

제 2절 데이터 저장기술

데이터베이스 관리시스템은 기본적으로 대용량의 데이터를 효율적으로 저장하고 관리할 뿐만 아니라 여러 사용자가 동시에 접근할 수 있는 환경을 제공하고 시스템의 고장에 대비하여 데이터베이스의 상태를 일관성 있게 유지하는 회복 기능을 제공한다.

데이터베이스 관리시스템에 대한 사용자의 만족도는 데이터의 저장기술과 관련된 시스템의 성능에 크게 좌우된다. 시스템의 성능은 데이터베이스 내의 데이터를 표현하기 위한 데이터 구조의 효율성과 데이터에 대한 연산을 얼마나 효율적으로 수행할 수 있는 지 여부에 달려있다. 자료저장 기술은 데이터베이스 시스템의 하부에서 데이터베이스 관리시스템의 성능을 좌우하는 기술이다.

본 절에서는 먼저 기존의 Oracle, DB2, Informix 등의 데이터베이스 관리 시스템에서 자료저장에서 사용되는 핵심 기술인 인덱스, 동시성 제어, 회복기법에 대하여 설명하고, 최근에 자료 저장 기술과 관련하여 비정형 대용량 데이터의 저장 기법과 백업 관련 기술, 그리고 저장시스템의 하부 인프라로 등장한 네트워크 자료저장 시스템의 기술과 그에 관련 된 최신기술인 NAS, SAN, iSCSI, DAFS 등에 대하여 다룬다.

1. 데이터 저장 핵심 기술

가. 인덱스

인덱스는 인덱스를 구성하는 키의 순서가 유지되는지의 여부에 따라 크게 두 종류로 나눌 수 있는데, 키가 일정한 순서로 유지되는 트리 계열과 키의 순서와 관계없이 무작위로 유지되는 해쉬 계열로 구분할 수 있다. 트리 계열 인덱스는 일정한 범위가 주어지는 연산을 처리할 때 유용하게 사용될 수 있으며, 해쉬 계열 인덱스는 특정한 키에 의한 빠른 데이터 접근을 제공한다. 트리 계열 인덱스로는 B+ 트리, T 트리 등이 있으며, 해쉬 계열 인덱스로는 체인 버킷 해싱(CBH, Chained Bucket Hashing), 확장 해싱(EH, Extensible Hashing), 선형 해싱(LH, Linear Hashing), 수정된 선형 해싱(modified Linear Hashing), 다중 디렉토리 해싱(Multi-directory Hashing) 및 확장된 체인 버킷 해싱(ECBH, Extensible Chained Bucket Hashing) 등이 있다. 최근에는 지리정

보시스템과 멀티미디어 데이터베이스 등에서 비정형 데이터의 검색을 위해 R-트리 계열의 인덱스 기법이 사용된다.

(1) 디스크 기반의 데이터베이스 시스템을 위한 인덱스 기법

기존의 Oracle, DB2, Informix, Sybase 등의 디스크를 기반으로 하는 상용 데이터베이스 관리시스템에서 기본적으로 사용되는 인덱스 기법은 B-트리, B+ 트리, B* 트리 계열의 인덱스와 확장 해싱과 같은 해쉬 계열의 인덱스가 대표적이다.

B 트리의 확장 및 변형된 형태로는 B+ 트리, B* 트리 등이 있으며, 특히 B+ 트리는 디스크를 기반으로 하는 데이터베이스의 인덱스를 생성할 때 가장 많이 사용되는 기법이다. B+ 트리는 B 트리 형태인 인덱스 부분과 실제 데이터들이 유지되는 말단 부분으로 구성되는데, 인덱스인 내부 노드의 데이터는 단지 빠르게 단말 노드를 찾아가는 길잡이 정도의 역할만 수행한다. 모든 데이터를 단말 노드에 유지하고 단말 노드끼리 연결 리스트를 구성함으로써 범위가 주어진 연산을 수행할 때 데이터에 대한 빠른 순차적 접근을 제공한다.

확장해싱(EH) 기법은 데이터의 양이 변화함에 따라 해쉬 테이블의 크기를 확장할 수 있게 함으로써 동적인 환경에 적합하도록 고안된 해싱 기법으로 빠른 데이터의 검색을 위해 많이 사용된다.

(2) 주기억장치 상주 기반의 데이터베이스 시스템을 위한 인덱스 기법

Altibase나 Timesten과 같이 주기억장치 상주 기반의 상용 데이터베이스 시스템에서는 트리 계열의 인덱스로 B-트리 계열의 인덱스와 함께 T 트리를 사용하고 있으며, 해싱 기능으로는 확장 해싱과 체인 버킷을 결합해서 만든 확장된 체인 버킷 해싱을 지원하고 있다.

T 트리는 AVL 트리와 B 트리의 장점을 이용하여 주기억장치 상주 데이터베이스를 접근하는 데 적합하도록 고안된 대표적인 트리 계열의

인덱스기법으로 한 노드에 많은 수의 데이터를 갖는 이진 탐색 트리로 구성되어 빠른 데이터 접근을 제공하며 메모리의 사용상에도 장점을 갖는다. T 트리의 노드에 포함되는 데이터는 실제 키 값이 아닌 키를 포함하는 레코드의 메모리 주소를 나타낸다.

키를 포함하는 레코드의 메모리 주소를 데이터로 사용함으로써 키 값을 구하는 연산으로 인한 부하가 전체 인덱스의 처리 속도에 거의 영향을 미치지 않는 상태에서 고정 길이 키에 대해서 뿐만 아니라 가변 길이 키를 처리할 때 발생하는 메모리 사용의 낭비 문제도 해결할 수 있다.

확장된 체인 버킷 해싱(ECBH)은 Kairos에 구현된 주기억장치 상주용 인덱스로서 확장 해싱과 체인 버킷을 결합해서 만든 해쉬 인덱스이다. ECBH 인덱스는 EH 테이블, CBH 테이블, RID 풀(pool)의 3 부분으로 구성되며, EH에서의 단말 페이지를 CBH의 해쉬 테이블과 RID 리스트로 대체한 형태를 갖는다.

(3) 멀티미디어 데이터베이스 시스템을 위한 인덱스 기법

멀티미디어 데이터는 특별한 형식이 없는 비정형의 구조를 갖는다. 즉 전통적인 관계형 데이터베이스에서 다루는 레코드 형태와 같은 일정한 구조를 갖지 않고 긴 문장이나 연속된 비트 스트림 형태를 갖는다. 이러한 비정형 데이터를 효율적으로 저장하고 이에 대한 내용 기반 검색을 지원할 수 있는 저장 및 검색 기법이 필요하다.

공간 데이터 액세스를 위한 인덱스 구조로서 R 트리, R* 트리, MLGF(Multi-Level Grid File) 등의 구조가 사용되고 있다. R-tree는 B-tree로부터 유도된 계층적 자료구조로 트리의 각 노드는 자신의 자식 노드를 포함하는 가장 작은 d-차원 사각형으로 구성된다. 단말 노드는 데이터베이스에 있는 자식을 포함하는 대신 실제 기하학적 객체에 대한 포인터를 포함한다. 객체들은 그들이 포함된 최소의 정렬된 사각형에 의해 표현된다. 가끔 노드는 디스크 페이지와 관

런시켜 트리를 정의하는 파라미터가 공간 질의 동안 적은 수의 노드가 방문되도록 선택된다. 서로 다른 노드에 해당하는 사각형들은 중복될 수 있다.

R-tree는 B-tree와 유사하게 높이가 균형화된 트리이다. 인덱스는 완전히 동적이기 때문에 주기적인 재구성은 필요 없으며, R-트리의 단말 노드는 인덱스 레코드 개체들을 포함한다. R 트리는 k 차원의 공간 객체를 자르거나 변환하지 않고 겹치는 k 차원의 사각형 영역 안에 객체가 완전히 포함된다.

R* 트리는 R 트리의 색인 구조와 같으며 중간 노드의 영역 사각형들이 겹치는 것을 허용한다. R 트리가 색인 구조를 최적화하기 위하여 한가지 기준만을 적용하여 영역 사각형의 면적을 최소화하는 것에 비해 R* 트리는 네 가지의 최적화 기준을 적용한다. R* 트리의 삽입 알고리즘은 R 트리의 삽입 알고리즘이 영역의 면적을 고려하는 데에 비하여 영역의 면적, 둘레, 겹치는 영역의 크기를 모두 고려하여 공간 객체를 삽입할 최적의 단말 노드를 선택한다.

나. 동시성 제어 기법

(1) 기존의 동시성 제어 기법

여러 트랜잭션이 동시에 수행되는 경우 데이터베이스의 일관성을 해치는 경우가 생길 수 있다. 이를 방지하기 위해서는 수행의 동시성을 제어해야 한다. 수행의 동시성을 제어하기 위하여 잠금 기법을 사용한다.

Oracle, DB2, Informix 등의 대부분의 상용 데이터베이스 시스템에서 사용하는 동시성 제어 기법은 데이터의 로크(lock)를 이용하는 두 단계 잠금(2 phase locking) 기법을 기반으로 하고 있다. 2단계 로킹 규약은 직렬 가능성을 보장할 수 있는 규약으로 가장 많이 알려진 것으로 모든 트랜잭션들이 잠금과 잠금 해제 연산을 트랜잭션은 잠금만 수행할 수 있고 잠금 해제는 수행할 수 없는 확장단계와 트랜잭션이 잠금 해제만 수행할 수 있고 잠금은 수행할 수 없는 축소 단계로 구분된다. 그러나, 2단계 잠금 규약은 직

렬 가능성을 보장하지만 반면에 교착 상태 문제를 내포하고 있어 이에 대한 해결 방법이 필요하다.

2단계 잠금 규약을 기반으로 Oracle과 같은 상용 제품에서는 여러 사용자들의 잠금의 기다림 시간을 최소화하면서 동시에 소요되는 로크의 수를 최소화하기 위하여 고안된 다단위 잠금(Multi-granularity locking) 기법이 사용된다. 기존의 데이터베이스 시스템에서 다단위 잠금 기법을 위해 선택되는 단위들은 필드, 레코드, 블록, 데이터베이스 등의 일반적이다. 단위를 너무 좁게 설정하면 작업의 동시성은 향상되나 잠금 처리에 소요되는 비용이 많으며, 단위를 너무 넓게 설정하면 잠금 처리에 소요되는 비용은 적게 되나 작업의 동시성이 떨어지게 되므로 응용 분야의 특성에 맞는 단위를 설정하는 것이 필요하다.

(2) 분산 잠금 관리

인터넷 웹서버와 같이 다수의 사용자가 이용하는 데이터베이스 시스템에서는 고성능의 시스템이 요구된다. 이러한 고성능의 요구를 만족하기 위해서 여러 대의 시스템이 데이터베이스 서비스를 제공하는 병렬 서버와 클러스터 서버가 등장하게 되었고 이러한 병렬서버와 클러스터 서버에서는 분산 서버간의 잠금의 관리를 위해서 분산 잠금 관리자(Distributed Lock Manager, DLM)를 이용한다. Oracle의 병렬서버에서는 DLM을 이용하여 병렬 서버가 모든 플랫폼에 걸쳐 다양한 기능을 제공할 수 있게 되어 있으며, 서버의 성능 향상을 위해 여러 특징이 추가되었다. DLM은 데이터 잠금 정보가 노드를 통해 분산되는 것을 막기 위해 잠금 정보를 캐시에 저장하고, 충돌 블록에 대한 액세스를 통제할 수 있는 알고리즘이 제공되어 시스템 부하의 균형을 이룰 수 있다.

분산 잠금 관리자의 중요 기능으로는 병렬 서버의 한 개 노드에 장애가 발생했을 때, 장애가 발생한 노드에서 수행되던 사용자 접속이 정상인 다른 노드의 세션으로 옮겨지며 접속이 자동

으로 재확립되는 장애 복구(fail-over)기능이 있으며, 애플리케이션 장애 복구는 시스템의 가용성을 높일 뿐 아니라 수동적인 부하 조절이나 강제적인 시스템의 정지에도 효과적으로 사용될 수 있다.

다. 회복 기법

기존의 상용 데이터베이스 시스템에서 대부분이 로그 기반의 회복기법을 사용하고 있으며, 트랜잭션의 수행시간의 지연을 최소화하기 위해 로그를 위한 디스크 입출력 횟수의 최소화와 고장이 발생한 경우 신속한 시스템 회복을 목표로 하고 있다. 이를 위해 데이터베이스 시스템에서는 검사점(Checkpoint) 기법을 이용하여 로그의 양을 줄임으로서 회복의 시간을 빠르게 하고 있다. 대용량 멀티미디어 데이터베이스 시스템 등에서는 자료의 효율적인 회복을 위하여 기본정보의 변화만을 수행기록으로 남기고, 자료의 내용은 그림자 페이징 기법(Shadow-paging) 기법을 이용하여 이전 자료의 내용을 디스크 내에 남아있게 한 후에 회복 시 그를 이용하고, 회복의 필요가 없을 경우 검사점이 수행되는 시기에 그를 삭제하는 방식 등의 별도의 처리과정이 행해지게 된다.

그림자 페이징 기법은 기본적으로 로그를 이용하지 않는다. 데이터베이스는 페이지라는 일정한 크기의 블록으로 나뉘어 디스크에 저장된다. 페이지라는 용어는 운영체제의 메모리 관리 기법에서 사용된 용어이다. 그림자 페이징 기법은 트랜잭션이 실행되는 동안 현재 페이지 테이블과 그림자 페이지 테이블의 두 페이지 테이블을 이용한다. 트랜잭션이 시작될 때에는 이 두 페이지 테이블은 동일하다. 트랜잭션이 실행되는 동안 현재 페이지 테이블만 사용하고 그림자 페이지 테이블은 사용하지 않는다. 트랜잭션이 쓰기 연산을 수행할 때 현재 페이지 테이블만 변경한다. 판독과 기록 연산을 위해 디스크 상에 존재하는 데이터베이스 페이지 블록을 찾을 때에도 현재 페이지 테이블만 사용한다. 따라서, 그림자 페이지는 전혀 변함없이 트랜잭션 실행 이전의

상태를 유지한다.

2. 최신 데이터 저장기술 및 현황

가. 비정형 대용량 데이터의 저장

멀티미디어 데이터는 기존의 문자와 수치 데이터와는 비교가 안 될 정도로 큰 용량의 기억장소를 요구한다. 따라서 멀티미디어 데이터베이스는 데이터를 저장할 때 압축 및 복원 기법을 필수적으로 사용해야 하며, CD-ROM과 같은 대용량의 저장 장치와 다양한 저장 장치의 관리가 필요하다. 데이터의 압축 및 복원 기법으로서 널리 사용되고 있는 기법은 이미지 데이터를 압축하기 위한 JPEG기법과 비디오 데이터를 압축하기 위한 MPEG기법으로, 국제적으로 널리 사용되고 있으며 많은 연구가 진행 중에 있다.

데이터베이스 내에 멀티미디어 데이터를 저장하고 관리하는 방법으로 우선 레코드 형태의 정형화된 데이터는 관계형 데이터베이스에 저장하고 이미지, 그래픽스 등 비정형 데이터는 파일에 저장하는 방법과 이미지, 오디오, 텍스트 등 크기가 크고 가변적인 데이터를 긴 자료 항목으로 저장할 수 있는 저장 기능을 제공하는 방법이 있다. 이러한 시스템들은 기존의 관계형 데이터 모델로 멀티미디어 응용을 지원하기 어려운 점들을 해소하기 위하여 BLOB(Binary Large Object)과 같은 기능을 가진 확장된 모델을 제공한다.

현재 관계형 DBMS 회사들이 자신의 제품에 멀티미디어 기능을 갖도록 확장한 제품들을 여러 개 발표하였는데 그 예로는 Informix-OnLine, Oracle 9, Sybase 등이 있다. 국내에서는 관계형 DBMS 자체에 대한 상품화가 제대로 안된 실정에 있지만, 삼성전자에서 관계형 DBMS인 CODA를 개발하면서 BLOB을 이용하여 멀티미디어 데이터베이스를 지원하는 시제품을 1994년에 개발한 바 있으며, ETRI에서는 '바다' DBMS에도 BLOB 기능을 추가하였다.

특히, Oracle에서는 이미지, 사운드, 비디오,

텍스트와 같은 비정형 데이터를 저장하기 위한 기반 타입으로 CLOB(Character LOB), BLOB(Binary LOB), 그리고 BFILES(데이터베이스 외부에 저장되는 LOB) 등이 있다. BLOB, CLOB 타입의 데이터는 데이터베이스 내의 지정된 테이블 공간에 저장된다. 반면 BFILE 데이터는 데이터베이스 밖의 운영체제가 관리하는 파일 시스템 내의 특정 디렉토리에 저장할 수 있다.

또한, Oracle9i R2에서는 Oracle Internet File System을 제공하여 인터넷을 위주로 한 기업의 데이터를 하나의 저장소에 병합할 수 있도록 해준다. Oracle Internet File System에서는 저장소에 저장된 문서를 사용자들이 Windows, Web, FTP 등의 인터페이스를 이용하여 접근할 수 있도록 파일과 폴더의 형태로 제공함으로써, 데이터베이스 관리시스템의 파일 관리자의 기능을 운영체제의 파일 시스템의 기능을 포함하도록 확장하였다.

나. 백업 기술

최근 들어 백업은 컴퓨터산업에서 정보의 중요성이 인식되면서 가장 각광을 받는 분야로 부각되고 있으며, 그 시장 규모도 크게 확장이 되었다. 기업에 전산과 관련된 비용 중에서 30~50%가 기업에서 발생하는 정보의 백업에 소요되는 비용으로 조사되고 있으며, 특히, 미국의 9.11 테러의 발생 이후 백업기술은 재해복구 기술과 결합하여 데이터베이스뿐만 아니라 볼륨, 파일, 그리고 응용프로그램 등에 발생하는 엄청난 양의 데이터를 효과적으로 백업하고 복구하는 기능을 수행해야 한다.

(1) 백업의 유형

백업의 유형으로는 모든 데이터베이스나 파일 시스템의 내용을 백업 받아 두는 방법으로 내용의 변화의 유무에 관계없이 복사를 하는 완전(full) 백업, 선택된 파일이나 폴더들의 변경된 부분만을 백업하여 복원을 할 때에 증분 백업을 한 모든 백업 데이터파일 또는 모든 백업 데이

터를 이용하여 복원하는 증분(Incremental) 백업, 변경된 파일의 전부를 저장하여 복원을 할 때에 맨 마지막날 차등 백업을 한 백업 데이터 파일 또는 백업 테이프만 있으면 복원을 할 수 있는 차등(differential) 백업, 그리고 백업 간격별로 단계를 정한 후 단계별로 다단계로 점진적인 백업을 수행하는 다중레벨 증분(Multilevel incremental) 백업이 있다.

Oracle에서는 베리타스 소프트웨어사의 Net-Bakcup과 결합하여 제공하는 백업 기능 중의 하나로 이전의 특정 시점을 기준으로 현재 백업하고자 하는 시점까지의 작업 내용 중에서 단지 변화된 블록의 내용만을 백업하므로 백업되는 데이터의 크기를 상당히 줄일 수 있다. 이로 인해 데이터 파일의 백업 시간을 대폭 줄일 수 있으며, 백업 간격별로 단계를 정한 후 단계별로 다단계로 점진적인 백업을 수행할 수 있다.

복구해야 할 경우에는 데이터베이스의 상태를 분석해 수행에 필요한 연산을 결정한다. 이러한 조작을 자동으로 수행하므로 관리자가 작업에 들이는 수고를 덜어주고 복구 작업 시의 에러 발생 가능성을 최소한으로 줄여준다.

(2) 백업의 기법

(가) 서버리스 백업

SAN 백업 구성 시에 활용 가능한 방법으로 만일 백업서버가 동시에 웹 서버나 데이터베이스 서버로 활용되고 있다고 가정해 보자. 백업을 수행하면 백업서버의 CPU에 부하가 걸리게 된다. CPU 사용률이 적게는 30%에서 많게는 100%까지도 올라간다. 이렇게 되면 웹 서버나 데이터베이스 서버로의 성능을 내기란 여간 힘든 일이 아닐 수 없다. 그렇지만 서버리스 백업을 사용하는 경우에는 백업서버에서는 백업을 위한 명령만을 백업장치에 전달하고, 백업장치는 디스크 스토리지와 직접 백업 데이터를 전송 받게 되고, 백업서버는 지정한 시간에 한번씩 백업 이상 유무를 확인하는 작업만을 수행하게 된다.

CPU의 사용률은 10% 전후가 되고, 서버의 또 다른 기능을 원활하게 사용하게 될 수 있게

된다. 물론, 서버리스 백업은 ISV에서 제공하는 백업 어플리케이션의 옵션으로 구성되는 것이다.

(나) snapshot

온라인 백업은 일상의 업무 수행에 가장 영향을 주지 않기 때문에 24시간 가용성을 필요로 하는 데이터베이스에 알맞다. 그러나, DBMS는 백업 도중에 트랜잭션을 로깅 해야 하므로, 백업과 백업 후에 쌓인 트랜잭션을 실제 데이터에 적용해야 하기 때문에 성능이 약간 저하된다. 볼륨 백업 시간을 최소화 하기 위해서, 고객들은 현재 한 볼륨을 snapshot으로 상용하여 그 snapshot 볼륨으로 백업을 받는 기능을 사용하고 있다. 이때, 실제 볼륨은 신속하게 실제 운영에서 사용할 수 있게 된다.

(다) LAN-Free 백업

데이터베이스 시스템의 하부 인프라가 SAN과 같은 네트워크 기반의 시스템으로 구성이 된 경우는 파이버 채널로 구성된 별도의 SAN에 각 서버의 스토리지가 직접 연결되어 있기 때문에 백업을 위한 데이터의 이동을 파이버 채널을 이용하고 스토리지간의 데이터 트래픽은 기존의 LAN을 사용하지 않는다. 따라서 시스템의 성능은 급격히 향상되는데, 이것은 데이터와 통신 트래픽이 더 이상 제한된 표준 LAN의 대역폭에 대해 경쟁하지 않기 때문이다. SAN 환경에서는 스토리지의 리소스는 통합 관리되며, 다수의 서버에 의해 공유되어 사용될 수 있다. 또한 파이버 채널 스위치의 추가를 통해 SAN 솔루션은 쉽게 확장 가능하다.

SAN 환경에서 데이터가 지능적인 백업 어플리케이션을 통해 관리되도록 다양한 소프트웨어를 개발하고 있다. 기존의 백업 방식에 비해 LAN-Free 방식은 광채널 구성요소를 설치하는데 많은 초기 투자비용이 들어가는 단점을 가지고 있다. 그러나 스토리지의 통합 및 중앙 관리는 전체소요비용을 크게 줄일 수 있어 궁극적으로는 기업에게 더 많은 이점을 안겨준다.

3. 저장 인프라의 발전 및 기술

인터넷의 폭발적인 성장, 온라인 상에서의 정보유지 필요성, 의사결정 지원정보의 수집/추적에 대한 필요성, 서버 통합, 비즈니스 중심 어플리케이션으로의 PC 서버 움직임, 그리고 어플리케이션의 복잡성 증가 등 다양한 요인들로 인해 현재 기업 규모의 시스템에서는 대용량 데이터베이스 시스템의 구축, 데이터웨어하우스(Data Warehouse)의 구축, 그리고 ERP(Enterprise Resource Planning) 시스템 등의 도입으로 대용량 스토리지 시스템에 대한 요구가 계속 높아지고 있다. 이러한 높은 신뢰성과 성능, 장애 복구성(fault tolerance), 그리고 통합된 관리와 고속 백업이라는 요구에 대한 솔루션으로 등장한 인프라가 바로 SAN(Storage Area Network)과 NAS(Network Attached Storage)로 대변되는 네트워크 자료 저장 시스템이다. SAN과 NAS는 분산 네트워킹에서 주류가 되고 있으며 조만간 데이터베이스 시스템의 위한 스토리지 부착 및 공유에 대한 일반적인 방법이 될 것으로 전망되고 있다.

가. NAS

DAS는 서버와 외장형 스토리지 사이를 전용의 케이블로 직접 연결하는 것으로 기존의 서버와 스토리지를 직접 연결하는 전통적인 방식이다. 만약 서버가 UNIX나 Linux서버와 같은 오픈 시스템이면 SCSI 케이블을 이용하여 연결하는 방법이다. 이러한 DAS는 저장된 데이터에 대한 서비스를 스토리지에 접속된 서버가 처리해야 함으로 서버에 큰 부하가 발생한다.

NAS는 네트워크 접속용 스토리지로서 Ethernet과 같은 LAN 인터페이스를 통해 네트워크에 직접 연결되는 새로운 개념의 데이터 저장장치로 네트워크 상에서 서로 다른 기기종 플랫폼간의 효율적인 데이터 통합 사용과 기존의 서버-중심형 스토리지와 같이 일반용 서버의 상위구조나 그 복잡함 없이 오직 데이터-펌프(Pump Data : 스토리지 시스템 자체와 클라이

언트간에 신속하게 바로 데이터 전송이 이루어지는 것) 라는 데이터 처리의 목적만을 위해 최적화 된 데이터-중심형의 데이터 전용 스토리지 시스템이다.

기존의 서버-중심형 방식과는 달리 NAS는 데이터 저장장치가 서버에 종속되지 않고 독립적으로 네트워크에 직접 연결된다. 그 결과 사용하는 파일서버를 거치지 않고 필요한 데이터를 직접 NAS 저장장치를 통해 훨씬 빠른 속도로 접근할 수 있다. 즉, 클라이언트에서 요구된 데이터의 저장/전송에 따른 네트워크 로드(Load)는 서버로부터 분리되어 NAS에서 부담함으로써 상대적으로 네트워크 상에서 서버의 성능은 향상되는 효과를 가져온다.

NAS의 장점은 파일의 공유 기능이다. NAS의 파일서버가 NFS나 CIFS 프로토콜을 통해 여러 어플리케이션 서버들 사이에 파일을 공유할 수 있게 해준다. 현재 존재하는 파일 공유 솔루션으로는 NAS가 가장 안정적이라는 장점을 지닌다. 그러나, NAS는 데이터의 전송에 LAN을 사용하므로 LAN이 불안정하거나 과부하가 발생한 경우에 성능이 떨어진다는 단점을 지니고 있다.

나. SAN

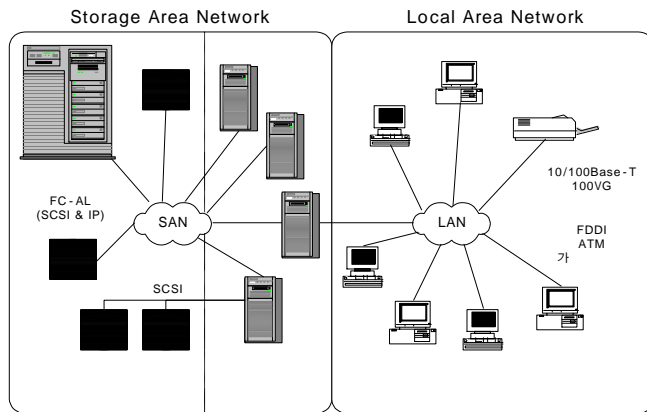
SAN은 다양한 업무 프로세스로 분산되어 있는 저장장치를 통합 운영하여 업무의 효율성 증

대, 인프라에 대한 중복투자 방지 및 이기종간 데이터 공유를 목적으로 만들어진 서버와 저장장치간의 네트워크를 의미한다. 개별적으로 연결되어 있는 저장 시스템을 하이버 채널을 이용하여 별도의 전용 네트워크에 연결하여 집중적으로 관리함으로써 데이터의 고속전송, 고 가용성, 확장성 및 공유 기능을 제공하기 위한 시스템이다. [그림 4-9-2]는 SAN의 구조를 보여 준다.

SAN은 데이터베이스 관리시스템 및 트랜잭션 프로세싱 시스템처럼 일반적으로 자체 대규모 데이터를 관리하는 고성능 어플리케이션에 적합하다. 모든 물리적 위치와 관련된 유연성과 특정 레코드나 작은 레코드 집합에 대한 빠른 접근에 초점을 맞추고 있다. 이러한 어플리케이션은 파일수준의 공통 접근을 요구하지 않고 블록 수준의 성능 및 제어를 요구한다. 데이터베이스 관리시스템에서는 블록 수준의 읽기와 쓰기를 요구하며 SAN은 이러한 요구를 충족시킬 수 있으며, 다양한 확장성을 제공함으로써 급증하는 데이터의 용량과 관리를 충족시킬 수 있다.

SAN은 다음과 같은 장점을 지닌다.

- 고속의 전송속도(200MB / Full- duplex 이상)
- 장거리 전송(100Km이상)
- 뛰어난 연결성과 확장성이 뛰어나다.(Over 16 Million)
- 용이한 Data 관리



[그림 4-9-2] SAN의 구조

- Heterogeneous Data Sharing(NT, UNIX)
- 용이한 통합 및 관리
- 저렴한 비용
- Multi-protocol Flexibility(SCSI, ESCON, IP)
- One active SAN another Standby(Fail Over)
- Hi-availability SAN(Clustering)

다. iSCSI

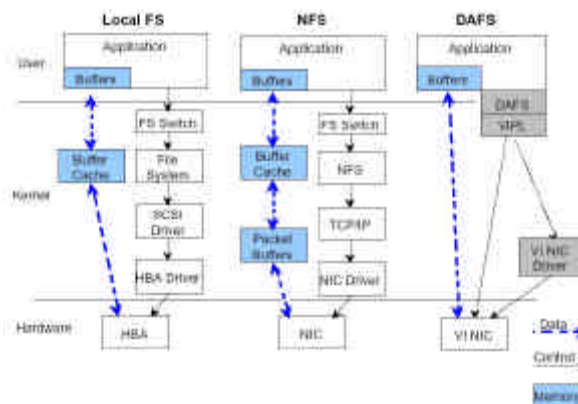
SAN은 DAS와 같이 서버에 직접적으로 스토리지를 연결하는 게 아니라 스위치를 통해서 전체 스토리지 영역을 구성해 DAS의 한계인 확장의 어려움을 해결했다. 그러나, SAN은 많은 장점에도 불구하고 시스템의 구축비용이 비싼 데다 벤더들간의 호환성 등이 문제로 지적됐다.

이러한 SAN의 문제점을 극복하고자 고안된 게 iSCSI라는 기술이다. 스토리지 시장의 새로운 화두로 떠오른 iSCSI는 'internet SCSI'의 약자로 스토리지 연결에 사용됐던 물리적 SCSI 연결 방식 대신 물리적으로는 이더넷을 사용하며 내부적으로는 SCSI 프로토콜을 사용하려는 기술 표준으로 IBM과 시스코가 2000년 2월 IETF(Internet Engineering Task Force)에 제출한 스토리지 네트워크 기술로써 현재 표준

안이 완성단계에 있다.

iSCSI의 핵심은 TCP/IP상에서 캡슐화 된 SCSI 정보를 이동시키는 것으로 로컬 혹은 원격지에서 서버와 스토리지를 연결하여 스토리지가 여러 지역에 분산돼 있고 IP 네트워크의 관리가 쉬운 고성능의 네트워크 스토리지 상에서 사용된다. iSCSI 이전에 SAN을 구축하기 위한 유일한 스토리지 네트워크 방법은 사용자가 각각의 분리된 SAN 인프라스트럭처에 투자해야 하는 파이버 채널뿐이었다. 왜냐하면 이것은 10M, 100Mbps 이더넷에 비해 빠른 기가비트의 속도를 제공하기 때문이다. 하지만 파이버 채널은 표준이 설정돼 있지 않아 벤더들 간에 호환이 어려웠으며 비용도 고가였다.

iSCSI에 관심을 갖는 이유는 스토리지 시장의 규모가 연간 4백억 달러에 달하며 지속적으로 성장하고 있기 때문이다. 즉 이러한 스토리지들의 연결이 가능하게 되면 사용자에게 더 높은 가치를 제공할 수 있다. 기존 스토리지는 DAS 형태로 전체 네트워크와 독립돼 있었다. 하지만 iSCSI의 도입으로 사용자는 원격에서도 데이터 베이스에 접근할 수 있다. 즉 고객에게 미치는 가장 큰 영향도 스토리지에 IP 네트워크를 이용해 접근, 자원의 사용이 가능하다는 것이다. 이것은 스토리지 네트워킹의 혁명이라 할 수 있다.



[그림 4-9-3] DAFS에서의 파일 접근 방법

자료 : <http://www.dafscollaborative.org>

라. DAFS

10GB 네트워크 기술의 등장은 스토리지 업계에도 상당한 파급력을 전해주고 있다. RAID 기술이 기존 스토리지의 물리적인 한계를 극복한 반면 iSCSI는 기존 SCSI 기술을 10GB 네트워크 상에서 사용할 수 있도록 개발이 진행 중이다. 하지만 본격적인 10GB 시대가 도래하면 iSCSI는 도태할 것이라는 전망도 조심스럽게 개진되고 있다. 이러한 상황에 DAFS(Direct Access File System)의 등장은 현재의 스토리지 네트워크를 한 단계 올릴 수 있을 것으로 기대되고 있다. [그림 4-9-3]은 DAFS에서의 파일 접근 방법을 보여 준다.

DAFS는 DAFS Collaborators라는 기관을 중심으로 IBM, HP, SUN, ORACLE 등 모두 85개 이상의 업체들이 참여하여 서로 역할을 분담하여 성능, 확장성, 안정성 개선이라는 목표를 가지고 노력해 왔다. 이들은 이미 노력의 결실을 일정 수준 이상까지 완성하여, 2001년 9월 DAFS Specification Version 1.0이 이미 완성되어 IETF에 제출되었으며 DAFS API Specification Version 1.0은 2001년 11월 완성되었다.

DAFS Collaborators는 크게 두 가지 분야에 초점을 두고 진행되고 있는데 그 중 하나는 DAFS라는 프로토콜 자체를 만드는 분야이며, 나머지 하나는 RDMA(Remote Direct Memory Access) 기술을 구현하고 그 기술을 DAFS에 접목시키고자 하는 분야이다.

현재 DAFS 프로토콜을 만드는 분야는 SNIA(Storage Networking Industry Association)의 한 부문으로써 DAFS Implementers Forum이라는 그룹을 중심으로 그 노력이 진행되고 있다. 초창기부터 이 포럼에 참가하여 초기 개발자로서 그 중심적인 역할을 수행하고 있는 업체들은 네트워크 어플라이언스를 중심으로 브로드밴드 스토리지, 에플렉스, 인텔, 그리고 베리타스 소프트웨어와 같은 회사들이 있다. 이 업체들이 초창기부터 구현하고자 했던 것은 dDAFS(driver

DAFS)였다. 그러나 진정한 DAFS의 실현은 uDAFS(user DAFS)의 완성이라고 보았을 때 이들만의 노력으로 이룩할 수 있는 것은 DAFS의 최종적인 목표는 될 수 없는 것이었다. 이러한 뜻에 동참하여 uDAFS의 최종 완성을 목표로 이들에 다시 합류하는 애플리케이션 업체들이 있는데 이들이 바로 오라클, 사이베이스, DB2 등 있다.