SAN 환경 공유 파일 시스템의 온라인 리사이징

임 승 호, 이 주 평, 조 준 우, 박 규 호
한국과학기술원 전자전산학과
전화 : 042-869-5425 / 핸드폰 : 019-9182-4533

Online Resizing of Shared File System In SAN Environment

Seung-Ho Lim, Jupyung Lee, Joon-Woo cho, Kyu Ho Park
Conputer Engineering Research Lab. EECS
Korea Advanced Institute of Science and Technology
E-mail : shlim@core.kaist.ac.kr

Abstract

In this paper, we developed the scheme to grow to use newly added disk space without having to kill the application, unmount file system. This scheme, called online resizing, can resize the file system layout with the advent of Logical Volume Manager. The online resizing scheme is designed and implemented in linux cluster system where multiple hosts share the disk data in storage area network environment. It is incorporated with SANfs shared file system and can perform resizing technique with SANfs-VM volume manager. The experimental result shows that it can maximize the availability and capacity of the SANfs system which are important for modern servers where must not lose their customer.

keywords : LVM, SAN, shared file system, online resizing

I. 서론

폭발적인 인터넷 사용의 증가, e-business, 멀티미디어 데이터의 사용 증가는 저장장치 시스템의 대용량화와 함께 관리의 중요성을 가져오게 되었다. 최근의 Storage Area Network(SAN) 기술의 발전은 저장 장치 시스템의 획기적인 변화를 가져왔으며 SAN을 이용해서 여러 대의 시스템을 묶어, 공유 파일 시스템을 사용함으로써 대용량 시스템을 만들 수 있게 되었다.

특히 인터넷 서버의 성능을 높이기 위해서, 여러 대의 서버를 묶어서 클러스터링해서 사용하는 병렬 클러스터 기반 웹 서버에 대한 연구가 여러 분야에서 진행중인데, 이런 서버에서 가장 중요한 것이 서버의 사용자, 즉 고객을 잃어버리지 않는 시스템의 가용성과 대용량을 다룰 수 있는 확장성이다.

시스템의 작업을 중단하게 되는 여러 가지 경우중에 하나가, 데이터의 저장 공간이 부족해서이다. 시스템의 가용성을 증대시켜 사용자를 잃지 않기 위해서는, 시스템의 가동중에 데이터의 저장공간을 확보하는 방법이 필요하다. 이처럼, 새롭게 생성되는 데이터의 저장공간을 확보하기 위해서 시스템의 작업 수행 중에 볼륨의 크기를 변경시키는 것을 온라인 리사이징(Online Resizing)이라 한다. 이 논문에서는 SAN 환경의 공유 파일 시스템인 SANfs의 가용성과 확장성을 증대시켜 주기 위한 온라인 리사이징 방법에 대해서 연구해보고, 실제로 Linux 시스템 상에서 설계하고 구현함으로써 시스템의 가용성과 확장성을 극대화시켜줄 수 있는 기술을 개발하도록 한다.

2장에서는 SAN 환경의 공유파일 시스템인 SANfs

시스템에 대해서 알아보고, 3장에서 SANfs의 온라인 리사이징 기능에 대해서 설명한다. 4장에서 결론을 말하도록 한다.

II. SANfs 시스템

SANfs 공유 파일시스템은 SAN 환경에서 대규모 저장장치를 겨냥하여 뛰어난 확장성을 갖도록 설계하였으며, Linux상에서 개발되었다. SANfs의 시스템 구성도는 그림 1과 같이 나타낼 수 있다. SANfs는 하나의 메타서버와 다른 여러 개의 SANfs client들로 구성되어 있으며, 저장 장치 시스템과 SANfs client들이 SAN을 통해서 연결되어 있다. 저장장치들은 여러 가지 레벨의 configuraion을 가질 수 있으며, 이것들은 SANfs-VM을 통해서 하나의 전체적인 저장장치 공간을 형성하게 된다.

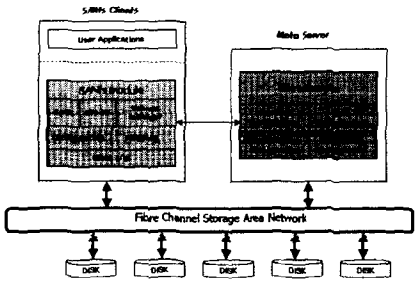

그림 1 시스템 구조

2.1 파일 데이터의 일관성 유지

클러스터 상황에서 서로 다른 호스트간에 동일한 파일을 공유하면서 쓰는 경우는 거의 일어나지 않고, block 단위의 consistency를 지원하는 것은 복잡하고 불필요하기 때문에, SANfs에서는 파일 데이터의 일관성을 block lock이 아닌 file lock을 통해서 유지한다. file lock의 관리는 중앙의 메타 서버를 통해서 이루어진다. 메타서버는 유저 레벨 프로그램으로 구현되어 있으며, 각 호스트는 자신이 수정하고자 하는 파일을 열 때, 메타서버로부터 그 파일에 대한 권한을 얻은 뒤 해당 파일에 대한 작업을 수행한다. lock의 종류는 읽기 전용 lock과 읽기/쓰기 lock의 두 가지가 있다. 한 호스트에서 액세스한 파일에 대해서는 해당 호스트가 다시 액세스하게 되는 가능성이 높기 때문에, 한번 그 파일에 대한 lock을 얻게 되면 다른 호스트에 의해서 lock 요청이 이루어질 때까지 그 호스트가 계속 lock을 가지고 있는 callback locking 방식을 사용한다.

2.2 free space 관리

SANfs 파일시스템의 free space는 메타서버에서 집중관리하며, 각 호스트는 free space가 필요할 때마다 메타서버로부터 가져온다. 메타서버의 부담 및 통신량을 줄이기 위해서 메타서버에서 한번에 2,048개의 block chunk를 가져온 다음 local process들의 free block요청을 서비스한다. Free block들은 unmount시에 메타서버로 되돌린다.

III. SANfs의 Online Resizing

볼륨의 온라인 리사이징은 실제로 그 볼륨의 디스크 공간을 사용하는 파일 시스템과 함께 확장해 주어야만 진정한 의미가 있다. 이것을 위해서는 파일 시스템 레이아웃의 변경이 필요하며, SANfs 레이아웃의 변경은 볼륨 관리자가 제공하는 정보를 이용하여 파일 매니저가 SANfs의 레이아웃을 확장시켜준다.

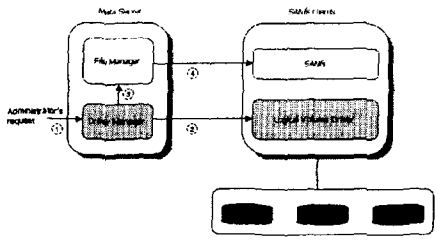

그림 2 온라인 리사이징 순서

그림 2는 SANfs 시스템에서 Online Resizing을 수행하는 과정을 나타낸 것이다. 온라인 리사이징 순서는 크게 두 단계로 나눌 수 있다. 첫 째, 관리자가 볼륨의 확장에 대한 요청을 하면, 메타 서버의 Driver Manager는 이전의 논리 볼륨과 확장되는 볼륨을 조합해서 새로운 Configuration과 Mapping Table을 생성해낸다. 그런 후, 이것을 각 호스트의 Logical Volume Driver에게 알려 볼륨의 확장을 시도한다. 이 때, SAN 환경의 모든 호스트의 볼륨에 일관성이 있도록 유지시켜주게 된다. 둘 째, 볼륨의 확장 후 Driver Manager는 확장된 정보를 File Manager에게 알려줘 SANfs 파일 시스템 레이아웃의 재조정을 하도록 한다. File Manager는 새로 추가된 볼륨을 SANfs에서 사용할 수 있도록 레이아웃을 변경하고, Metadata에 대한 정보를 포맷하고 SANfs의 각 호스트들이 사용할 수 있도록 수퍼 블록과 Metadata를 업데이트한다.

각 단계에 대해서 자세히 설명하도록 한다.

3.1 블록 디바이스 볼륨의 resizing

논리 볼륨의 확장은 크게 두 가지 경우로 나눌 수 있다. 기존의 논리볼륨의 Configuration을 확장하여 기존과 같은 Mapping을 가지도록 확장하는 방식과, 이전

의 Configuration과는 다른 새로운 Configuration을 만들어 논리 볼륨을 추가하는 방식이다.

첫번 째 경우, 기존의 논리볼륨에 추가로 논리볼륨을 확장하는 경우는 각 Configuration 레벨에 따라서 다르게 진행된다. Linear mapping된 논리 볼륨의 확장은, 간단하게 확장된 사이즈 만큼의 논리 볼륨과 물리 볼륨 사이의 매핑을 만들어 준 후, 기존의 매핑 테이블과 이어주면 된다. RAID 레벨의 Configuration의 경우에는, 물리적 디스크의 개수만큼 스트라이핑되는 인덱스가 변화하기 때문에 확장의 방식에 따라서 RAID 구성의 재조정이 있어야 한다. 볼륨의 확장으로 인해서 스트라이핑 사이즈와 스트라이핑 인덱스가 변화하게 되면, 매핑 관계도 변화하게 되고, 새로은 매핑 테이블에 맞도록 데이터의 이동이 수반된다. 데이터의 이동은 매핑 단위인 Extent단위로 이루어지게 되며, 데이터의 이동시 일반적인 애플리케이션의 작업에 방해가 되지 않도록 Extent 단위의 락을 사용하여 데이터의 Consistency를 유지시켜준다.

두번 째의 경우, 새로은 논리볼륨을 생성하는 경우에는 기존의 논리 볼륨에 영향이 없기 때문에 기존의 논리 볼륨에 대한 재구성이 필요하지 않다.

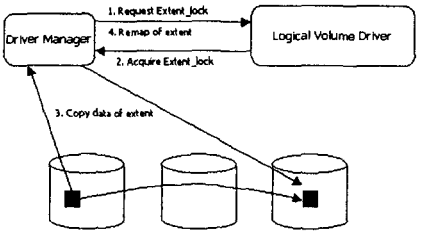

그림 3 데이터 migration 과정

시스템의 확장으로 인해 RAID 구성의 재조정이 필요해서 데이터를 한 곳에서 다른 곳으로 이동시키는 작업을 데이터 migration이라 한다. 데이터 migration의 단위는 *Extent*단위이다. 하나의 Extent는 디스크의 Sector offset으로 나타낼 수 있다. Extent를 migration하기 위해서는 그 Extent에 대한 lock을 소유해야 하며, 이를 위해서 SANfs 볼륨관리자는 *Extent_lock*을 관리한다. 그러나 *Extent_lock*은 데이터의 migration시에만 생성되기 때문에, 데이터 migration이 없는 평상시의 시스템 오퍼레이션에는 아무런 오버헤드가 되지 않는다. migration을 수행하는 방법은 다음과 같다. 먼저, Driver Manager는 각 호스트의 Logical Volume Driver로부터 migration하려는 Extent에 대한 lock을 얻어온다. Logical Volume Driver의 글로벌한 mapping table의 유지와 데이터의 일관성을 보장해 주기 위해서 모든 호스트로부터 락을 얻어올 필요가 있다. 모든 호스트로부터 Extent에 대한 락을 얻어온 후, Driver Manager는 Extent를 이동하려는 디스크로 copy한다. 그런 후 Driver Manager는 Remap 함수를 통해서 Extent에 대한 새로운 논리 주소 대 물리 주소의 매핑을 만들어 각 호스트의 Logical Volume Driver로 보내 매핑 테이블을 유지시켜준다. 이런 Extent의 migration이 일어나는 동안, 파일 시스템은 그 Extent의 블록에 수정을 가할 수 없다.

그림 3은 하나의 Extent에 대한 이동 과정을 나타낸 것이다. 이와 같은 이동을 RAID 구성의 재조정이 끝날 때까지 Exteny by Extent형태로 이루어지게 된다. Mapping 단위가 클수록, 옮기는 Extent의 갯수가 적어지므로 전체 migration 시간이 짧아지는 반면, 정상적인 파일 시스템 작업을 수행할 수 없는 영역도 커지기 때문에 시스템 작업의 딜레이 타임은 길어질 것이다. 반대로 mappint 단위가 작으면 전체 migration은 길어지지만 시스템 작업의 딜레이 타임은 짧아지게 된다.

3.2 파일 시스템 레이아웃의 resizing

볼륨 관리기는 file manager와 함께 볼륨의 확장과 더불어 파일 시스템의 온라인 확장을 해 줄 수 있도록 구성되었다. SANfs 파일 시스템의 온라인 리사이징 지원을 위해서 본 논문에서 새롭게 제안된 SANfs 의 파일 시스템 레이아웃은 그림 4와 같다.

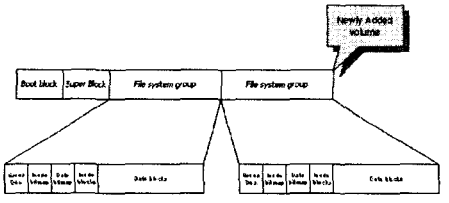

그림 4 SANfs 레이아웃의 변경

Boot Block과 Super Block 이후의 Inode와 data 에 관련된 block들은 여러개의 파일 시스템 그룹으로 나타내어진다. 각각의 파일 시스템 그룹에는 그 그룹에 속하는 Inode에 대한 bitmap과 블록, data에 대한 bitmap과 블록을 가지고 있으며, 이런 정보를 포함하고 있는 Group Descriptor가 자신의 그룹을 관리한다. 그룹 디스크립터의 구체적인 자료구조는 다음과 같은 항목이 있다.

- group_number 그룹의 번호
- group_offset 그룹이 시작하는 데이터 블록의 번호

- group_ninodes 그룹내의 Inode의 수
- group_nzones 그룹 내의 Data block의 수
- group_imap_blocks Inode bitmap의 블록 수
- group_zmap_blocks Data block bitmap의 블록 수
- group_firstdatazone 그룹 내의 첫번째 Data block 이 시작하는 블록의 번호

수퍼블록은 파일 시스템 전체의 Inode와 data에 대한 정보를 가지고 있을 뿐만 아니라, 각 그룹들의 정보도 함께 가지고 있다.
수정된 수퍼 블록에 대한 데이터 구조는 다음과 같다.

```
struct super_block {
        ...
        unsigned short s_group_number;
        fdfs_block_group **s_group_des;
        ...
}
```

Metadata 즉, Inode와 block에 대한 관리가 메타 서버의 파일 매니저에서 이루어지기 때문에, 파일 시스템 레이아웃의 변경은 메타 서버의 파일 매니저에서의 수퍼블록 변경과 그룹의 추가만으로 가능하다. Driver Manager가 확장된 볼륨에 대한 정보를 파일 매니저에게 보내주면, 파일 매니저는 새로운 볼륨에 대해서 Group descriptor를 이용해 하나의 그룹을 생성하고 이것을 수퍼 블록에 등록한다. 파일 매니저는 변경된 수퍼블록에 대한 업데이트를 하게 되고, 새로 등록된 그룹에 대한 Inode와 Free block에 대한 bitmap은 linked list 형태로 기존의 파일시스템의 그룹들과 연결되게 된다. 그리고 이 변경된 정보에 대해서는 각 호스트가 remount 옵션을 이용해서 변경된 레이아웃에 대한 업데이트를 해 줄 수 있다. SANfs의 각 호스트들에 대한 Inode와 Free Block 할당은 기존의 방식과 같으며, 파일 매니저는 한 그룹에 속한 Inode와 Block에 대해서 빈 Inode나 Block을 할당하지 못하면 다음 그룹의 Inode와 Block을 할당해준다.

IV. 결론

인터넷 사용의 증가와, 서버 성능의 향상으로 인해서 서버 시스템의 가용성과 확장성은 더욱 중요한 이슈로 자리잡게 되었다. 서버의 가용성을 극대화시켜주어 고객을 잃지 않기 위해서 여러 가지 기능이 필요한데, 그 중 중요한 기능이 온라인 리사이징 기능이다.

본 논문에서는 SAN 환경의 대용량 파일 시스템을 기반으로 하는 웹 서버 시스템의 가용성과 용량의 동시 확보를 위해서 SAN 환경 공유 파일 시스템인 SANfs의 온라인 리사이징 기능을 제안하고 구현하였다. 그 결과 시스템의 사용 시간을 극대화 시켜 줌과 동시에 시스템의 저장 공간의 확보를 충족시켜 줄 수 있었다.

참고문헌(또는 Reference)

[1] Heinz Maulschagen. Logical Volume Manager for Linux. http://linux.msede.com/lvm
[2] David C. Teigland, The Pool Driver: A Volume Driver for SANs. http://www.sistina.com
[3] A. Dilger, "Online ext2 and ext3 Filesystem Resizing", Ottawa Linux Symposium, 2002
[4] T.Ts'o, "Planned Extensions to the Linux Ext2/Ext3 Filesystem", USENIX Annual Technical Conference, 2002
[5] SANfs Shared File System. http://core.kaist.ac.kr
[6] Joo Young Hwang, Chul Woo Ahn, Se Jeong Park, Kyu Ho Park "A Scalable Multi-Host RAID-5 with Parity Consistency" IEICE transactions on Information and Systems VOL.E85-D No.7 JULY 2002
[7] S.H.Lim, J.Y.Hwang, K.H.Kim, J.P.Lee and K.H.Park,"Resource Volume Management in SAN Environment", PDCS, 2003
[8] S.H.Lim, "Design and Implementation of Volume Manger for Shared File System in SAN Environment", M.S. Thesis