

# 소프트웨어 RAID와 LVM 기반의 네트워크 저장장치의 성능 평가

이태근<sup>0</sup>, 강용혁\*, 손재기\*\*, 이형수\*\*, 민수영\*\*, 박창원\*\*, 엄영익\*  
\*성균관대학교 전기전자 및 컴퓨터공학부  
\*\*전자부품연구원 정보 시스템 연구센터  
email : {cyberlee, yhkang1}@ece.skku.ac.kr  
{jgson, hslee, minsy, parkcw}@keti.re.kr  
yieom@ece.skku.ac.kr

## Performance Evaluation of S/W RAID and LVM-based Network Attached Storage

Taekeun Lee<sup>0</sup>, Yong-Hyeog Kang\*  
Jae Gi Son\*\*, Hyung Su Lee\*\*, Soo Young Min\*\*, Chang Won Park\*\*  
Young Ik Eom\*  
\*School of Electrical and Computer Engineering, SungKyunKwan University  
\*\*IT System Research Center, Korea Electronics Technology Institute

### 요 약

대용량, 고속의 멀티미디어 서비스가 일반화됨에 따라 저장장치 또한 대용량화되고 있다. 그러나 여전히 저장장치의 낮은 입출력 속도와 오류로 인해 시스템 전반의 성능에 영향을 미치고 있으며 대용량화로 인한 관리의 부담은 더욱 증가하였다. 본 논문은 소프트웨어 RAID와 LVM 기반의 네트워크 저장장치의 구현에 관한 것이다. 소프트웨어 RAID를 사용하여 저장장치의 성능과 안정성을 높이고 LVM을 사용하여 저장장치의 관리를 쉽게 할 수 있다. 리눅스 커널 버전 2.4에서 제공하는 이러한 기능들을 사용함으로써 적은 비용으로 안정적이며 고성능의 저장장치를 구현할 수 있으며 관리까지 편리한 네트워크 저장장치를 구축할 수 있다.

### 1. 서론

최근의 정보량의 추세는 폭발적이라고 할만큼 급격히 늘어나고 있다. 기존 텍스트 데이터와 같은 기본적인 형태의 데이터는 물론이고 오디오, 비디오, 이미지 같은 대용량의 멀티미디어 데이터가 주류를 이루고 있다. 또한 네트워크의 발달로 인하여 대용량 멀티미디어 서비스가 확산되고 있다. 이러한 추세에 맞추어 우리가 손쉽게 사용하는 PC에도 이미 수십 기가바이트대의 하드디스크가 장착되고 있는 것처럼 저장 장치의 용량 또한 급격히 증가하는 추세이다. 저장 장치가 대용량화되어 많은 양의 데이터를 저장할 수 있지만 저장 장치의 오류로 인해 한순간에 잃을 수 있는 데이터의 양도 그만큼 더 커졌으므로 안정성에 대한 요구가 커지고 있으며 대용량화 되고 있는 저장장치를 보다 쉽게 관리하기 위한 요구 또한 커지고 있다.

따라서 많은 시스템의 사용자들은 보다 안정적이며 빠른 저장 장치를 필요로 하게 되었는데 이것에 대응하는 기술이 RAID(Redundant Array of Inexpensive/Independent Disks)이다. 또한 기존 시스템의 환경을 그대로 유지하면서 급변하는 저장 장치 용량에 대응하는 기술이 LVM(Logical Volume Manager)이다.

최근에는 일반적인 서버를 사용하는 대신 저장 장치 기능을 전담하는 독립적인 네트워크 저장 장치(NAS:Network Attached Storage)를 사용하는 추세이다. 이러한 네트워크 저장 장치는 네트워크 상의 이 기종 클라이언트에게 효율적으로 파일 서비스를 제공할 수 있다. 본 논문에서는 소프트웨어 RAID와 LVM을 적용하여 안정적이며 관리하기 쉬운 NAS을 구축하였다[1].

본 논문의 구성은 다음과 같다. 2장에서는 시스템 구축과 관련된 연구에 대해 기술하였으며, 3장에서는 시스템의 설계 및 구축에 대해 기술한다. 4장에서는 시스템에 대한 성능을 평가하며 마지막으로 결론에 대해 기술한다.

### 2. 관련연구

#### 2.1 NAS

NAS란 파일서버 기능이 내장된 네트워크 저장장치이다. 일반적인 저장장치가 붙어 있는 서버의 경우 응용 서비스와 파일 서비스를 동시에 수행하므로 입출력, CPU, 네트워크에 병목현상을 발생시키는 문제점을 갖는다. 뿐만 아니라 확장에 한계를 가지며 서버가 다운될 경우 모든 서비스가 중단되는 단점이 있다. 따라서 NAS는 응용과 데이터를 분리하여 독립적으로 파일 서버 기능만을 처리하도록 만든 네트워크 저장장치이다.

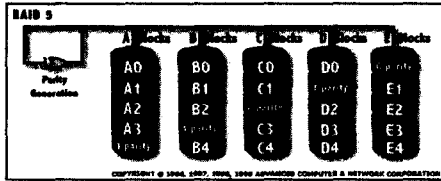
NAS는 기존의 LAN 프로토콜을 사용하여 독립적으로 네트워크에 참여하며 응용서버와는 별도로 존재한다. 따라서 NAS는 이 기종간의 파일 공유를 지원해야 하며 네트워크 및 디스크 성능, 안정성, 관리의 용이성 등이 최적화 되어야 한다.

#### 2.2 RAID

RAID란 여러 개의 디스크를 사용하여 데이터를 저장하는 방법으로 성능과 안정성을 향상시킬 수 있다. RAID는 여러 가지 레벨로 구현되며 레벨 0, 1, 5가 일반적으로 사용된다.

본 논문에서는 사용할 RAID 레벨 5의 개념도는 [그림 1]과 같다. RAID 레벨 5는 가장 유용한 RAID 레벨로써 3개 이상의

디스크로 구성된다. 데이터의 복원 정보에는 패리티 방식을 사용한다. 데이터는 블록 단위로 기록되며, 패리티도 데이터용 디스크에 분산되어 기록된다. RAID 레벨 4에서는 패리티를 하나의 디스크에 저장하는 방식을 사용하는데 이것은 패리티 정보가 하나의 디스크에 집중되어 병목현상을 일으키는 문제점을 갖는다. RAID 레벨 5는 이러한 패리티 정보를 분산하여 RAID 레벨 4가 갖는 병목현상을 제거할 수 있다. RAID 레벨 5는 하나의 디스크의 오류에 대해서만 안전하며 2개 이상의 디스크가 동시에 오류를 일으킬 경우 모든 데이터를 잃을 수 있다.



[그림 1] RAID 레벨 5 개념도

2.2.1 소프트웨어 RAID

RAID를 구성하는 방법에는 하드웨어 RAID와 소프트웨어 RAID가 있다.

고속의 전용 프로세서를 사용하는 하드웨어 RAID는 빠른 속도를 내긴 하지만, 보드 상에 충분한 DSP와 메모리를 가지고 있기 때문에 비용이 많이 든다. 오래된 하드웨어 RAID는 DSP와 캐쉬의 병목현상으로 빠른 CPU를 사용하는 시스템의 속도를 저하시킬 수 있다. 때로는, 소프트웨어 RAID를 사용하는 것보다 더 느릴 수도 있다. 하드웨어 RAID가 소프트웨어 RAID에 비해 장점이 있을 수도 있지만, 최근 대부분의 하드디스크는 그렇지 않다. 하드웨어 RAID는 일반적으로 다른 메이커와 모델의 하드디스크에게 호환성을 제공하지 않지만, 리눅스 상의 소프트웨어 RAID는 어떤 특별한 설정 없이 대부분의 하드웨어가 잘 동작한다.

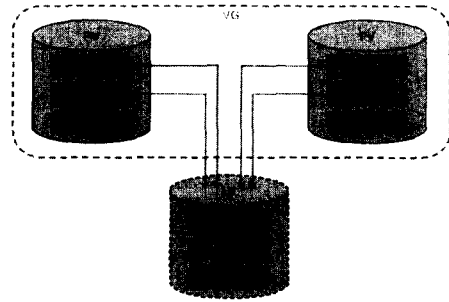
소프트웨어 RAID는 하드웨어 RAID에 비해 적은 비용으로 안정적인 시스템을 구성할 수 있다. 반면 CPU가 소프트웨어 RAID를 위해 많은 시간을 할당하게 되어 CPU를 많이 사용하는 시스템에서는 다소 성능 저하가 있을 수 있다[1].

2.3 LVM

LVM이란 복수의 디스크 또는 파티션을 종합해 하나의 논리적인 영역으로 취급하는 것이다. LVM으로 구성된 영역은 커널에서는 하나의 디스크처럼 보이기 때문에 디스크와 파티션 구성에 유연성을 갖도록 할 수 있다. RAID의 경우 여러 개의 디스크를 이용하여 속도와 안정성을 향상시킬 수 있다. 한편 LVM은 시스템 운영 시 동적으로 디스크 또는 파티션 구성을 바꿀 수 있는 유연성을 제공하는데 목적이 있다.

LVM에서 논리적인 볼륨은 작성한 후에라도 새로운 파티션을 추가 또는 삭제할 수 있다. 따라서 시스템 운영 중에 디렉토리의 크기를 변경할 수 있는 유연성을 제공한다. LVM의 개념도는 [그림 2]와 같다. 디스크 또는 파티션을 LVM전용 파티션으로 만든 것을 PV(Physical Volume)라고 한다. PE(Physical Extent)는 PV를 구성하는 기본단위로서 기본 크기는 4MB 이다. VG(Volume Group)는 복수개의 PV로 구성된 하나의 영역으로 가상의 하드디스크라고 할 수 있다. LV(Logical Volume)는 VG내의 PE 몇 개를 할당하여 만든 가상 파티션에 해당하는 것으로 실제 파일시스템이 작성되게 된다. LE(Logical Extent)는 LV에 할당된 VG내의 PE를 지칭한다. [그림 2]에서 예시하는 바와 같이 LV에 PE를 어떻게 할당하느냐에 따라 LV의 크기를 조정할 수 있는 유연성을 갖게 되

는 것이다[2][3][4].

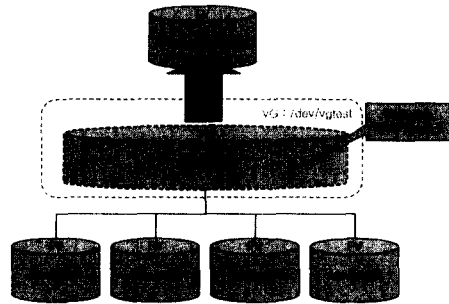


[그림 2] LVM 개념도

3. 소프트웨어 RAID 레벨 5와 LVM이 적용된 NAS

본 논문에서는 소프트웨어 RAID 레벨 5로 구성된 NAS에 LVM을 적용하여 성능과 안정성 향상, 그리고 관리의 유연성을 제공하는 시스템을 구축하였다. 호스트 컴퓨터는 AMD Athlon 1Ghz 시스템에 와우리눅스 버전 7.1을 설치하였으며 소프트웨어 RAID를 위한 스카시 인터페이스 카드와 4개의 스카시 디스크를 사용하였다.

리눅스 커널은 버전 2.4.7을 사용하였다. 소프트웨어 RAID는 리눅스 커널의 소프트웨어 RAID를 사용하였으며 LVM 1.0 버전을 사용하였다. LVM 패치를 적용한 후에 커널 컴파일을 하였다. 소프트웨어 RAID 및 LVM을 모듈 또는 포함시켜 커널을 컴파일 해야한다.



[그림 3] 소프트웨어 RAID 레벨 5와 LVM이 적용된 저장장치 시스템

본 논문에서 구축한 저장장치 시스템은 [그림 3]과 같다. 이 저장장치 시스템은 아래와 같은 순서를 거쳐서 구축된다.

- ① 디스크 4개를 PV로 만들고 소프트웨어 RAID 레벨 5를 적용하여 하나의 가상 PV를 생성
  - 리눅스에서 /dev/mdx로 인식[5]
- ② 가상 PV(/dev/mdx)를 VG에 포함 시킴
  - 임의로 VG의 이름을 vgtest로 설정
- ③ VG로부터 PE를 할당하여 LV를 생성
  - 임의로 LV의 이름을 lvtest로 설정
  - /dev/vgtest/lvtest 블록 디바이스 생성
- ④ /dev/vgtest/lvtest에 파일시스템을 작성하고 마운트하여 사용

현재 LVM으로 작성된 LV에 작성할 수 있는 파일시스템으로는 ext2, reiserfs와 xfs가 있다. reiserfs와 xfs같은 저널링 파일시스템을 사용하면 정전과 같은 상황에도 빠른 파일시스템 복구 가능한 고 가용성 기능도 추가할 수 있다. 그러나 본

논문에서는 일반적인 ext2 파일시스템을 사용하여 저장장치 시스템을 구축하였다. 이렇게 구축된 저장장치 시스템은 본 논문에서 구축한 NAS의 핵심이 되는 서버 시스템이다[3].

3.1 RAID 장치 추가를 통한 저장장치의 확장

앞에서와 마찬가지로 방법으로 소프트웨어 RAID 장치를 만든다. 기존의 RAID 장치는 이미 VG(vgtest)에 포함되어 있으므로 이러한 VG에 새로 작성한 RAID 장치를 포함시키면 VG에 포함된 PE의 개수가 늘어나 LV에 더 많은 PE를 할당할 수 있게 된다. 간단히 VG에 새로운 PV추가하는 것만으로도 LV의 사이즈를 확장할 수 있다. LV를 확장한 후에는 각 파일시스템에 해당하는 크기 재조정 도구를 이용하여 파일시스템의 크기를 조정해 주고 마운트하여 사용하면 된다.

4. 성능 평가

성능 평가 도구로는 bonnie++ 1.01d를 사용하였다. bonnie++은 POSIX 표준 C API를 사용하여 파일시스템의 입출력속도와 CPU사용률을 측정한다. NAS는 주로 대용량의 파일 서비스를 목적으로 하므로 본 논문에서는 bonnie++가 제공하는 성능 평가 항목 중에 블록 입출력 및 CPU 사용률을 측정하여 성능 평가를 수행하였다.

성능 평가에 사용한 디스크는 Quantum Atlas V 18.3GB 제품을 사용하였으며 인터페이스는 Ultra2 SCSI를 사용하였다.

성능 평가는 다음과 같이 3가지의 경우로 나누어 평가하였다.

- ① 디스크 1개를 사용한 경우(Single Disk)
- ② 디스크 4개로 소프트웨어 RAID 레벨 5를 구성한 경우(RAID5)
- ③ RAID5에 LVM을 추가한 경우(RAID5+LVM)

각각의 경우 100회씩 bonnie++을 사용하여 측정 후 평균값으로 결과를 정리하였다. 메모리로 인한 캐시효과를 제거하기 위해서 시스템에 설치된 메모리 용량인 256MB의 2배가 넘는 600MB을 파일을 생성하고 삭제하도록 하였다.

먼저 블록 쓰기/읽기에 대한 성능 평가 결과는 [표 1]과 같다. 블록 쓰기/읽기의 경우 모든 경우에서 성능이 RAID5 > RAID5+LVM > Single Disk 순으로 나타났는데 이것은 소프트웨어 RAID를 통한 성능 향상 때문이다. 여기서 주목할 것은 RAID5+LVM의 경우 RAID5보다 한 단계 더 소프트웨어 층을 거침에도 불구하고 성능 저하가 거의 없다는 점이다. 게다가 블록 읽기의 경우 RAID5와 RAID5+LVM이 Single Disk에 비해서 거의 2배에 가까운 성능 향상을 얻을 수 있었는데 대부분의 유닉스 파일시스템에서는 읽기와 쓰기의 비율이 65~80 대 20~35 사이를 차지할 만큼 읽기의 비중이 높다는 점을 고려할 때 읽기 성능의 비약적인 향상은 전체 시스템의 성능 향상에 많은 도움을 줄 것이다[6].

(단위 : Kbyte/sec)

	Block Write	Block Read
Single Disk	34835.5	28437.3
RAID5	46845.8	59220.2
RAID5+LVM	46702.7	54286.6

[표 1] 블록 쓰기/읽기에 대한 성능 평가

다음으로 블록 쓰기/읽기에 대한 CPU 사용률은 [표 2]와 같다. 블록 쓰기/읽기에서 CPU 사용률이 Single Disk일 때가 가장 낮게 나타났다. 그 이유는 Single Disk일 때는 기존의 파일 시스템을 그대로 사용하는 것이므로 CPU에 추가적인 오버헤드가 필요 없기 때문이다. 소프트웨어 RAID를 사용하므로 RAID5의 경우 Single Disk일 때보다 CPU를 더 사용하며

LVM이 추가된 경우 RAID5일 때보다 소프트웨어 층을 한 단계 더 거침으로 인한 CPU 사용률의 증가가 있으나 그 양이 매우 적다. 특이한 점은 블록 읽기의 경우에 RAID5일 때보다 RAID5+LVM의 CPU 사용률이 더 적은 경우가 발생했는데 이것은 [표 1]에서 보이듯이 RAID5가 RAID5+LVM보다 단위시간에 더 많은 읽기 작업을 수행하므로 더 많은 소프트웨어 오버헤드를 필요로 하기 때문이다.

블록 쓰기/읽기에서 RAID5와 RAID5+LVM의 경우 Single Disk에 비해 높은 CPU 사용률을 나타내지만 NAS는 파일 서비스만 담당하므로 다른 추가적인 CPU 부담이 필요하지 않다. 따라서 RAID5와 RAID5+LVM의 사용으로 인한 보다 높은 CPU 사용률이 NAS의 성능에 큰 영향을 미치지 않는다.

(단위 : %)

	Block Write	Block Read
Single Disk	14.7	10.7
RAID5	27.1	30.2
RAID5+LVM	27.8	27.0

[표 2] 블록 쓰기/읽기에 대한 CPU 사용률

5. 결론

본 논문은 소프트웨어 RAID와 LVM 기반의 네트워크 저장장치의 구현과 성능 평가에 관한 것이다. 성능 평가의 결과에서 보이듯이 RAID5를 사용하여 성능을 향상시킬 수 있었으며 LVM을 추가한다고 하더라도 성능 저하가 미약하므로 LVM을 추가하여 얻을 수 있는 장점을 고려했을 때 NAS에 LVM을 탑재하는 것이 적합한 선택임을 알 수 있다. 물론 RAID5와 LVM을 탑재함으로써 전체적인 CPU 사용률이 높아지는 단점이 발생하지만 NAS는 파일 서비스만을 제공하므로 추가적인 오버헤드가 필요하지 않다는 점을 고려해보면 이러한 단점이 시스템에 큰 영향을 미치지 않는다는 것을 알 수 있다.

따라서 RAID5를 사용하여 성능을 향상시킬 수 있고 LVM을 추가함으로써 저장장치 관리가 유연해 지므로 RAID5+LVM 기반의 NAS를 구축하는 것이 가장 적합한 선택임을 알 수 있다.

참고문헌

- [1] 손재기, 이형수, 민수영, 박창원, "임베디드 리눅스를 이용한 네트워크 저장장치 연구 개발," 자료저장 시스템 연구회 워크샵, 2001.
- [2] David Teigland, "Volume Manager in Linux," Sistina Software, Inc., 2001.
- [3] AJ Lewis, "LVM HOWTO," <http://www.sistina.com>.
- [4] 정은경, "Logical Volume Manager," Linux Magazine, July 2001.
- [5] Jakob Østergaard, "The Software-RAID HOWTO," <http://www.linuxdoc.org>.
- [6] J. K. Ousterhout, H. Da Costa, D. Harrison, J. A. Kunze, M. Kupfer, and J. G. Thompson, "A trace-driven analysis of the UNIX 4.2 BSD file system," in Proceedings of the 10th ACM Symposium on Operating System Principles, (New York, New York), pp.15~24, ACM, 1985.
- [7] Peter M. Chen, Edward K. Lee, Garth A. Gibson, Randy Kats, and David A. Patterson, "RAID: High-Performance, Reliable Secondary Storage," ACM Computing Surveys 26(2), pp.145~185, 1994.
- [8] Daniel Stodolsky, Mark Holland, William V. Courtright II, and Garth A. Gibson, "A Redundant Disk Array Architecture for Efficient Small Writes," Carnegie Mellon Technical Report, 1993.