

SCSI 명령어를 이용한 SANtopia 백업시스템 설계 및 구현

오상기*, 최경웅*, 김정기*, 신범주**, 김창수**, 박순철*

*전북대학교 정보통신공학과

**전자통신연구소

{smartki,pinkjaeho,xb@internet.chonbuk.ac.kr}

{bjshin,cskim77@etri.re.kr}

Desion of SANtopia Backup system and implemented using of SCSI Command

Sang-Gi Oh*, Kyong-Ung Choi*, Goung-Gi Kim*

**Bum-Ju Shin, **Chang-Su Kim, *Soon-Cheol Park

**Dept of Infomation & Telecommunication Engineering, Chonbuk
National University, Eletronic Telecommunications Resarch Institute

요약

본 연구는 데이터웨어하우징/의사결정시스템, 그리고 업무 처리 자동화 소류선등 새로운 애플리케이션 개발로 이하여 발생하는 엄청난 양의 데이터를 효율적으로 관리하기 위한 백업 저장 시스템을 SAN 환경을 지원하는 SANtopia 파일시스템에서 동작하도록 설계 구현하였다. 지금까지 데이터 저장시스템은 하나의 서버에 접속되어 있어서 네트워크 상의 다른 컴퓨터가 직접 활용하는 것을 허용하지 않기 때문에 기존의 네트워크를 기반으로 하는 파일서버 형태로는 이러한 대용량의 자료를 저장하고 관리하는데 한계가 있다. 이러한 문제를 해결할 수 있는 방법으로 Fibre Channel과 LAN 네트워크 모델을 결합하여 대역폭(Bandwidth)의 병목 현상을 크게 완화할 수 있는 SCSI 명령어를 이용하여 SANtopia 파일 시스템에서 파티션별 Full backup과 Restore 할 수 있는 시스템을 리눅스 기반에서 구현하였다.

1. 서론

기업에 있어서 정보는 중요한 자산이다. 그 정보를 안전하게 보관하는 것은 성공의 중요한 요소이며, 백업 저장 작업은 그 정보를 보호하는 핵심 요소이다. 그런데 기존의 저장 시스템 환경은 주로 SCSI(Small Computer System Interface) 채널을 매개로 적은 수의 저장 장치가 서버에 연결되어 있기 때문에 시스템을 관리하는데 있어 특별한 기능을 요구하지 않았지만 데이터의 증가에 따라 서버에 많은 부하를 주게 되었다. 이러한 문제로 네트워크에 저장 장치를 직접 연결하여 여러 컴퓨터들이 이 저장시스템을 공유할 수 있도록 해주는 SAN이라는 새로운 개념의 저장 환경이 대두되었다. SAN은 개별적으로 연결되던 저장 시스템을 별도의 전용 네트워크에 연결해 중앙 집중적인 관리가 이루어지도록 하여 데이터의 신속한 접근, 데이터의 가용성 향상, 데이터 손실의 가능성을 배제하는 높은 신뢰성을 제공함으로써

새로운 시스템 구조로 등장하였다.

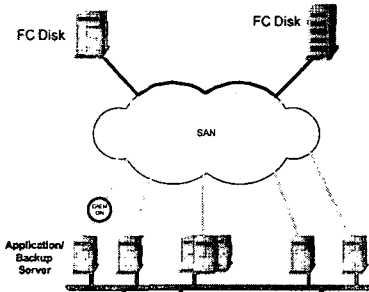
2. SANtopia 파일 시스템

SANtopia 파일 시스템은 SAN 환경을 기반으로 하는 클러스터 파일 시스템이며 64비트 어드레스를 지원하는 시스템으로 익스텐트(1KB~64KB)를 할당의 기본 단위로 한다. 이 시스템은 논리 볼륨 관리자를 하부 계층으로 사용하며, 클라이언트/서버 모델의 공유 파일시스템과는 구별되는 대등 관계 모델의 공유파일 시스템이다. SANtopia가 동작하는 시스템 환경은 디스크, JBOD, RAID 및 백업 장비와 이를 사용하기 위한 호스트들의 화이버 채널로 연결된 SAN 환경에서 각 호스트에서 수행되는 Linux 운영 체제에 포함된 시스템 소프트웨어다.

3. SANtopia 백업 시스템의 전체적인 구조

본 시스템의 구조는 클라이언트와 백업 서버 구

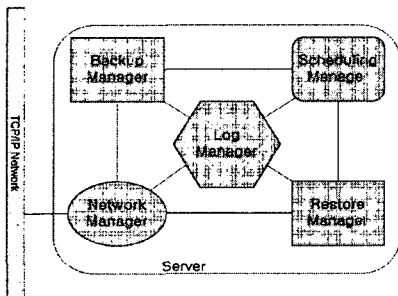
조의 모델을 취하고 있다. 클라이언트는 플랫폼에 종속적이지 않도록 중립성과 이식성을 고려해서 자바 언어로 개발, 어느 플랫폼에서도 백업 데몬에 접속을 할 수 있도록 하였다. 백업 데몬은 SAN 환경에 연결되어 있는 호스트에 항상 상주를 하도록 해서 클라이언트로부터 접속 및 요구를 처리하도록 하였다. 또한 백업 서버는 SCSI-3 Extended Copy Command를 지원하는 장비의 경우 데이터의 이동이 서버를 거치지 않는 Serverless 백업을 구현하도록 하였다. 전체적인 시스템 구조는 다음과 같다.



<그림 1> 백업 시스템의 전체적인 구조

4. 백업 서버 구조

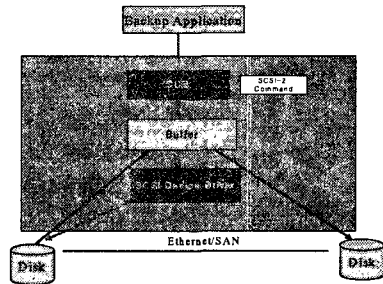
백업 서버는 크게 5가지 모듈로 나뉘어져 있다. 각각의 모듈은 첫째 클라이언트로부터 각종 요구와 백업 및 리스토어 정보, 스케줄링에 관한 정보 등을 요구하고, 서버 쪽에 필요한 정보를 전송하는 네트워크 모듈이 있고, 두 번째로 네트워크 모듈로부터 필요한 정보를 받아서 실제적인 백업을 담당하는 백업 모듈, 세 번째로 백업한 것을 원래의 디스크로 복원하는 리스토어 모듈이 있다. 네 번째로 미리 정한 백업 정책에 따라 실행을 시킬 수 있도록 스케줄을 관리하는 스케줄링 모듈, 다섯 번째로 지금까지 위의 모듈에서 작업한 내용이나 작업을 위해 생성된 정보를 저장하기 위한 로그 모듈로 나뉘어져 있다.



<그림 2> 백업 시스템의 전체적인 구조

4-1 백업 서버의 기능

백업 서버는 클라이언트 요청이 있을 경우 백업 서버가 떠 있는 호스트의 파티션 정보 및 지금까지 백업을 수행했던 기록 그리고 호스트의 스케줄링 정보를 클릭하여 정보를 제공해준다. 본 백업 서버의 주된 기능중의 하나는 호스트의 디스크를 전체 혹은 부분적인 파티션별로 백업을 하고 리스토어 하는 것이다. 그리고 클라이언트/백업 서버 모델을 취함으로써 원격에서의 관리를 지향했으며 자바 인터페이스를 도입 확장성과 이식성을 높였다. 또한 보다 빠른 백업을 구현하기 위해 하위 레벨의 전송 구조의 SCSI 프로토콜을 사용하여 서버의 메모리를 경유하지 않고 SAN 환경에 직접 장착이 된 디바이스들 간에만 데이터들이 전송할 수 있도록 했다.



<그림 3> SCSI-2 명령어의 흐름 및 데이터 흐름

4-2 네트워크

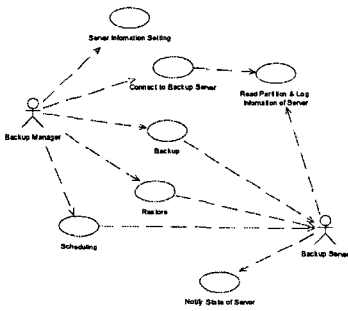
네트워크 모듈은 클라이언트와의 연결 통로를 만들어준다. 클라이언트와 백업 서버 사이의 통신은 TCP/IP를 기반으로 하는 소켓(socket)을 사용한다. 클라이언트가 백업 서버 프로세서에 연결 및 단절을 수행하고 클라이언트나 서버 모두의 요구사항이다 실행 결과를 메시지 형태로 상대방 측으로 전달한다. 그러나 서버 쪽은 C로 구현하고 어플리케이션은 자바로 구현되었기 때문에 서로 다른 언어 사이의 통신은 네트워크 바이트 순서가 중요시된다. 따라서 자바와 C프로그램 사이에서 데이터를 주고받으려면 기본 유형으로 주고받아야만 한다.

4-3 SCSI 명령어

SCSI 명령어의 수행은 스카시 커맨드 블록(SCSI Descriptor Block)을 초기화하는 부분과 이 명령어를 가지고 스카시 블록 디바이스 드라이버의 시스템

호출을 한다. 스카시 커맨드(cmd)의 길이는 첫 번째 필드의 값으로 정해진다. 백업 수행은 백업 데몬에 의해서 요구된 파티션 백업을 수행하게 된다. 이때 사용되어지는 스카시 커맨드는 READ(10)과 Write(10)이다. 백업 수행 루틴은 우선 지정된 파티션에 대한 장치 파일을 찾고 있으면 그 장치 파일을 open해서 파일 디스크립터를 얻는다. 이후 READ_CAPACITY 커맨드를 이용하거나 ioctl에 BLKGETSIZE의 파라미터를 이용하여 장치 파일의 총 논리 블록 어드레스를 얻는다. 블록 어드레스를 찾는 이유는 해당 개수의 만큼 루프를 돌면서 백업을 하기 위해서이다. 이후 위에서 언급한 스카시 커맨드 수행에 필요한 절차를 마친 후 완성된 CDB를 ioctl 시스템 호출을 이용, 백업을 수행하게 된다.

5. 백업 시스템 어플리케이션
5-1 UseCase Diagram



<그림 4> 백업 시스템 클라이언트의 UsecaseDiagram

Backup System의 Client부는 백업 시스템 관리자에게 백업 시스템의 Graphical한 화면 Interface를 제공해주며, Client-Server 환경에서 백업서버에 TCP/IP 기반의 Socket을 이용한 접속을 위해 서버의 주요 정보(IP, Port Number, User ID, Passwd 등)를 설정하고, 실제적인 연결을 설정하며, 백업에 필요한 여러 가지 Tool을 제공한다.

5-2 클래스 다이어그램

① 연결 클래스

클라이언트와 서버의 연결은 TCP/IP 프로토콜을 사용하여 연결하게 되는데, 이때 클라이언트에서 필요한 서버의 정보는 미리 설정되어 있다. 먼저는 서버에 정보를 이용하여 소켓을 만들어 서버와의 연결을 설정하게된다. 일단 서버와의 연결이 인증 과정

에서 문제없이 올바르게 설정되면, 초기 정보들인 서버의 현재 파티션 정보와 그간의 백업로그 정보를 요청하여 이를 얻어오게 된다. 다만, 이에 대한 처리는 다시 Read_partition 클래스와 Read_BackupLog 클래스들에게 위임하게 된다.

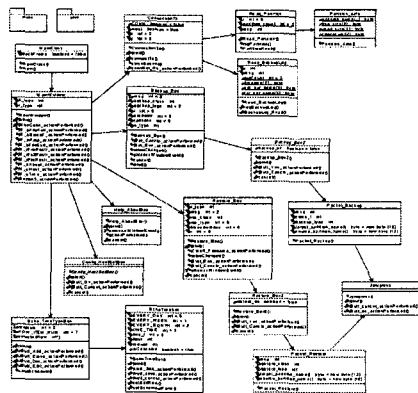
②백업 클래스

이 클래스는 백업 실행을 위한 파티션 정보를 이미 저장된 클래스에서 리스트박스로 불러 보여주고 이를 선택함으로써 수동적인 백업을 가능하게 만들어 준다. 이 클래스 하나로 여러 타입의 백업 실행의 요청을 처리 할 수 있도록 한다.

③복구 클래스

시스템이 예기치 않은 내부 오류나, 외적인 재해나 재난으로 인하여 데이터를 복구해야 할 일이 발생 시에 관리자는 이 기능을 이용하여 복구할 수 있다. 복구는 그리 흔히 있는 일이 아니고, 흔히 있어서도 안되는 일이므로 자동화된 Tool 보다는 수동으로 효과적으로 복구를 해야한다.

④ 전체적인 클래스 다이어그램은 아래와 같다.



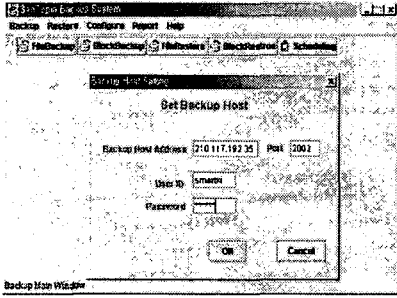
<그림 5> 클래스 다이어그램

6 구현결과

6-1 서버 정보

서버에 접속하기 위한 정보를 입력한다.

- ▶Backup Host Address - 백업을 수행하기 위해 정보를 전달할 서버의 주소와 포트 번호를 입력한다.
- ▶UserID, Password - 서버에 연결하기 전에 사용자 이름과 Password를 입력한다.

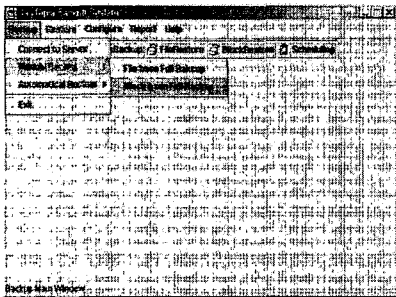


<그림 6> 서버정보 설정

6-2 Backup 메뉴

Backup 메뉴 바의 종류는 다음과 같다.

- ▶ Connect to Server - 백업에 관련된 정보를 전달하기 위하여 서버에 접속한다.
- ▶ Manual Backup - 백업할 종류를 선택한다.
- ▶ Automatic Backup - 스케줄링 정보에 따라 백업하기 위한 환경을 설정한다.
- ▶ Exit - 백업 어플리케이션을 종료한다.

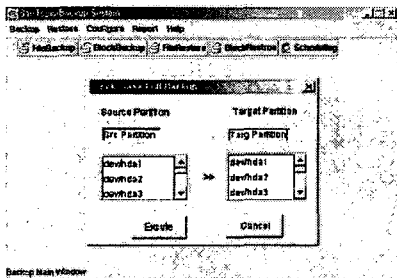


< 그림 7> 백업 메뉴

5-3 Backup 정보

백업할 데이터가 존재하는 파티션을 선택한다.

- ▶ Source Partion - 백업할 데이터가 존재하는 스토리지의 파티션을 선택한다.
- ▶ Target Partioion - 백업해야할 다른 스토리지의 파티션을 선택한다.

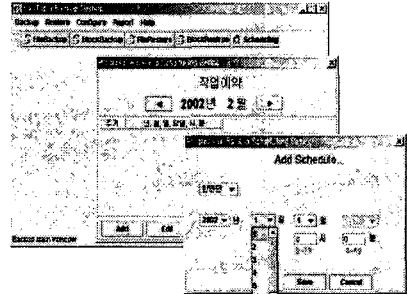


<그림 8> 백업 파티션 정보입력

5-4 작업예약

스케줄링 정보를 입력한다.

- ▶ Configuration에서 Schedule을 누르면 작업 예약을 할 수 있다.
- ▶ 선택 - 백업을 정해진 시간에 한번만 할 것인지, 매번 할 것인지 체크한다.
- ▶ 년도를 선택하고 달과 월, 시간을 선택한다.



<그림 9> 스케줄링 정보 입력

참고문헌

- [1] 신병주, 김병배, 김창수, 김명준 “네트워크 저장 장치를 위한 클러스터 파일 시스템 개발: 정보처리 학회 제 8권 제 4호 Jul 2001
- [2]A Silberschata, "Operation System Concepts 3rd Edition", Addison-Wesley Publishing company,1993
- [3]"SAN Data Protection", Legato Systems Inc., 2000, <http://www.sanacademy.com/SAN110/>
- [4]X3T 9.2 "Information technology - Small Computer System Interface-2"
- [5]Eric Youngdale, "Benchma가 raw SCSI I/O performance under Linux", <ftp://tsx-11.mit.edu/pub>
- [6]Eric Youngdale "Reads various mode pages and bits of other information from a SCSI device
- [7]Heiko Ei Feldt. "The Linux SCSI programming HTO" <http://rscl.auto.inha.ac.kr/~treeman/Document>
- [8]Alessandro Rubini, " Linux Device Drivers O'Reilly & associates, 1988
- [9]Lyon, "Design and Implementation of the Sun Network File System", In Proc. of the Summer 1985 USENIX, June 1985, pages 119-130.