J. Qian, A.-I. A. Wang, P. Reiher, and G. Kuenning. PARAIID: a gear-shifting power-aware RAID. In *Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST '07)*, 2007. 20.
- [8] E. Pinheiro and R. Bianchini. Energy conservation techniques for disk arraybased servers. In *Proceedings of the 18th International Conference on Supercomputing*, pp.68-78, 2004.
- [9] L. N. Bairavasundaram, M. Sivathanu, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau. X-RAY: A Non-Invasive Exclusive Caching Mechanism for RAIDs. *SIGARCH Comput. Archit. News*, 32(2), 2004.
- [10] D. Li and J. Wang. EERAID: energy efficient redundant and inexpensive disk array. In *Proceedings of the 11th ACM SIGOPS European Workshop (EW11)*, 2004.
- [11] Q. Zhu, F. M. David, C. F. Devaraj, Z. Li, Y. Zhou, and P. Cao. Reducing Energy Consumption of Disk Storage Using Power-Aware Cache Management. In *Proceedings of the 10th International Symposium on High Performance Computer Architecture (HPCA '04)*, 2004.
- [12] S. H. Baek and K. H. Park. Prefetching with Adaptive Cache Culling for Striped Disk Arrays. In *Proceedings of the Annual USENIX Technical Conference*, 2008.
- [13] S. W. Son and M. Kandemir. Energy-aware data prefetching for multi-speed disks. In *Proceedings of the 3rd ACM Conference on Computing Frontiers (CF'06)*, 2006.