

SSD로 구성된 RAID 장치의 성능 분석

이상수^o, 신동군
 성균관대학교 정보통신대학
 mapia87@skku.edu, dongkun@skku.edu

Performance Analysis on SSD RAID Device

Sang-Su Lee^o, Dongkun Shin
 College of Information and Communication Engineering,
 Sungkyunkwan University.

요 약

SSD는 고성능 저전력 스토리지 시스템으로 각광받고 있다. 본 연구에서는 SSD를 사용한 RAID 장치의 성능을 분석하는 연구를 수행하였다. 하드웨어 RAID 5 시스템에서 SSD의 개수, 스트라이프 크기, 리퀘스트 크기, NCQ의 값을 변화시키며 임의 읽기/쓰기 동작의 성능을 분석 하였다. 실험 결과를 바탕으로 본 논문에서는 패리티를 기반으로 한 RAID 5 시스템에서 임의 쓰기 성능의 문제점을 보여주고, RAID 5 시스템의 구성에 대한 재고가 필요함을 보인다.

1. 서 론

오늘날 인터넷이 널리 보급됨에 따라, 거의 모든 분야에서 정보의 사용량이 극에 달하고 있다. 이에 따라 빠른 정보 확보 및 손실방지를 위한 정보의 저장과 관리의 중요성이 증대되었다. 이러한 이유 때문에 HDD를 대체하기 위한 SSD의 사용량이 증가하고 있다. 특히, 워크스테이션을 위한 고성능 고신뢰성 스토리지 시스템을 구현하기 위해 다수의 SSD를 논리적으로 묶어 하나의 디스크처럼 사용하게 해주는 RAID 기술이 널리 사용될 것으로 예상된다.

RAID를 구현하는 방법은 디스크에 데이터를 나누는 방법에 따라 구분된다. 대표적으로 레벨 0, 레벨 1, 레벨 5가 있으며, RAID 0은 스트라이핑 구성으로서 처리해야 하는 데이터를 다수의 디스크에 분산시켜 저장하고, RAID 1은 미러링 구성으로 다수의 디스크에 동일한 데이터를 중복으로 저장한다. 따라서 RAID 0은 성능의 이득을, RAID 1은 신뢰성의 이득을 볼 수 있다. RAID 5는 레벨 0과 마찬가지로 데이터를 여러 디스크에 분산시켜 저장하는 시스템이다. 하지만 동시에 신뢰성을 확보하기 위해 패리티 블록을 두어 데이터의 손실에 대비한다. 하지만 RAID 5의 쓰기 연산은 패리티 블록을 갱신해야 하는 과정의 오버헤드 때문에 성능이 RAID 0보다 떨어진

다. 또한, RAID를 구성한 SSD의 여분의 공간에 따라 RAID 0에 비하여 RAID 5의 성능이 크게 떨어지는 것을 확인한 연구[4]가 있었으나 스트라이프 크기나 SSD의 개수 등 RAID의 설정과 패리티 데이터 구성에 따른 성능 변화는 알 수 없었다.

본 연구는 기존에 이루어진 연구에서 나아가 SSD로 구성된 RAID 5 시스템을 집중적으로 분석하였다. SSD로 구성된 RAID 5 시스템에서 성능이 떨어지는 포인트를 찾기 위한 실험을 하였다. 실험을 통해 기존의 SSD의 특성과 RAID 장치 설정에 따른 성능 변화를 분석하고 이러한 환경에서 보다 좋은 성능을 얻을 수 있는 구성을 탐색하였다. 성능에 영향을 미치는 요인을 분석하기 위해서 SSD의 개수, 스트라이프 크기, 리퀘스트 크기, NCQ의 값을 변화시키며 성능의 변화를 시뮬레이션 한다. 이를 통해 RAID 5 시스템에서의 성능이 위와 같은 요인들에 의하여 어떻게 나타나는지를 분석한다. 또한 SSD를 사용하여 RAID 5를 구성하였을 때 패리티 블록 관리 비용이 전체 시스템의 성능에 어떠한 영향을 미치는지 알아보았다.

사용자는 위와 같은 각각의 RAID 시스템의 구현 방법과 구성 방법에 따른 장점과 단점 및 성능을 고려하여 원하는 환경에 맞는 RAID 시스템을 선정해야 한다. 이에 따라 SSD로 구성된 RAID와 HDD로 구성된 RAID의 RAID 레벨간 성능 평가에 대한 연구[1][2]와 SSD의 개수와 스트라이프 크기에 대한 RAID 0의 성능 분석이 이루어졌다[3]. 이러한 연구는 RAID 레벨간의 성능 비교, SSD RAID0의 확장성, 전체 성능에 병목 현상을 발생시키는 요인 등에 대한 결과를 보여줬으나 패리티 데이터를 사용하는 RAID 레벨에서의 특성을 관찰할 수 없

2. 실험 환경

실험은 Intel(R) Xeon(R) CPU E5620 2.40GHz 프로세서, 3GB RAM을 장착한 서버를 사용하였으며, Windows7 Ultimate K 32bit 운영체제에서 이루어 졌다. RAID 5 시스템을 구성하기 위한 RAID 컨트롤러로는 Adaptec RAID 6805 6Gb/s SAS&SATA(512MB of DDR2 667MHz cache memory 5.2.0 version)을 사용하였다. RAID 컨트롤러의 쓰기 캐시와 읽기 캐시는 모두 Enabled로 설정하여 RAID 5 시스템을 구성하였으며, RAID 5 시스템을 구성하는데 사용한 SSD는 Samsung SSD 830 Series 64GB이다. 마지막으로 벤치마크 툴은 IOMeter 2006.07.27 version을 사용하여 시뮬레이션 하였다.

3. 실험 변수

RAID 5 시스템에서 성능에 영향을 미치는 요인을 분석하기 위하여 몇 가지의 설정 값을 변수로 두었다. 본 실험에서 변수로 설정한 것은 SSD의 개수, 디스크의 스트라이프 크기, IOMeter를 사용한 시뮬레이션 상의 리퀘스트 크기와 NCQ로 4가지이다. 위의 4가지 변수의 값을 변화시키면서 RAID 5 시스템의 성능 변화를 관찰하였다. 각 변수 값은 [표 1]과 같이 적용하였다.

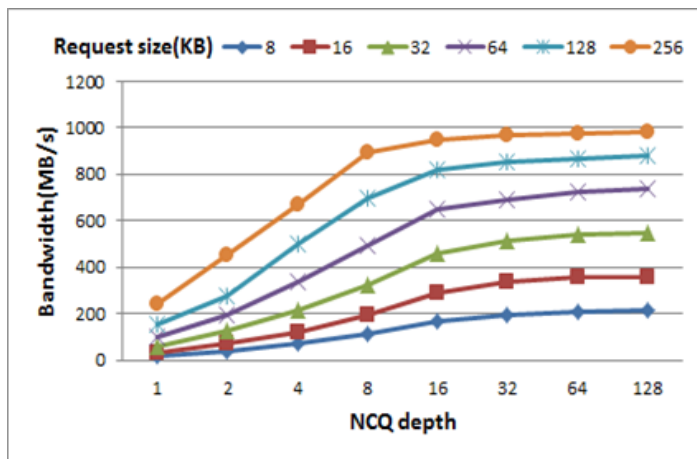
변수	변수 값
SSD의 개수	3~6개
스트라이프 크기	16KB~1024KB
리퀘스트 크기	8KB~256KB
NCQ	1~128

(표 1) 실험에서 변화시킨 변수와 그 변수의 값

4. 실험 결과

4.1. NCQ 및 리퀘스트 크기에 따른 읽기 성능 변화

SSD의 개수와 스트라이프 크기는 각각 3개, 128KB로 고정하고, NCQ와 리퀘스트 크기를 각각 1~128, 8KB~256KB까지 변화시키면서 임의의 읽기 성능을 측정하였다. 그림 1에서 보여주는 바와 같이 NCQ와 리퀘스트 크기가 증가함에 따라 점진적으로 성능이 증가하다가 완만해지는 병목현상을 보인다. 이는 NCQ가 늘어남에 따라 RAID 컨트롤러의 처리량이 한계점에 도달한 것으로 생각할 수 있다.

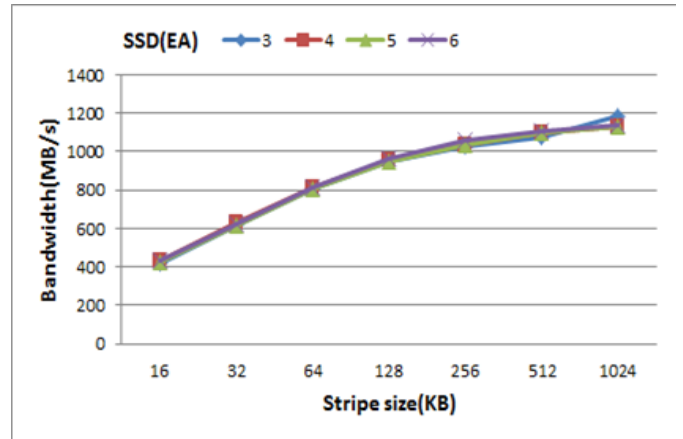


(그림 1) NCQ 및 리퀘스트 크기에 따른 임의의 읽기 성능의 변화.

4.2. 스트라이프 크기 및 SSD의 개수에 따른 읽기 성능 변화

리퀘스트 크기와 NCQ는 각각 128KB, 16으로 고정하고, 스트라이프 크기와 SSD의 개수를 각각 16KB~1024KB,

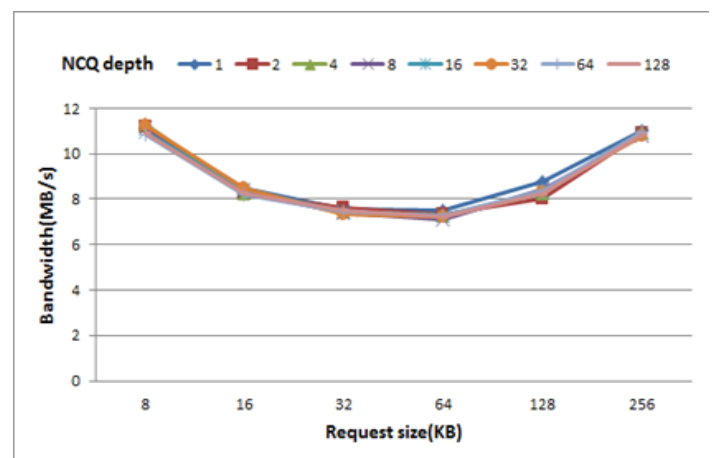
3~6개까지 변화시키면서 임의의 읽기 성능을 측정하였다. 그림 2에서 보여주는 바와 같이 성능이 스트라이프 크기의 변화에 대해서는 증가하는 모습을 보이나, SSD의 개수에 대해서는 변화가 없다. 이는 임의의 읽기의 경우 임의의 디스크에서 데이터를 읽어오게 되는데 NCQ의 영향 때문에 디스크의 개수가 변하여도 성능에 변화가 없는 것으로 생각할 수 있다.



(그림 2) 스트라이프 크기 및 SSD의 개수에 따른 임의의 읽기 성능의 변화.

4.3. NCQ 및 리퀘스트 크기에 따른 쓰기 성능 변화

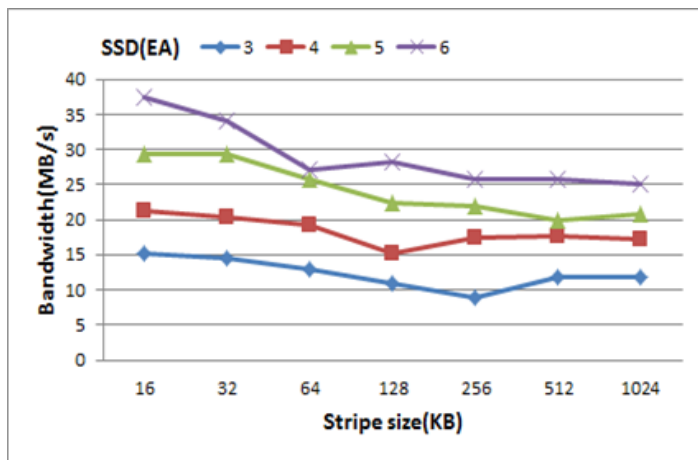
4.1에서 임의의 읽기 성능을 측정한 것과 동일한 방법으로 임의의 쓰기 성능의 변화에 대해서도 실험하였다. 그림 3에서 보여주는 바와 같이 RAID 5 시스템에서의 성능은 리퀘스트 크기가 증가함에 따라 감소하다가 리퀘스트 크기와 스트라이프 크기가 같아지는 지점에서 다시 상승하는 모습을 보이고 있다. 하지만 NCQ의 변화에 대해서는 성능의 변화를 보이지 않는다. 이는 임의의 쓰기 동작의 경우 패리티 블록을 갱신하는 과정에 의한 영향으로 생각할 수 있다.



(그림 3) NCQ 및 리퀘스트 크기에 따른 임의의 쓰기 성능의 변화.

4.4. SSD의 개수 및 스트라이프 크기에 따른 쓰기 성능 변화

4.2에서 임의 읽기 성능을 측정한 것과 동일한 방법으로 임의 쓰기 성능의 변화에 대해서도 실험하였다. SSD의 개수와 스트라이프 크기의 변화에 따른 임의 쓰기 성능의 변화는 그림 4에서 보여주는 바와 같이 SSD의 개수에 따라서는 성능이 증가하는 모습을 보이고 있다. 하지만 스트라이프 크기의 변화에 대해서는 성능이 점차적으로 조금씩 감소하다가 일정해지거나 혹은 약간의 성능이 다시 상승하는 모습을 보인다. 이 결과 또한 패리티 블록의 갱신 과정에 대한 영향으로 성능이 스트라이프 크기가 증가함에 따라 패리티 블록의 사이즈 또한 커짐으로 성능이 감소하는 것으로 볼 수 있다.



(그림 4) 스트라이프 크기 및 SSD의 개수에 따른 임의 쓰기 성능의 변화.

5. 결과 분석

실험을 통해서 RAID 컨트롤러를 사용한 RAID 5 시스템 상에서 SSD의 개수, 스트라이프 크기, 리퀘스트 크기, NCQ의 값의 변화에 따른 성능의 변화를 확인할 수 있다. 이 결과를 통해 RAID 0와 RAID 5의 성능 비교에 대한 연구[4]의 결과보다 자세한 RAID 5의 성능의 변화를 확인할 수 있다. 위의 그림 1, 그림 2를 살펴보면 임의 읽기의 경우 SSD의 개수를 제외한 모든 변수에서 성능 상승의 효과를 볼 수 있었다. 반면에 그림 3, 그림 4를 보면 임의 쓰기의 경우에는 임의 읽기에 비해 성능이 매우 낮고, 임의 읽기의 성능 변화와는 다른 양상을 보인다. SSD의 개수에 의해서는 성능이 약간 증가하지만 스트라이프 크기, 리퀘스트 크기의 변화에 있어서는 성능이 감소하거나 혹은 감소하다가 약간 다시 증가하는 모습을 보이고 있다. 또한, NCQ에 대해서는 임의 쓰기의 경우 성능에 아무런 영향을 미치지 못하고 있다. 이는 RAID 5 시스템에서는 패리티 블록의 존재로 인하여 끊임없이 패리티를 갱신하는 과정에서의 오버헤드 동작으로 생각할 수 있으며, 이로 인하여 임의 쓰기의 성능이 떨어짐을 확인할 수 있다. 특히 리퀘스트 크기가 스트라이프 크기보다 작은 상태에서 증가할 때 성능이 지속적

으로 감소하는 경향을 나타낼 수 있다. 따라서 RAID 5 시스템에서는 위의 결과와 같이 임의 읽기에 있어서는 각 변수들의 값을 높게 설정하여 줄으로써 높은 성능을 기대 할 수 있지만, 임의 쓰기에 있어서는 위의 실험 결과를 바탕으로 사용 환경에 맞는 적합한 변수 값의 선택이 필요하다.

6. 결론

본 연구에서는 SSD로 구성된 하드웨어 RAID 5 시스템에서 임의 읽기 동작의 경우 여러 요인들에 의한 성능 상승을 확인하였다. 하지만 패리티의 영향으로 임의 쓰기 동작에서 스트라이프 크기, NCQ Depth, 리퀘스트 크기의 변화에 따라 성능이 상승하지 못하는 것을 실험을 통하여 확인 하였다. 또한 리퀘스트 크기가 Stripes size의 크기의 약 50%가 되는 지점에서 가장 임의 쓰기 성능이 낮게 나타남을 확인하였다. 이 실험 결과는 하드웨어 RAID 5 시스템을 구성 할 때 참고가 될 수 있을 것이다.

향후 연구 과제로 하드웨어 RAID 5 시스템에서의 성능 향상이 고르지 않은 임의 쓰기에 대한 성능 평가 실험을 소프트웨어 RAID 5 시스템에서 동일한 방법으로 수행하여 분석할 계획이다. 이를 통하여 리퀘스트 크기와 스트라이프 크기의 변화에 따른 성능 변화의 불규칙성의 원인을 규명하고 SSD를 사용한 RAID 5 시스템의 최적화된 구성 방법 및 개선 방안을 찾아내고자 한다.

7. 참고 문헌

- [1] 문성업, 김상우, 이상원, Flash SSD상의 RAID 구성의 고찰, 한국인터넷정보학회, 2008 정기총회 및 추계학술발표대회, 제9권, 제2호, pp. 237~242
- [2] 구분근, 곽윤식, 정승국, 고대식, 구현 방법에 따른 RAID 시스템 성능 분석, 한국정보기술학회, Green IT 융합기술 워크숍 및 하계 종합 학술 대회 논문집, 2009.
- [3] Ilia Petrov, Guillermo G. Almeida, Alejandro P. Buchmann, Ulrich Gräf, Building large storage based on flash disks, Proceedings of ADMS 2010, September, 2010.
- [4] Nikolaus Jeremic, Gero Muhl, Anselm Busse, Jan Richling, The Pitfalls of Deploying Solid-state Drive RAIDs, Proceedings of the 4th Annual International Conference on Systems and Storage, 2011.