

네트워크 스토리지 기술 동향

신 범 주*, 김 경 배**

*밀양대학교, **한국전자통신연구원

I. 서 론

클러스터(Cluster) 시스템은 독립적으로 동작할 수 있는 컴퓨터들을 네트워크로 연결하여 사용자 및 응용 프로그램에 단일 이미지를 제공하는 시스템을 의미한다. 클러스터 시스템은 가격 대 성능비가 뛰어나고 확장성(High Scalability), 고 가용성(High Availability)을 지원하기 때문에 다양한 컴퓨팅 분야에 활용되고 있으며, 특히 인터넷 컴퓨팅을 위한 기반 구조로 각광을 받고 있다.

그러나 이 같은 클러스터 시스템도 스토리지가 각 노드에 독립적으로 부착되기 때문에 응용의 특성에 따라 특정 자료가 존재하는 노드에 로드 가 집중되어 성능 병목 현상을 유발할 수 있다. 네트워크 스토리지는 고속의 네트워크에 스토리지와 컴퓨터들을 연결하여 여러 노드들이 공유할 수 있게 함으로써 클러스터 구조가 제공하는 고 확장성, 고 가용성, 고 성능(High Speed)과 같은 장점뿐만 아니라 자료의 공유가 가능케 하기 때문에 상기에 기술한 바와 같은 특정 서버의 병목 현상을 제거할 수 있게 한다.

네트워크 스토리지 시스템은 공유되는 주체가 무엇인지에 따라 NAS(Network-Attached Storage)와 SAN(Storage Area Network)으로 구분한다. NAS는 기존의 인터넷을 기반으로 이기종 클라이언트 환경에서 CIFS(Common Internet File System)이나 NFS(Network File System)를 이용하여 파일 단위의 전송 서비스를 제공하는 반면, SAN은 기존의

망 외에 서버와 스토리지를 연결하는 별도의 파이버 채널(Fibre Channel)을 이용하여 서로 연결된 서버와 스토리지가 블록 단위의 전송 서비스를 제공하는 시스템이다.

초기의 SAN은 파이버 채널이라는 전용 네트워크를 이용하여 스토리지를 연결하는 시스템을 의미하였으나, 최근에는 IP(Internet Protocol)를 기반으로 스토리지를 연결하는 방법 등을 총칭하고 있다. IP를 기반으로 하는 스토리지는 TCP/IP 위에 운영체제 혹은 장비의 SCSI와 인터페이스를 담당하는 별도의 프로토콜을 추가한 것이다. 현재의 파이버 채널 SAN에서 미래의 IP SAN으로의 이전을 위한 다양한 방안으로 Internet Engineering Task Force(IETF)에서는 FCIP, iFCP, iSCSI 3가지의 IP 스토리지 프로토콜을 제시하고 있다.

상기에 기술한 바와 같은 하드웨어적인 네트워크 스토리지를 어플리케이션에서 활용하기 위해서는 스토리지 가상화와 공유 기능을 제공하기 위한 논리 볼륨 관리 기술과 클러스터 파일 시스템 기술이 요구된다. 볼륨 관리 기술은 네트워크에 연결된 여러 개의 물리적 디스크를 이용하여 대용량의 스토리지 볼륨을 사용자에게 제공하는 기술이다. 클러스터 파일 시스템은 가상화된 대용량의 논리 볼륨에 대용량의 파일을 저장하고 관리하기 위한 대용량 파일시스템을 사용자에게 제공하고 네트워크 스토리지에 연결된 서버들이 파일을 공유할 수 있는 기능을 제공한다.

네트워크 스토리지와 같은 대용량 스토리지 시스템의 등장으로 가장 각광을 받고 있는 분야가 백업이다. 기존의 백업 방식은 소량의 데이터를

백업하기 위한 방안으로 개발되어 테라 바이트 이상의 스토리지를 백업하기 위해서는 다양한 형태의 백업 기술이 요구된다.

본 고에서는 II장에서 네트워크 스토리지 시스템의 하드웨어 기술인 NAS, SAN, IP 스토리지 기술에 대하여 소개하고, III장에서 네트워크 스토리지를 지원하기 위한 스토리지 시스템 소프트웨어 기술인 논리 볼륨 관리 기술과 대용량 공유 기능을 제공하는 클러스터 기반의 파일 시스템 기술에 대하여 설명한다. IV장에서는 데이터의 안전성을 보장하기 위한 백업 기술에 대하여 설명을 하고, V장에서 결론을 맺는다.

II. 네트워크 스토리지 시스템

1. NAS

NAS는 TCP/IP 네트워크 상에서 이기종 클라이언트에 파일 단위의 입출력을 제공하는 장치이다. NAS는 대표적인 네트워크 스토리지로 단일 장치에서 파일 서비스에 최적화된 운영체제와 RAID 기능 및 논리 관리자 등의 통합된 관리기능을 제공함으로써 기존의 파일 서버를 대체하면서 저장장치 분야에서 비중을 높여 가고 있다. 현재 제품화된 NAS 시스템 등은 고성능 프로세서

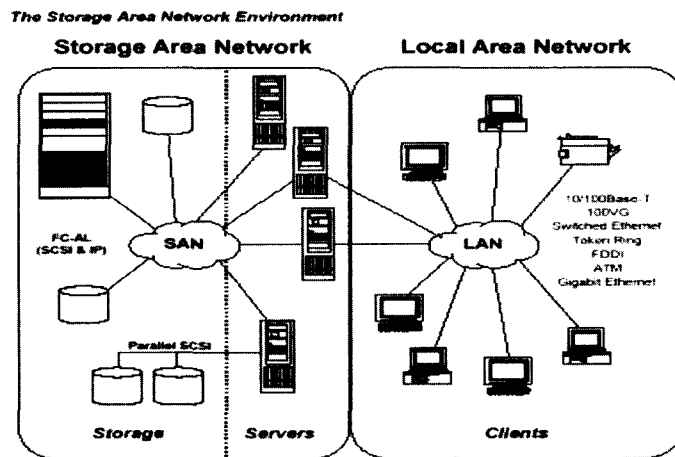
와 대용량의 저장공간을 이용하여 대규모의 사용자들에게 네트워크 상에서의 파일 서비스를 제공하고 있다.

NAS시스템은 NFS(Network File System)나 CIFS(Common Internet File System) 기반의 클라이언트에게 파일을 서비스할 수 있다. 소형 파일에 있어서는 클라이언트에 직접 연결한 파일과 성능 차이가 거의 없는데 이것은 NAS에 설치된 전용 프로세서와 RAID 기능 덕분이다. 이 기능은 NAS 시스템이 소규모 그룹의 파일 서비스 업무를 탁월하게 수행하도록 한다.

NAS 시장은 장점에도 불구하고 몇 가지 성능상의 단점을 지니고 있다. NAS가 아주 낮은 네트워크 오버헤드를 제공한다고 해도 다수의 클라이언트를 서비스하면 종종 과부하에 걸린다. 이외에도 확장성, 백업, 그리고 대용량 데이터베이스 등의 지원에서 한계를 지니고 있기 때문에 이를 해결하기 위하여 클러스터 NAS와 같은 기술들이 개발되고 있다.

2. SAN

SAN은 분산되어 있는 저장장치를 통합운영하여 업무의 효율성증대, 인프라에 대한 중복투자 방지 및 기종간 데이터 공유를 목적으로 만들어진 서버와 저장장치간의 네트워크를 의미한다. SAN의 활성화는 파이버 채널의 기본규약이 1994



<그림 1> SAN의 구성도

년 미국 표준협회(ANSI)인증을 계기로 수 많은 업체가 SAN 관련제품을 개발하고 있으며 현재 SAN 표준화 단체인 SNIA(Storage Networking Industry Association), FCIA(Fibre Channel Industry Association) 등에서 SAN 관련제품들에 대한 호환성 검증 및 표준화를 수행하고 있다.

SAN은 <그림 1>과 같이 구성되어 있으며, 파이버 채널(Fibre Channel)을 이용하여 스토리지 장치와 서버를 연결하는 전용 네트워크를 구성하고 있다. SAN 하드웨어 환경을 보다 효과적으로 활용할 수 있기 위해서는 사용자 또는 상위 계층의 응용에 하드웨어 투명성(Transparency)을 지원할 수 있는 일관된 기능의 인터페이스를 제공하는 것이 필요하다. SAN 가상화(SAN Virtualization)로 불리는 이러한 기능은 Virtual Memory의 개념과 유사하다. 소프트웨어로 제공되는 SAN 가상화는 시스템에 따라 다양한 기능들이 제공될 수 있으나, 일반적으로 소프트웨어 RAID, Snapshot, Storage on Demand 등의 기능을 포함한다.

3. IP 스토리지

TCP/IP와 이더넷 기술 개발은 여러 종류의 안정되고 상호 접속 가능한 제품 생산을 가능하게 했다. 고성능의 기가비트 이더넷 백본 스위치에서 정교한 압축 및 승인 도구까지, 고객의 여러 가지 요구를 충분히 충족시켜 왔다. IP 스토리지는 TCP/IP 위에 운영체제 혹은 장비의 SCSI와 인터페이스를 담당하는 별도의 프로토콜을 추가한다. 현재까지 광 채널 SAN에서 미래의 IP SAN으로의 이전을 위한 다양한 방안을 제시하고 있는 Internet Engineering Task Force(IETF)에서 FCIP, iFCP, iSCSI의 3가지 IP 스토리지 프로토콜을 개발 중이다.

1) Fibre Channel over IP (FCIP)

FCIP는 TCP/IP의 모든 광 채널 프레임을 둘러싸기(wrapping) 위한 터널링 프로토콜이다. 즉, FCIP는 TCP/IP를 통해 파이버채널 터

널링을 제공하고 IP 패킷 안에 파이버채널 프로토콜을 캡슐화 한 것이다. FCIP는 원격지에 파이버채널 SAN을 연결하기 위해 사용되며, 단지 프레임 전송만을 위해 TCP/IP를 사용한다. 표준 광 채널 스위치 연결을 이용하여, FCIP로 연결되어 있는 SAN들은 이전에 분리되어 있던 패브릭 상태가 아닌 하나의 거대한 파이버 채널 패브릭을 생성한다. 전송 측에서는 광 채널 데이터를 TCP/IP에 탑재하여 네트워크를 통해 전송하고 수신 측에서는 TCP/IP 헤더가 제거되고, 본래의 파이버 채널 데이터만이 목적지 스위치에 남게 된다. 따라서 FCIP는 최소의 IP 내용만을 포함하고 있기 때문에 파이버 채널의 확장과 지속적인 사용을 위한 전략이라고 할 수 있다. FCIP는 이용하기 편하다는 이점 등으로 인해 특성상 원격 백업 및 재해 복구 시스템에 활용의지가 많다.

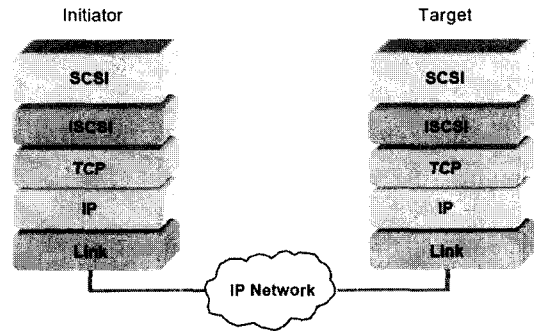
2) Internet Fibre Channel Protocol (iFCP)

iFCP는 IP와 SCSI 혹은 파이버채널 사이에서 게이트웨이의 역할을 한다. 이 프로토콜은 장비간 고유의 TCP/IP 연결을 제공하며, 현재 SAN 환경에 적용되어 있는 스토리지 시스템, 테이프 라이브러리 및 호스트가 광 채널 프로토콜을 사용하고 있다는 인식에서 개발되었다. FCIP가 원격지 연결을 통해 반대쪽 사이트에 파이버채널 패브릭을 유지하는 반면, iFCP는 파이버채널 장비를 직접 확장한다. 따라서 iFCP는 파이버 채널 스위치가 필요하지 않으며, 대신 IP 스토리지 스위치 네트워크가 필요하다. 또한 iFCP는 광 채널 라우팅 프로토콜에 독립적이기 때문에 광 채널 패브릭과 연관된 확장성 문제를 극복할 수 있다. 어플리케이션 관점에서 보면 iFCP는 기존에 사용 중인 인프라를 크게 변경하지 않기 때문에 소프트웨어와 하드웨어 제품에 대한 매우 높은 상호 접속성을 제공한다. 파이버 채널 SAN 환경에서 작성된 백업 및 스토리지 관리 애플리케이션은 상위 레이어의 인터페이스가 동일하기 때문에 iFCP 기반의 SAN 환경에서도 운영될 수 있다.

3) iSCSI (Internet SCSI)

SCSI는 높은 전송률과 신뢰성 등을 강점으로 저장장치와 서버를 연결하는 이상적인 프로토콜로 평가받아 왔다. 그러나 서버에 연결된 스토리지의 용량과 환경이 복잡해짐에 따라 SCSI는 물리적 속도, 스토리지의 공유 그리고 연결거리 등과 같은 문제가 발생하게 되었다. 이 같은 문제점을 해결하기 위한 것이 iSCSI는 IP 기반의 네트워크 환경에서 디스크와 서버간의 데이터 전송을 위한 SCSI 프로토콜을 사용할 수 있도록 하기 위해 IETF (Internet Engineering Task Force)의 IP 스토리지 그룹에서 제안한 표준 프로토콜이다. 즉, iSCSI는 TCP/IP 네트워크 환경에서 블록 스토리지 응용이 가능하도록 해주는 프로토콜이다.

SCSI는 높은 전송률과 신뢰성 등을 강점으로 저장장치와 서버를 연결하는 이상적인 프로토콜로 평가 받았다. 그러나 서버에 연결된 스토리지의 용량과 환경이 복잡해짐에 따라 SCSI는 물리적인 속도, 스토리지의 공유, 연결거리 등과 같은 문제가 발생되었다. iSCSI는 TCP/IP가 지니는 통신과 연결거리의 장점과 SCSI 지니는 신뢰성

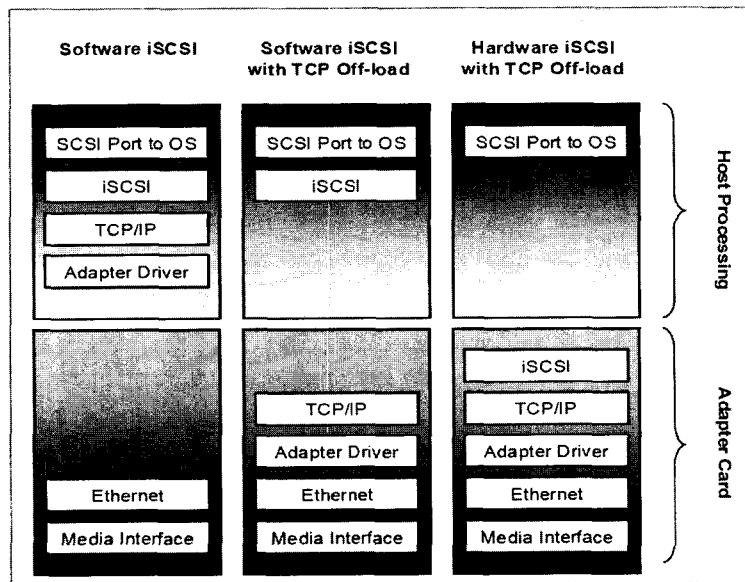


<그림 2> iSCSI 프로토콜 계층 모델

과 전송속도의 장점을 결합한 프로토콜이다.

<그림 2>는 iSCSI의 프로토콜의 계층을 보여주고 있다. iSCSI는 SCSI 명령어를 구동시키는 initiator와 initiator의 SCSI 명령을 받고 이를 처리해주는 target으로 구성된다. iSCSI는 구동된 SCSI 명령어를 iSCSI 프로토콜 스택에서 TCP/IP의 패킷 안에 캡슐화시켜 IP 네트워크를 통해 target에 도착하여 TCP/IP를 프로토콜을 거쳐 iSCSI 스택에서 SCSI 명령 만을 상위로 전송하여 처리한다.

iSCSI를 구현하기 위한 방법으로는 <그림 3>과 같이 3가지 방법이 있다.



<그림 3> iSCSI를 구현하는 3가지 방법

- S/W iSCSI : iSCSI 프로토콜을 호스트에서 처리하는 방법.
- TCP/IP Off-load를 한 S/W iSCSI : iSCSI 프로토콜은 호스트에서 처리를 하고 TCP/IP의 기능을 네트워크 카드에서 처리하는 방법
- H/W iSCSI : iSCSI 프로토콜을 모두 하드웨어로 처리하는 방법.

III. 네트워크 스토리지 시스템 소프트웨어 기술

1. 볼륨 관리 기술

볼륨 관리의 기능은 물리적으로 독립적인 여러 개의 저장 공간을 모아 하나의 가상적 기능을 제공하는 논리적 저장 공간을 상위 계층에 제공하는 것이다. 즉 SAN 가상화 기능들을 제공하여 물리 볼륨이 제공치 못하는 기능들을 제공한다. 파일 시스템, 데이터베이스 또는 일반적인 데이터 관리 시스템은 이 논리 볼륨 위에 생성되고 볼륨 관리자에 의해 제공되는 서비스를 이용하게 된다. 볼륨 관리자는 운영체제 커널의 가상 디바이스 드라이버로 동작하게 되며, 하나의 논리 볼륨은 하나의 가상 디바이스 파일로 표현된다. 볼륨 관리자는 SAN에 연결된 물리 저장 장치들을 인식하고 시스템 관리자가 정한 방식에 따라 물리 저장장치를 분할하여 논리 볼륨을 구성하고 관리한다. 즉 상위 계층의 모듈이 요청하는 I/O를 논리 볼륨의 특성에 따라 적절한 물리 볼륨 I/O를 실행할 수 있도록 해주는 모듈이다.

1) 볼륨 구성

볼륨 관리자의 역할 중 하나는 논리 볼륨을 구성하는 것이다. 독립적인 물리적 디스크 장치들을 한 곳으로 모아 저장 공간 풀(storage pool)을 형성하고 이들을 사용자의 요구에 따라 적절한 구성을 갖도록 배치하여 사용자가 원하는 용량만큼을 원하는 RAID level로 구성해 주는 역

할을 수행한다. 논리 볼륨은 디스크 파티션을 최소 구성 단위로 하며, 연속적인 주소 공간을 이루는 확장 가능한 디스크 파티션들의 집합이다. 논리 볼륨의 크기는 시스템 운영 중 언제든지 변경 가능하며, 디스크 파티션 단위로 변경이 행해지게 된다.

논리 볼륨의 구성 시에 고려되어야 하는 요소 중 하나는 익스텐트(extent)의 크기를 결정하는 것이다. 익스텐트는 동일 크기를 갖는 연속적인 공간이다. 또한, 정보의 저장을 위해 할당될 수 있는 디스크 공간의 최소 단위이다. 익스텐트의 크기는 하나의 논리 볼륨에 대해 동일하다. 그러나 서로 다른 논리 볼륨은 서로 다른 크기의 익스텐트를 가질 수 있으며 그 크기는 논리 볼륨 생성시에 결정된다.

논리 볼륨 구성에 대한 정보 및 해당 논리 볼륨을 관리하기 위한 메타데이터는 디스크 파티션의 헤더 영역에 기록된다. 메타데이터에는 맵핑 정보, 물리 디스크 영역의 할당 맵 등이 포함된다.

2) Software RAID

전통적인 디스크 관리 기법에서는 사용자에게 최대의 가용시간, 최적의 성능, 대용량 등의 요구 사항에 만족스러운 해법을 제시하지 못했다. 이러한 문제점을 해결하고자 하드웨어 RAID 시스템이 개발되었다. 하드웨어 RAID 시스템은 여러 개의 독립적인 디스크 장치들을 모아서 RAID 컨트롤러의 제어 하에 가용성, 성능, 용량적인 측면에서 사용자의 요구에 부응하도록 다양한 RAID 레벨을 제공한다. RAID 레벨은 저장 장치의 사용 환경이나 사용자의 요구에 따라, 스트라이핑(striping), 미러링(mirroring), 패리티를 갖는 스트라이핑(striping with parity) 등으로 구분된다. 하드웨어 RAID 시스템은 RAID의 관리를 RAID 컨트롤러가 수행하기 때문에 CPU의 활용도를 높일 뿐 아니라 버스 트래픽을 감소시킴으로써 높은 성능을 제공한다. 그러나 하드웨어 RAID 시스템의 비용이 매우 고가이며, 컨트롤러에 오류가 발생할 경우 모든

RAID 시스템을 사용할 수 없게 되는 문제를 안고 있다.

소프트웨어 RAID는 하드웨어 컨트롤러 대신 CPU가 RAID의 관리를 수행하게 된다. 이것은 컨트롤러의 오류 발생 시 모든 것이 정지되는 문제를 해결하며, 상대적으로 저렴한 가격의 RAID 시스템을 구성할 수 있게 해준다. 이렇게 가상화된 저장공간은 물리적 저장장치가 제공하는 한계를 극복할 수 있게 할 뿐 아니라 온라인 상에서 동적인 구성 환경을 지원할 수 있게 함으로써 사용자에게 많은 융통성을 부여할 수 있게 한다. 이러한 물리적 저장공간에 대한 가상화는 논리 저장공간과 물리 저장공간과 매핑 과정에 이루어지게 된다. 물리적 저장공간의 가상화는 논리 주소를 해당 물리 주소로의 매핑에 있어서의 가변성을 수용하는 것이다. 즉, 이러한 매핑이 계산에 의해서 이루어지는 형식처럼 고정되어 있는 것이 아니고 어떠한 상황이 발생하여 매핑 관계가 변해야 할 때, 이를 수용할 수 있는 형태이어야 하는 것이다.

3) Online Resizing

이미 존재하는 논리 볼륨의 크기를 재조정해야 할 상황이 발생할 경우가 있을 수 있다. 사용 중인 논리 볼륨의 크기를 확장 또는 축소할 수 있는 기능을 online resizing이라 한다. SAN-topia의 볼륨 관리자는 논리 파티션을 기본 단위로 하는 online resizing을 지원한다. 특정 볼륨에 논리 파티션의 추가 및 삭제는 해당 볼륨을 구성하는 모든 물리 디스크 장치의 특정 영역에 해당 정보의 갱신을 야기하며, 데이터의 이동을 동반할 수 있다.

볼륨을 축소하는 경우 삭제될 논리 파티션 내에 데이터가 없어야 한다. 만약 데이터가 존재한다면 매핑 관리자가 제공하는 데이터 이동 기능을 사용하여 해당 데이터를 다른 논리 파티션으로 이동시킨 후 삭제할 수 있다. 추가 시는 새로 추가된 파티션을 포함하여 전체 볼륨에 대해 RAID 구성이 재조정되어야 한다. 매핑 변환 함수 등의 사용에 의한 고정 방식의 매핑 기법에서

는 이러한 재조정 시 시스템의 일반적인 운영이 제한된다.

4) 스냅샷

대용량의 SAN 환경에서 백업을 효율적으로 처리하기 위해서는 스냅샷(snapshot) 기능이 특별히 요구된다. 스냅샷 기능은 매핑 관리자에 의해 제공된다. 매핑 관리자는 스냅샷 요구 시, 현재의 매핑 테이블을 그대로 복제한다. 복제된 매핑 테이블은 백업이 완료될 때까지 또는 다음 번 스냅샷이 요구될 때까지 유지된다. 사용자의 요구에 의해 데이터의 변경, 추가 혹은 삭제가 발생할 경우 해당 데이터의 디스크 익스텐트는 새로운 공간을 할당 받아 복사된 후, 새로이 할당된 영역에서 변경이 이루어지게 되는 변경 시 복제(copy-on-write) 기법을 사용한다. 즉, 원래 매핑 테이블은 스냅샷 시점의 매핑 정보를 그대로 유지한 채 관리되고 새로이 복제된 매핑 테이블이 사용자의 요구에 의해 변경된 매핑 정보를 유지하는 것이다. 이 상황에서 백업이 발생하면 백업은 원래의 매핑 정보를 이용하여 수행되게 됨으로써 스냅샷 시점과 일치하는 데이터의 백업을 취하게 된다.

2. 대용량 클러스터 공유 파일 시스템

SAN과 같은 고속의 신뢰성이 제공되는 네트워크 기반의 스토리지 환경에서 여러 대의 호스트가 연결되어 네트워크에 연결되어 있는 스토리지를 공유할 수 있는 스토리지 클러스터 시스템은 기존의 시스템과는 달리 고속의 신뢰성 있는 네트워크를 이용하여 테라 바이트 급에서 페타 바이트 급의 초대형 스토리지 공간이 제공되고, 네트워크에 연결되어 있는 호스트들이 이들 스토리지를 공유할 수 있다는 장점이 제공된다. 그러나, 기존의 유닉스(ufs: unix file system)나 리눅스(ext2)에서 제공되는 파일 시스템은 독립적인 호스트(single host)에서 호스트에 직접 연결된 기가 바이트 급의 소형의 저장 공간을 효율적으로 다룰 수 있도록 설계되어 있기 때문에 대형 저장공간을 공유하는 스토리지 클러스터 환

경에서는 성능과 기능의 부족으로 사용하기가 어렵다. 따라서 SAN을 기반으로 하는 스토리지 클러스터 환경에서 논리블록 관리자에서 제공되는 대용량의 가상 디바이스에 대용량의 파일 시스템을 생성하고, 생성된 파일 시스템에 저장된 파일들을 스토리지 클러스터를 구성하는 여러 호스트들이 공유할 수 있는 클러스터 파일시스템이 필요하다.

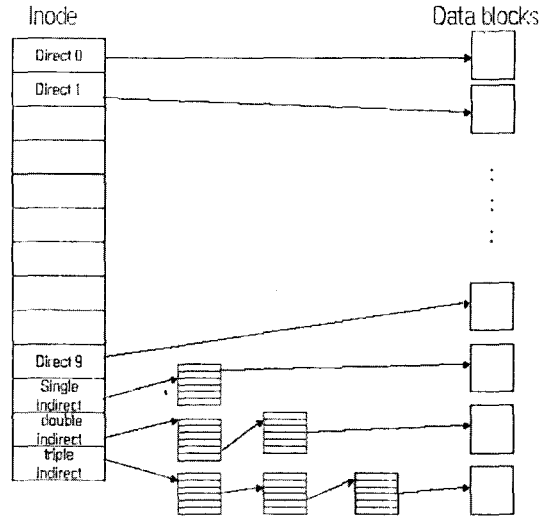
논리블록 관리자는 각각의 서버에 실제 저장장치가 있는 것처럼 가상의 저장장치를 설정해 데이터를 공유하는 가상의 디바이스 개념의 개념인 스토리지 가상화 기능을 제공하는 시스템 소프트웨어로 스토리지 가상화는 물리적인 경로를 논리적인 경로로 가상화 함으로써 SAN 시스템이 이기종 서버 시스템간의 스토리지 공유, 데이터 공유, 성능 최적화, SOD(Storage On Demand) 등의 다양한 관리기능을 수행할 수 있도록 지원한다.

SAN에 연결된 각 서버는 볼륨 관리자에서 제공되는 가상화 기능을 이용하여 물리적으로 분할되어 있는 각 디스크의 파티션을 묶어서 하나의 커다란 저장공간을 제공 받는다. 제공된 대용량의 볼륨을 호스트에서 사용하기 위해서는 mkfs를 이용하여 파일 시스템을 생성하고 생성된 파일시스템을 마운트하여 시스템에서 사용해야 한다. 파일시스템은 대용량 파일시스템 기능, 전역 파일 공유 기능, 빠른 회복기능을 제공해야 한다.

1) 대용량 파일 시스템

기존의 파일 시스템의 구조는 GB급의 파일을 위해서 개발되었기 때문에 수십 테라 바이트 이상의 파일 시스템을 지원할 수 없다. 이를 위해서는 64비트를 지원하는 파일 시스템을 개발하고, 파일 시스템을 구성하는 메타데이터에 대한 수정을 해야 한다.

- 기존의 시스템에서는 파일에 대한 연산을 위한 시스템 콜(system call)에서 32비트의 offset을 사용한다. 32비트의 경우 2GB 이상의 파일에 대한 연산을 수행할 수 없으므로 64비



<그림 4> Ext2의 아이노드 구조

트를 지원할 수 있도록 수정을 해야 한다.

- 파일시스템의 메타데이터는 슈퍼블록, 아이노드, 디렉토리 구조, 비트맵 등과 같이 파일시스템을 관리하는 데이터이다. <그림 4>는 Ext2의 아이노드 구조이다. 아이노드는 실제 데이터 블록에 대한 포인터를 저장하고 있는 10개의 직접 포인터와, 데이터 블록에 대한 포인터를 가리키는 간접, 이중간접, 삼중 간접 포인터가 1개씩 존재한다. 따라서 데이터 블록의 크기가 1KB로 가정할 때 생성할 수 있는 파일의 최대 크기는 약 16GB가 된다(32비트를 사용하는 시스템에서는 최대 2GB).

* 데이터 블록의 크기: 1KB인 경우	
- Direct : 10 * 1KB	= 10KB
- Indirect : 1 * 256 * 1KB	= 256KB
- Double : 1 * 256 * 256 * 1KB	= 65,536KB
- Triple : 1 * 256 * 256 * 256 * 1KB	= 16GB
■ Max. 파일 크기	: 16 GB.

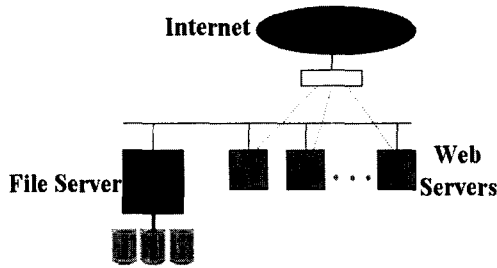
스토리지 클러스터 환경에서는 테라 바이트 급에서 페타바이트 급에 이르는 저장 공간이 제공될 수 있기 때문에 대용량의 파일을 생성하고, 파일에 대한 효과적인 연산 기능을 제공하는 대용량 파일 시스템이 요구된다.

2) 파일 공유 기능

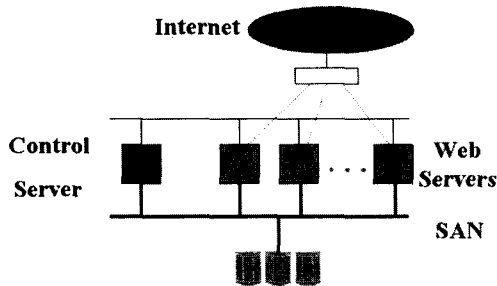
스토리지 클러스터 환경에서의 파일 공유 기능은 SAN과 같은 네트워크 스토리지 시스템에 연결된 서버들이 네트워크 스토리지에 저장되어 있는 파일들을 공유할 수 있는 기능을 말한다.

〈그림 5〉는 파일 공유가 지원되지 않는 환경에서 시스템의 구성 예를 보여주고 있다. 인터넷을 통해서 접속하는 사용자는 웹 서버에 접속하여 원하는 서비스를 요청하면, 웹 서버는 파일서버에게 해당 정보에 대한 처리를 요구하고 파일서버에 결과를 전송 받아 사용자에게 전송한다. 이러한 시스템에서는 사용자가 증가하는 경우 각각의 웹 서버에서 동시에 파일서버에게 정보에 대한 처리를 요청을 함으로써 웹 서버는 여유가 있는 반면, 파일시스템에 부하가 집중되어 전체 시스템의 성능이 저하된다. 특히 최근에는 멀티미디어와 대용량의 데이터에 대한 서비스를 요구하는 환경에서는 큰 영향을 받게 되며, 시스템의 성능의 향상을 위해 웹 서버 시스템의 확장도 파일서버의 성능에 의해서 제한을 받게 된다.

〈그림 6〉은 파일 공유 기능이 지원되는 시스템



〈그림 5〉 파일 공유가 지원되지 않는 시스템 예



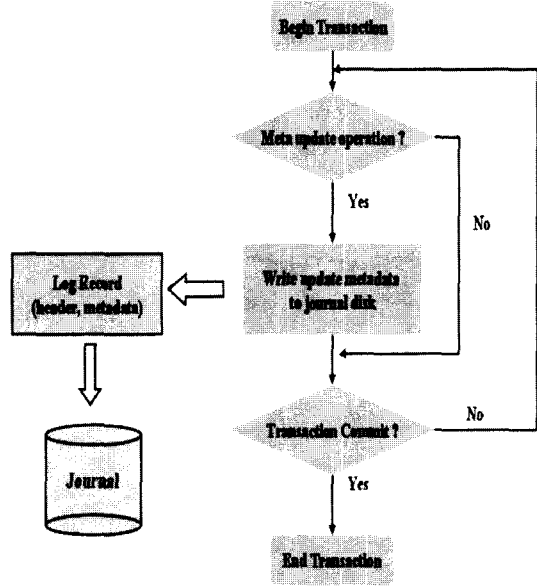
〈그림 6〉 파일 공유가 지원되는 구성 예

의 구성의 예를 보여 주고 있다. 데이터가 저장되어 있는 디스크를 SAN과 같은 네트워크 기반의 스토리지로 구축을 하게 되면 각 웹 서버는 사용자가 요구한 정보를 파일서버 없이도 직접 디스크로부터 읽어서 서비스할 수 있으므로 전체 시스템의 성능이 향상되며, 시스템을 쉽게 확장할 수 있다.

3) 빠른 회복 기능

테라 바이트 급 이상의 스토리지에서 fsck 이용하여 파일시스템의 회복을 수행하게 되면, 너무 많은 시간이 소요될 뿐만 아니라 회복을 수행하는 동안에는 시스템 전체가 정지해야 함으로 대용량의 네트워크 파일 시스템에서는 사용할 수 없다.

대용량 파일 시스템에서는 fsck를 사용하지 않고 복구시간을 줄이고, 유용성을 증가시키는 저널링(journaling) 기법을 사용한다. 저널링 기법은 〈그림 7〉과 같이 트랜잭션과 로그를 이용하여 파일시스템에 대한 연산이 발생하는 경우 메타 데이터에 대한 정보만을 로깅하여 이상의 발생한 경우에 로깅된 메타데이터를 이용하여 회복



〈그림 7〉 저널링 기법의 흐름도

을 진행하는 기법이다.

리눅스를 기반으로 개발된 저널링 파일 시스템은 Ext3, JFS(Jouraling File System), XFS, GFS 등이 있으며, 국내에서는 한국전자통신연구원의 SANtopia와 매크로임팩트의 SANnique가 있다.

IV. 백업 기술

백업되는 데이터의 범위를 기준으로 전체 백업과 증분 백업으로 크게 구분한다. 증분 백업은 다시 인크리멘탈 백업과 디퍼렌셜 백업으로 나뉘어 있는데, 각각에 대한 설명은 아래와 같다.

1. 풀 백업(Full Backup)

변경 사항 유무에 관계없이 전체 데이터를 백업하는 방법이다. 백업에 필요한 시간과 용량이 가장 많이 소요되지만 가장 간단하게 복구할 수 있다는 장점이 있다.

2. 인크리멘탈 백업(Incremental Backup)

가장 최근의 풀 백업, 또는 인크리멘탈 백업이 수행된 후 변경된 사항을 선택적으로 백업하는 방식이다. 풀 백업과 비교했을 때 백업에 필요한 데이터 량이 적고 백업에 소요되는 시간이 짧다는 장점이 있다. 하지만 인크리멘탈 백업은 전체 백업에 종속적이라는 단점이 있다. 따라서 풀 백업이 없는 상태에서는 효용 가치가 거의 없다. 또 인크리멘탈 백업이 많을수록 복구에 오랜 시간이 소요된다. 이러한 이유로 최소한 일 주일에 1회 이상 풀 백업을 수행하는 것이 권장된다.

3. 디퍼렌셜 백업(Differential Backup)

디퍼렌셜 백업은 증분 백업 형태의 한 가지 방법이라는 점에서는 인크리멘탈 백업과 같다. 단지, 최근에 수행된 디퍼렌셜 백업과 무관하게 풀 백업 이후의 모든 변경사항을 백업한다는 점이 다를 뿐이다. 디퍼렌셜 백업은 인크리멘탈 방식과 비교했을 때 백업 데이터의 크기가 커진다는

단점이 있지만, 복구 시 풀 백업본과 디퍼렌셜 백업본 각각 하나씩만을 필요로 하므로 더 빠른 시간 안에 복구가 가능하다는 장점을 제공한다.

4. 블록 단위 증분 백업

(Block-level Incremental Backup)

일반적인 증분 백업은 파일에 변경이 발생하면 경우 파일 전체를 백업하는 방식인 반면, 블록 단위 증분 백업에서는 업데이트된 파일 안에서도 실제로 변경된 블록 또는 페이지만을 복제함으로써 더 적은 데이터를 더 짧은 시간에 백업할 수 있게 한다. 이 기술은 일부 데이터베이스 백업 솔루션과 PC 백업 솔루션에 제한적으로 응용되고 있다.

V. 결 론

최근 인터넷의 발전과 디지털화 시대가 도래하면서 스토리지 시장은 급격히 성장하였다. 이제 IT 시장은 서버 중심에서 스토리지 중심으로 이 전되었으며, 스토리지 관련된 각종 제품들이 개발되고 있다.

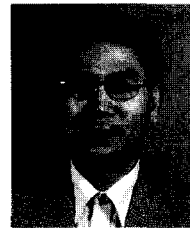
본 고에서는 스토리지 시장의 핵심으로 떠오르고 있는 네트워크 스토리지의 전반에 대한 소개를 하였다. 네트워크 스토리지의 하드웨어 기술인 NAS, SAN, 그리고 iSCSI를 중심으로 한 IP 스토리지에 대하여 설명하였고, 이를 네트워크 스토리지를 위한 시스템 소프트웨어 기술인 가상화를 위한 볼륨관리 기술과 대용량 클러스터 파일 시스템 기술에 대하여 소개하였다.

참 고 문 헌

- [1] A. F. Benner, "Fibre Channel : Gigabit Communications and I/O for Computer Network", McGraw-Hill, 1996.
- [2] C. A. Thekkath, T. Mann and E. K. Lee, "Frangipani : A scalable Distributed File System", DEC, 1996.

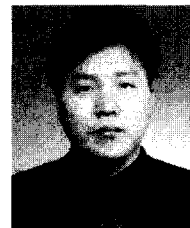
- [3] C. S. Kim, G. B. Kim and B. J. Shin, "Volume Management in SAN Environment", IEEE ICPADS2001, June 2001.
- [4] C. S. Park, B. J. Shin, S. I. Song and J. S. Yoo, "Parallel High-Dimensional Index Structure on the SAN", The 3rd ICACT, Feb. 2001.
- [5] E. K. Lee and C. A. Thekkath, "Petal : Distributed Virtual Disks", 13th Symp. On Operating System Principles, Oct. 1996.
- [6] G. B. Kim, C. S. Kim, and B. J. Shin, "A-64 bit, Scalable File System for Storage Area Networks", 5th WSES, Greece, 2001. 7.
- [7] IBM, "A Practical Guide to Tivoli SANergy", IBM Readbook, June 2001.
- [8] K. W. et al., "A 64-bit, Shared Disk File System for Linux", In The 7th NASA Goddard Conference on Mass Storage System and Technologies in cooperation with the 16th IEEE Symposium on Mass Storage Systems, pp. 22-41, San Diego, USA, March 1999.
- [9] M. Baker and R. Buyya, "Cluster Computing at a Glance", High Performance Cluster Computing Vol 1, Prentice Hall, 1999.
- [10] M. T. O'Keefe, "Standard File Systems and Fibre Channel", 6th Conference on Mass Storage System and Technologies in cooperation with the Fifteenth IEEE Symposium on Mass Storage Systems, pp.1-16, Colleague Park, Maryland, Mar. 1998
- [11] M. J. Folk et al., "File Structures", Addison-Wesley, Mar. 1998.
- [12] R. H. Katz, "High-Performance Network and Channel Based Storage", Proceedings of IEEE, Vol.80, No.8, pp.1238-1261, 1992.
- [13] S. R. Soltis, "The Design and Implementation of a Distributed File System based on Shared Network Storage", Ph. D. Thesis, University of Minnesota, 1997.
- [14] T. E. Anderson et al, "Serverless Network File Systems", 15th ACM Symp. On Operating Systems Principles, Dec. 1995.
- [15] U. Vahalia, "Unix Internals : The New Frontiers", Prentice-Hall, 1996.
- [16] Y. K. Lee, S. W. Kim, G. B. Kim and B. J. Shin, "Metadata Management of the SANtopia File System", IEEE ICPADS2001, June 2001.
- [17] 김경배, 김영호, 김창수, 신범주, "SAN을 위한 전역 파일 시스템의 개발", 정보과학회지 제19권 3호, 2001. 3.

저자 소개



신범주

1983 경북대 전자공학과(학사), 1991 경북대 컴퓨터공학과(석사), 1998 경북대 컴퓨터공학과(박사), 1987~2002 : 한국전자통신연구원 근무(책임연구원, 자료저장시스템 S/W연구팀장), 2002~현재 : 밀양대학교 컴퓨터공학부 교수, <주관심 분야 : 분산객체시스템, 이동 컴퓨팅 및 네트워크 저장 시스템 등>



김경배

1992 인하대학교 전자계산학과(학사), 1994 인하대학교 전자계산학과(석사), 2000 인하대학교 전자계산학과(박사), 2000~현재 : 한국전자통신연구원 컴퓨터 시스템 연구부 선임연구원, <주관심 분야 : 자료저장 시스템, SAN, 리눅스 시스템, 이동컴퓨팅>