(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。Int. Cl.
G06F 15/16 (2006.01)

(11) 공개번호 10-2006-0080239
(43) 공개일자 2006년07월07일

(21) 출원번호 10-2006-7008391
(22) 출원일자 2006년04월28일
번역문 제출일자 2006년04월28일
(86) 국제출원번호 PCT/US2004/032122
국제출원일자 2004년09월30일
(87) 국제공개번호 WO 2005/033945
국제공개일자 2005년04월14일

(30) 우선권주장 10/911,987 2004년08월05일 미국(US)
60/507,329 2003년09월30일 미국(US)

(71) 출원인 세파톤 인코포레이티드
미합중국 매사추세츠주 01752 말보로우 니커슨 로드 - 세컨드 플로어 400

(72) 발명자 산도르피 미클로스
미국 매사추세츠주 02035 폭스보로 14 맥켄지 레인

(74) 대리인 하상구
하영욱

심사청구 : 없음

(54) 인스턴트 볼륨 복구를 지원하는 에뮬레이티드 스토리지시스템

요약

백업 스토리지 시스템에 있어서, 백업 데이터 세트에 대응하는 데이터 볼륨을 호스트 컴퓨터에 마운팅하는 장치 및 방법으로서, 상기 방법은 백업 스토리지 시스템에 저장된 가장 최근 백업된 버전의 하나 이상의 데이터 파일에 대응하는 데이터 파일을 하나 이상 포함하는 데이터 볼륨을 호스트 컴퓨터에 마운팅하는 단계, 및 상기 가장 최근 백업된 버전의 하나 이상의 데이터 파일을 보존하는 동안, 상기 백업 스토리지 시스템에 저장된 상기 가장 최근 백업된 버전의 하나 이상의 데이터 파일보다 더 최근의 제 2 버전의 하나 이상의 데이터 파일에 대응하는 데이터를 상기 백업 스토리지 시스템에 저장하는 단계를 포함한다.

대표도

도 1

색인어

인스턴트 볼륨 복구, 에뮬레이티드 스토리지 시스템, 백업 스토리지 시스템

명세서

기술분야

본 발명은 데이터 스토리지에 관한 것이고, 특히, 기존 풀 백업(full back up) 및 후속 증분적 백업(incremental back-up)을 사용하여 풀 백업의 동등함을 제공하기 위해 테이프 스토리지 시스템을 에뮬레이팅(emulating)하여 엔드-유저가 상기 백업으로부터 데이터를 복구할 수 있게 하는 장치 및 방법에 관한 것이다.

배경기술

대부분의 컴퓨터 시스템은 하나 이상의 호스트 컴퓨터와 이 호스트 컴퓨터에 의해 사용된 데이터를 저장하는 하나 이상의 데이터 스토리지 시스템을 포함한다. 이 호스트 컴퓨터와 스토리지 시스템은 일반적으로 파이버 채널 네트워크, 이더넷 네트워크, 또는 다른 형태의 통신 네트워크 등을 사용하여 함께 네트워킹(networking)된다. 파이버 채널(fibre channel)은 채널 기반 전송 방식의 속도와 네트워크 기반 전송 방식의 유연성을 조합하여 멀티플 이니시에이터(multiple initiator)가 네트워크를 통해 멀티플 타겟(multiple target)과 통신할 수 있게 하는 표준이고, 상기 이니시에이터와 상기 타겟은 네트워크에 연결된 임의의 장치일 수 있다. 파이버 채널은 일반적으로 광섬유 케이블 등의 빠른 전송 매체를 사용하여 구현됨으로써 대용량 데이터를 전송하는 스토리지 시스템 네트워크에 널리 선택되고 있다.

도1은 여러 호스트 컴퓨터와 백업 스토리지 시스템을 포함한 일반적인 네트워킹된 컴퓨팅 환경의 일례를 나타낸다. 하나 이상의 애플리케이션 서버(application server)(102)는 근거리 통신망(LAN)(103)을 통해 복수의 유저 컴퓨터(104)에 연결된다. 애플리케이션 서버(102) 및 유저 컴퓨터(104)는 모두 "호스트 컴퓨터"로 간주될 수 있다. 애플리케이션 서버(102)는 SAN(storage area network)(108)을 통해 하나 이상의 제 1 스토리지 장치(106)에 연결된다. 제 1 저장 장치(106)는 예컨대, EMC Corporation, IBM Corporation 등에서 이용될 수 있는 디스크 어레이일 수 있다. 대안으로 버스(도시안됨) 또는 기타 네트워크 링크는 애플리케이션 서버와 제 1 스토리지 시스템(106) 사이의 상호접속을 제공할 수 있다. 버스 및/또는 파이버 채널 네트워크 연결은 호스트 컴퓨터[예컨대, 애플리케이션 서버(102)]와 서장 시스템(106) 사이에서 전송된 패킷의 포맷을 지시하는 SCSI(Small Component System Interconnect) 프로토콜 등의 프로토콜을 사용하여 작동될 수 있다.

도1에 도시된 네트워킹된 컴퓨팅 환경은 예컨대, 대형 금융기관 또는 대기업에 의해 사용될 수 있는 대형 시스템의 전형적인 예이다. 대부분의 네트워킹된 컴퓨팅 환경이 도1에 도시된 요소를 모두 포함할 필요는 없다. 예컨대, 작은 네트워킹된 컴퓨팅 환경은 스토리지 시스템에 직접 또는 LAN을 통해 연결된 호스트 컴퓨터를 간단히 포함할 수 있다. 또한, 도1에는 유저 컴퓨터(104), 애플리케이션 서버(102) 및 매체 서버(114)가 따로따로 도시되어 있지만, 이 기능들은 하나 이상의 컴퓨터에 결합될 수 있다.

제 1 스토리지 장치(106) 뿐만 아니라 대부분의 네트워킹된 컴퓨팅 환경은 하나 이상의 제 2 또는 백업 스토리지 시스템(10)을 포함한다. 백업 스토리지 시스템(110)은 대용량이긴 하지만, 신뢰성 있는 제 2 스토리지 시스템이 사용될 수 있을지라도 일반적으로 테이프 라이브러리(tape library)가 될 수 있다. 일반적으로, 제 2 스토리지 시스템은 제 1 스토리지 장치보다 저속이지만, 오프사이트(off-site)에서 저장 및 삭제가 가능한 몇가지 형태의 분리가능한 매체(예컨대, 테이프, 자기 디스크, 또는 광 디스크)를 포함한다.

도시된 예에 있어서, 애플리케이션 서버(102)는 예컨대, 이더넷 또는 다른 통신 링크(112)를 통해 백업 스토리지 시스템(110)과 직접 통신할 수도 있다. 그러나, 이러한 연결은 비교적 느리고, 프로세서 타임 또는 네트워크 대역 등의 리소스를 소모할 수도 있다. 따라서, 도시된 바와 같은 시스템은 예컨대, SAN(108)과 백업 스토리지 시스템(110) 사이에서 파이버 채널을 사용한 통신 링크를 제공할 수 있는 하나 이상의 매체 서버(114)를 포함할 수 있다.

매체 서버(114)는 호스트 컴퓨터[유저 컴퓨터(104), 매체 서버(114), 및/또는 애플리케이션 서버(102) 등], 제 1 스토리지 장치(106), 및 백업 스토리지 시스템(110) 사이에서 데이터 전송을 제어하는 백업/복구 애플리케이션을 포함하는 소프트웨어를 실행할 수 있다. 백업/복구 애플리케이션의 예로서 Veritas, Legato사 등의 제품이 될 수 있다. 데이터 보호를 위해, 네트워킹된 컴퓨팅 환경내의 다양한 호스트 컴퓨터 및/또는 제 1 스토리지 장치로부터의 데이터는 공지의 백업/복구 애플리케이션을 사용한 백업 스토리지 시스템(110)에 주기적으로 백업될 수 있다.

물론, 상기한 바와 같이, 대부분의 네트워킹된 컴퓨터 환경은 도1에 도시된 예시적인 네트워킹된 컴퓨터 환경보다 작고, 더 적은 구성요소를 포함할 수 있다. 따라서, 매체 서버(114)는 또한, 실질적으로 단일 호스트 컴퓨터내의 애플리케이션 서버(102)와 결합될 수 있고, 백업/복구 애플리케이션은 백업 스토리지 시스템(110)에 네트워크를 통해 직접적 또는 간접적으로 연결된 임의의 호스트 컴퓨터상에서 실행될 수 있다고 인식되어야 한다.

전형적인 백업 스토리지 시스템의 일례는 다수의 테이프 카트리지, 하나 이상의 테이프 드라이브, 및 테이프 드라이브로의 카트리지의 로딩과 언로딩을 제어하는 로보틱 메카니즘을 포함하는 테이프 라이브러리이다. 백업/복구 애플리케이션은 로보틱 메타니즘이 특정 테이프 카트리지, 예컨대, 테이프 번호 0001의 위치를 결정하여, 테이프 드라이브에 테이프 카트리지를 로딩함으로써 데이터가 테이프상에 기록될 수 있도록 지시한다. 또한, 백업/복구 애플리케이션은 데이터가 테이프상으로 기록되는 포맷을 제어한다. 일반적으로, 백업/복구 애플리케이션은 SCSI 명령, 또는 기타 표준화된 명령을 사용하여 로보틱 메카니즘에 지시하고, 테이프 드라이브를 제어하여 테이프상에 데이터를 기록하고, 테이프로부터 기록된 데이터를 미리 복구시킨다.

종래의 테이프 라이브러리 백업 시스템은 속도, 신뢰성, 및 고정된 용량을 포함하는 여러가지 문제점을 가지고 있다. 대부분의 대기업은 매주 테라바이트의 데이터를 백업할 필요가 있다. 그러나, 고비용임에도 불구하고 하이-엔드 테이프(high-end tape)는 일반적으로 시간당 약 50 기가바이트(GB/hr)로 변환하는 초당 30~40 메가바이트(MB/s)의 속도로만 데이터를 판독/기록할 수 있다. 따라서, 1 또는 2 테라바이트의 데이터를 테이프 백업 시스템에 백업하기 위한 연속 데이터 전송 시간은 적어도 10~20 시간이 될 수 있다.

또한, 대부분의 테이프 제조사들은 테이프가 떨어지거나(사람이나 로보틱 메카니즘이 테이프를 이동하거나 로딩 동작중에 떨어뜨릴 수 있기 때문에 전형적인 테이프 라이브러리에 있어서 비교적 빈번히 발생할 수 있음) 극도의 온도 및 습도 등의 비이상적인 환경조건에 테이프가 노출되는 경우에 테이프로 또는 테이프로부터 데이터를 저장 또는 복구할 수 있도록 보장해주지 않는다. 따라서, 조정된 환경에서 저장 테이프의 저장에는 상당한 주의가 필요하다. 또한, 복잡한 구조의 테이프 라이브러리(로보틱 메카니즘을 포함함)는 유지비가 비싸고, 개개의 테이프 카트리지는 비교적 고가이며, 수명이 제한되어 있다.

발명의 상세한 설명

본 발명의 실시형태는 종래의 테이프 라이브러리 시스템이 갖는 문제점의 일부 또는 전부를 경감시키거나 극복하고 종래의 테이프 라이브러리 시스템에 비해 더욱 신뢰할 수 있는 백업 스토리지 시스템을 제공한다.

전체를 개괄하자면, 본 발명의 실시형태는 백업/복구 애플리케이션이 장치 및 매체를 물리적 테이프 라이브러리와 동일하게 간주하도록 종래의 테이프 백업 스토리지 시스템을 에뮬레이팅하는 랜덤 액세스 기반 스토리지 시스템을 제공한다. 본 발명의 스토리지 시스템은 소프트웨어와 하드웨어를 사용하여 물리적 테이프 매체를 에뮬레이팅하고, 하나 이상의 랜덤 액세스 디스크 어레이, 트랜스레이팅 테이프 포맷(translating tape format), 선형, 일련의 데이터를 디스크에 저장하기에 적합한 데이터로 대체시킨다. 또한, 하드웨어 및/또는 소프트웨어에서 구현된 애플리케이션은 백업 스토리지 시스템에 저장된 데이터를 복구시키기 위해 제공된다.

본 발명의 여러 실시형태에 의하면, 일련의 테이프-포맷된 데이터를 랜덤 액세스 I/O에 적합한 포맷으로 변환하는 메카니즘이 제공된다. 제 1 실시형태에 있어서, NFS(network file system) 또는 CIFS(common Internet file system) 마운팅 볼륨(mounted volume)으로서의 호스트 컴퓨터상의 테이프-포맷된 데이터의 변환된 표현을 마운팅하기 위해 메카니즘이 제공된다.

본 발명의 다른 실시형태에 의하면, 마운팅된 파일 시스템에 대한 기록을 "세이프 스토리지(safe storage)로 전환함으로써 오리지날 데이터가 변경되지 않은 상태로 남아있게 하기 위한 메카니즘이 제공된다. 제 1 실시형태에 있어서, 랜덤 액세스 I/O가 가능하도록 오리지날 데이터에 대한 실시간 변화를 추적하기 위한 메카니즘이 제공된다. 다른 실시형태에 있어서, 새로 기록된 데이터 백(data back)을 일련의 테이프-특정 I/O에 적합한 테이프-포맷된 데이터로 변환하기 위한 메카니즘이 제공된다.

제 1 실시형태에서의 방법은 백업 스토리지 시스템에 저장된 가장 최근 백업된 버전의 하나 이상의 데이터 파일에 대응하는 하나 이상의 데이터 파일을 포함하는 데이터 볼륨을 호스트 컴퓨터상에 마운팅하는 단계, 및 가장 최근 백업된 버전의 하나 이상의 데이터 파일을 보존하는 동안 백업 스토리지 시스템에 저장된 가장 최근 백업된 버전의 하나 이상의 데이터 파일보다 더 최근의 제 2 버전의 하나 이상의 데이터 파일에 대응하는 데이터를 백업 스토리지 시스템에 저장하는 단계를 포함한다. 상기 방법은 가장 최근 백업된 버전의 하나 이상의 데이터 파일과 제 2 버전의 하나 이상의 데이터 파일의 링킹(linking)을 포함할 수도 있다. 일례에 있어서, 상기 방법은 가장 최근 백업된 버전의 하나 이상의 데이터 파일과 제 2 버전의 하나 이상의 데이터 파일을 동일한 것으로 간주하는 데이터 구조의 생성을 포함할 수도 있다. 다른 예에 있어서, 제 2 버전의 하나 이상의 데이터 파일은 가장 최근 백업된 버전의 하나 이상의 데이터 파일의 수정된 버전일 수 있다.

다른 실시형태에 있어서, 백업 스토리지 시스템은 백업 데이터 세트를 저장하기 위한 백업 스토리지 매체, 및 상기한 방법을 구현하는 지시의 세트를 실행하기 위해 구성된 하나 이상의 프로세서를 포함하는 제어기를 포함한다.

다른 실시형태에 따르면, 데이터 구조가 저장되어 있는 컴퓨터 판독가능 매체가 제공되고, 상기 데이터 구조는 하나 이상의 데이터 파일을 포함하는 백업 데이터 세트에 대응하는 시스템 파일을 독자적으로 식별하는 제 1 식별자, 및 백업 데이터 세트에 있어서의 하나 이상의 데이터 파일 각각의 최근 버전이 저장된 스토리지 매체상의 개별 저장 위치를 식별하는 하나 이상의 제 2 식별자를 포함한다.

도면의 간단한 설명

첨부도면은 일정한 비율로 도시되지 않았다. 도면에 있어서, 여러 도면에 도시된 각각의 동일한 또는 거의 동일한 구성요소는 동일한 참조번호로 나타냈다. 명확함을 위해, 모든 도면에 도시된 모든 구성요소마다 참조번호를 부여하지는 않았다.

도1은 백업 스토리지 시스템을 포함하는 대형의 네트워킹된 컴퓨팅 환경의 일례를 나타낸 블럭도이다.

도2는 본 발명에 의한 스토리지 시스템을 포함하는 네트워킹된 컴퓨팅 환경의 제 1 실시형태의 블럭도이다.

도3은 본 발명에 의한 스토리지 시스템의 제 1 실시형태의 블럭도이다.

도4는 본 발명에 의한 스토리지 시스템의 제 1 실시형태의 가상 레이아웃을 나타낸 블럭도이다.

도5는 본 발명의 실시형태에 의한 시스템 파일의 일례의 개략적인 레이아웃이다.

도6은 본 발명의 실시형태에 의한 테이프 디렉토리 구조의 일례를 나타낸 도면이다.

도7은 본 발명의 실시형태에 의한 종합 풀 백업을 생성하는 방법의 일례를 나타낸 도면이다.

도8은 본 발명의 실시형태에 의한 종합 풀 백업을 포함하는 백업 데이터 세트의 시리즈의 일례의 개략적인 도면이다.

도9는 메타데이터 캐시 구조(metadata cache structure)의 일례를 나타낸 도면이다.

도10은 종합 풀 백업 데이터 세트를 저장하는 가상 카트리지의 일례를 나타낸 도면이다.

도11은 종합 풀 백업 데이터 세트를 저장하는 가상 카트리지의 다른 예를 나타낸 도면이다.

도12는 본 발명의 실시형태에 의한 백업 스토리지 시스템으로부터 데이터를 복구하기 위한 방법의 제 1 실시형태의 흐름도이다.

도13은 본 발명의 실시형태에 의한 백업 스토리지 시스템을 포함하는 네트워킹된 컴퓨팅 환경의 다른 실시형태의 블럭도이다.

도14는 본 발명의 실시형태에 의한 파일 디스크립터 구조(file descriptor structure)의 일례를 나타낸 도면이다.

도15는 파일 데이터가 테이프 포맷으로 저장될 수 있는 방법의 일례를 나타낸 도면이다.

도16은 도15에 도시된 파일에 대한 파일 디스크립터를 나타낸 도면이다.

도17은 본 발명의 제 1 실시형태에 의해 마운팅된 데이터 볼륨에 데이터를 기록하는 방법의 흐름도이다.

도18은 새로 기록된 파일의 일례를 나타낸 도면이다.

도19는 본 발명의 일실시형태에 의한 오리지널 파일, 새로 기록된 파일, 및 최종 수정된 파일 사이의 관계에 대한 일례를 나타낸 도면이다.

도20은 도19에 도시된 수정된 파일을 나타내는 파일 디스크립터의 일례를 나타낸 도면이다.

실시예

다양한 실시형태를 첨부도면을 참조하여 더 상세히 설명할 것이다. 본 발명은 도면에 도시되거나 후술되는 설명에 있어서의 구성요소의 배치 및 구조의 상세사항에 한정되지 않는다. 본 발명은 다양한 방법과 형태로 실시될 수 있다. 또한, 여기서 사용된 표현 및 용어는 본 발명을 한정하고자 하는 것이 아닌 설명을 위한 것이다. 여기에 사용된 "포함하다", "가지다", "구성되다", "이루어지다" 등의 표현은 후술되는 아이템과 동등한 것뿐만 아니라 추가적인 아이템을 포함하는 의미이다.

이 명세서에 사용된 "호스트 컴퓨터"라는 용어는 스토리지 시스템 또는 다른 호스트 컴퓨터와 통신이 가능한 퍼스널 컴퓨터, 워크스테이션, 메인프레임, 네트워킹된 클라이언트, 서버와 같은 하나 이상의 프로세서를 갖는 임의의 컴퓨터를 의미한다. 호스트 컴퓨터는 유저 컴퓨터(유저 워크스테이션, PC, 메인프레임 등이 될 수 있음)뿐만 아니라 매체 서버, 및 애플리케이션 서버(도1을 참조하여 상기한 바와 같음)를 포함할 수 있다. 또한, 이 명세서 내에서 "네트워킹된 컴퓨터 환경"이라는 용어는 스토리지 시스템이 각각의 호스트 컴퓨터와 통신이 가능한 방법으로 하나 이상의 공유된 스토리지 시스템에 복수의 호스트 컴퓨터가 연결된 임의의 컴퓨팅 환경을 포함한다. 파이버 채널은 본 발명의 실시형태에 사용될 수 있는 통신 네트워크의 일례이다. 그러나, 이 네트워크는 파이버 채널에 한정되지 않고, 다양한 네트워크 구성요소는 파이버 채널 대신에 또는 이에 추가로 토큰링, 또는 이더넷 등의 임의의 네트워크를 통해 또는 다른 네트워크 연결의 조합을 통해 서로 통신이 가능하다고 이해되어야 한다. 또한, 본 발명의 실시형태는 SCSI 또는 병렬 SCSI와 같은 버스 토폴로지에 사용될 수도 있다.

본 발명의 다양한 실시형태에 의하면, 분리가능한 매체 기반 스토리지 시스템을 에뮬레이팅하기 위해 하나 이상의 디스크 어레이를 사용할 수 있는 가상 분리가능한 매체 라이브러리 백업 스토리지 시스템이 제공된다. 본 발명의 실시형태에 의하면, 유저가 기존의 백업 절차를 수정 또는 조정하거나 새로운 백업/복구 애플리케이션을 구매할 필요 없이 분리가능한 매체(테이프, 자기 디스크, 광 디스크 등)에 데이터를 백업하는데 사용되는 것과 동일한 백업/복구 애플리케이션을 사용하여 디스크 어레이에 데이터를 백업할 수 있다. 상기한 제 1 실시형태에 있어서, 테이프가 에뮬레이팅된 분리가능한 매체는 테이프이며, 본 발명의 백업 스토리지 시스템은 테이프, 및 종래의 테이프 라이브러리 시스템에서 테이프를 핸들링하는데 사용된 로보틱 메카니즘을 포함하는 테이프 라이브러리 시스템을 에뮬레이팅한다.

본 발명의 실시형태에 의한 스토리지 시스템은 호스트 컴퓨터(백업/복구 애플리케이션을 구동함)와 백업 스토리지 매체를 함께 인터페이스하는 하드웨어와 소프트웨어를 포함한다. 스토리지 시스템은 테이프, 또는 다른 형태의 분리가능한 스토리지 매체를 에뮬레이팅하여 백업/복구 애플리케이션이 장치 및 매체를 물리적 테이프 라이브러리와 동일하게 간주하게 되고, 선형, 일련의 테이프 포맷 데이터를 랜덤 액세스 디스크상에 저장하기에 적합한 데이터로 변환하도록 설계될 수 있다. 이러한 방식으로, 본 발명의 스토리지 시스템은 새로운 백업/복구 애플리케이션 소프트웨어 또는 정책을 필요로하지 않고 향상된 기능(이하 설명되는 바와 같이, 유저가 개인적으로 백업된 유저 파일을 검색할 수 있게 하는 등의 기능)을 제공할 수 있다.

도2는 본 발명의 실시형태에 의한 백업 스토리지 시스템(170)을 포함하는 네트워킹된 컴퓨팅 환경의 제 1 실시형태의 블럭도를 나타낸다. 도시된 바와 같이, 호스트 컴퓨터(120)는 네트워크 연결(121)을 통해 스토리지 시스템(170)에 연결된다. 이 네트워크 연결(121)은 예컨대, 호스트 컴퓨터(120)와 스토리지 시스템(170)간의 고속 데이터 전송이 가능한 파이버 채널 연결 등이 될 수 있다. 호스트 컴퓨터(120)는 하나 이상의 애플리케이션 서버(102)(도1) 및/또는 매체 서버(114)(도1)가 되거나 포함할 수 있고, 네트워킹된 컴퓨팅 환경내에 존재하는 임의의 컴퓨터 또는 제 1 스토리지 시스템(110)(도1)로부터 데이터의 백업을 가능하게 할 수 있는 것으로 인식되어야 한다. 또한, 하나 이상의 유저 컴퓨터(136)는 이더넷 연결 등의 다른 네트워크 연결(138)을 통해 스토리지 시스템(170)에 연결될 수도 있다. 후술하는 바와 같이, 스토리지 시스템은 유저 컴퓨터(136)의 유저가 스토리지 시스템으로부터 백업된 유저 파일을 보고 선택적인 복구가 가능하게 할 수도 있다.

스토리지 시스템은 예컨대, 아래에 보다 상세히 설명된 바와 같은 하나 이상의 디스크 어레이가 될 수 있는 백업 스토리지 매체(126)를 포함한다. 백업 스토리지 매체(126)는 호스트 컴퓨터(120)로부터 백업된 데이터를 위한 실제 저장 공간을 제공한다. 그러나, 스토리지 시스템(170)은 테이프 라이브러리와 같은 분리가능한 매체 스토리지 시스템을 에뮬레이팅하여 호스트 컴퓨터(120)상에 백업/복구 애플리케이션을 실행함으로써 종래 분리가능한 스토리지 매체에 데이터가 백업된 것처럼 보이도록 하는 추가적인 하드웨어, 및 소프트웨어를 포함할 수도 있다. 따라서, 도2에 도시된 바와 같이, 스토리지 시스템(170)은 예컨대, 테이프와 같은 가상 또는 에뮬레이팅된 분리가능한 스토리지 매체를 의미하는 "에뮬레이팅된 매체"(134)를 포함할 수 있다. 이 "에뮬레이팅된 매체"(134)는 스토리지 시스템 소프트웨어 및/또는 하드웨어에 의해 호스트 컴

퓨터에 제공되고, 물리적 스토리지 매체로서 호스트 컴퓨터에 보여진다. 실제 백업 스토리지 매체(126)와 에뮬레이팅된 매체(134) 사이의 인터페이싱은 이하에 상세히 설명하는 바와 같이, 호스트 컴퓨터(120)로부터 데이터를 받아들여 백업 스토리지 매체(126)에 데이터를 저장하는 스위칭 네트워크(132) 및 스토리지 시스템 제어기(도시되지 않음)가 될 수 있다. 이러한 방식으로, 스토리지 시스템은 종래의 테이프 스토리지 시스템을 호스트 컴퓨터(120)에 에뮬레이팅한다.

제 1 실시형태에 따르면, 스토리지 시스템은 스토리지 시스템(170)상의 호스트 컴퓨터(120)로부터 백업된 유저 데이터와 관련된 메타데이터(metadata)를 저장하는 "로지컬 메타데이터 캐시"(242)를 포함할 수 있다. 여기서 사용된 "메타데이터"라는 용어는 유저 데이터에 대한 정보를 나타내고, 실제 유저 데이터의 특성을 기술하는 데이터를 의미한다. 로지컬 메타데이터 캐시(242)는 유저 및/또는 소프트웨어 애플리케이션이 백업된 유저 파일을 랜덤하게 배치하고, 서로 유저 파일을 비교하고, 그렇지 않으면, 백업된 유저 파일에 액세스하고 조정할 수 있게 하는 검색가능한 데이터의 모음을 의미한다. 로지컬 메타데이터 캐시(242)내에 저장된 데이터를 사용할 수 있는 소프트웨어 애플리케이션의 두가지 예는 보다 상세히 후술될 엔드 유저 복구 애플리케이션(300) 및 종합 풀 백업 애플리케이션(240)을 포함한다.

요컨대, 종합 풀 백업 애플리케이션(240)은 기존의 하나 이상의 풀 백업 데이터 세트와 하나 이상의 증분적 백업 데이터 세트로부터 종합 풀 백업 데이터 세트를 생성할 수 있다. 종합 풀 백업은 주기적(예컨대, 매주) 풀 백업을 수행할 필요가 없기 때문에 시간과 네트워크 리소스를 상당히 절약할 수 있다. 종합 풀 백업 애플리케이션(240)이 상세히 후술될 것이다. 엔드 유저 복구 애플리케이션(300)은 엔드 유저[예컨대, 유저 컴퓨터(136)의 오퍼레이터]가 스토리지 시스템(170)으로부터 미리 백업된 유저 파일을 브라우징, 로케이팅, 뷰잉, 및/또는 복구할 수 있게 한다. 이에 대해서도 상세히 후술한다.

상기한 바와 같이, 스토리지 시스템(170)은 호스트 컴퓨터(120)와 백업 스토리지 매체(126)를 인터페이스시키는 하드웨어 및 소프트웨어를 포함한다. 본 발명의 실시형태에 의한 하드웨어 및 소프트웨어는 종래의 테이프 라이브러리 백업 시스템을 에뮬레이팅하여 호스트 컴퓨터(120)의 관점에서는 테이프상에 데이터가 백업된 것으로 보이지만, 실제로는 복수의 디스크 어레이와 같은 다른 스토리지 매체상에 백업이 된다.

도3은 본 발명의 실시형태에 의한 스토리지 시스템(170)의 제 1 실시형태를 나타낸 블럭도이다. 제 1 실시형태에 있어서, 스토리지 시스템(170)의 하드웨어는 스토리지 시스템 제어기(122), 및 백업 스토리지 매체(126)에 스토리지 시스템 제어기(122)를 연결하는 스위칭 네트워크(132)를 포함한다. 스토리지 시스템 제어기(122)는 스토리지 시스템 소프트웨어의 전부 또는 일부를 구동할 수 있는 프로세서(127)(단일 프로세서 또는 복수의 프로세서가 될 수 있다), 및 메모리(129)(RAM, ROM, PROM, EEPROM, 플래시 메모리, 및 그 조합 등)를 포함한다. 메모리(129)는 백업 스토리지 매체(126)에 저장된 데이터에 관련된 메타데이터를 저장하는데 사용될 수도 있다. 본 발명의 실시형태를 실행하는 프로그램밍 코드를 포함하는 소프트웨어는 일반적으로 RAM, ROM, 광 디스크, 자기 디스크, 또는 테이프 등의 컴퓨터가 기록 및/또는 판독할 수 있는 비휘발성 기록 매체에 저장되고, 이후 프로세서(127)에 의해 실행될 수 있는 메모리(129)로 복사된다. 이러한 프로그래밍 코드는 복수의 프로그래밍 언어, 예컨대, Java, Visual Basic, C, C#, 또는 C++, Fortran, Pascal, Eiffel, Basic, COBAL, 또는 그 조합 중의 어느 하나로 기록될 수 있고, 본 발명은 특정 프로그래밍 언어에 한정되지 않는다. 일반적으로, 동작시, 프로세서(127)는 본 발명의 실시형태를 실행하는 코드와 같은 데이터가 비휘발성 기록 매체로부터 비휘발성 기록 매체보다 프로세서에 의해 정보에 빠르게 액세스 가능하게 하는 RAM과 같은 다른 형태의 메모리에서 판독되게 한다.

도3에 도시된 바와 같이, 제어기(122)는 제어기(122)를 호스트 컴퓨터(120) 및 스위칭 네트워크(132)에 연결하는 다수의 포트 어뎁터(124a, 124b, 124c)를 포함하기도 한다. 도시된 바와 같이, 호스트 컴퓨터(120)는 예컨대, 파이버 채널 포트 어뎁터 등의 포트 어뎁터(124a)를 통해 스토리지 시스템에 연결된다. 스토리지 시스템 제어기(122)를 통해 호스트 컴퓨터(120)는 데이터를 백업 스토리지 매체(126)에 백업하고, 백업 스토리지 매체(126)로부터 데이터를 복구할 수 있다.

도시된 예에 있어서, 스위칭 네트워크(132)는 하나 이상의 파이버 채널 스위치(128a, 128b)를 포함할 수 있다. 스토리지 시스템 제어기(122)는 스토리지 시스템 제어기를 파이버 채널 스위치(128a, 128b)에 연결하는 복수의 파이버 채널 포트 어뎁터(124b, 124c)를 포함한다. 파이버 채널 스위치(128a, 128b)를 통해 스토리지 시스템 제어기(122)는 데이터가 백업 스토리지 매체(126)에 백업되게 한다. 도3에 도시된 바와 같이, 스위칭 네트워크(132)는 이더넷 포트 어뎁터(125a, 125b)를 통해 스토리지 시스템 제어기(122)에 연결된 하나 이상의 이더넷 스위치(130a, 130b)를 더 포함할 수 있다. 일례에 있어서, 스토리지 시스템 제어기(122)는 예컨대, LAN(103)에 연결되어 스토리지 시스템(170)이 후술하는 바와 같이 호스트 컴퓨터(예컨대, 유저 컴퓨터)와 통신 가능하게 하는 다른 이더넷 포트 어뎁터(125c)를 더 포함한다.

도3에 도시된 예에 있어서, 스토리지 시스템 제어기(122)는 2개의 파이버 채널 스위치와 두개의 이더넷 스위치를 포함하는 스위칭 네트워크를 통해 백업 스토리지 매체(126)에 연결된다. 스토리지 시스템(170)내의 두개 이상의 각각의 형태의 스위치의 제공은 시스템내에서의 모든 단일 포인트의 실패를 제거한다. 즉, 하나의 스위치[예컨대, 파이버 채널 스위치(128a)]가 실패하더라도 스토리지 시스템 제어기(122)는 여전히 다른 스위치를 통해 백업 스토리지 매체(126)와 통신할

수 있다. 이러한 배열은 신뢰도 및 속도의 면에서 장점을 갖는다. 예컨대, 상기한 바와 같이, 여분의 구성요소의 제공과 단일 포인트 실패의 제거를 통해 신뢰도가 향상된다. 또한, 몇몇 실시형태에 있어서, 스토리지 시스템 제어기는 병렬 파이버 채널 스위치의 전부 또는 일부를 사용한 백업 스토리지 매체(126)상에 데이터를 백업할 수 있기 때문에 전체 백업 속도가 빨라진다. 그러나, 시스템은 두개 이상의 각각의 형태의 스위치를 포함하거나 스위칭 네트워크가 파이버 채널 및 이더넷 스위치를 포함할 필요가 없다. 또한, 백업 스토리지 매체(126)가 단일 디스크 어레이를 포함하는 예에 있어서는 스위치가 전혀 필요치 않다.

상기한 바와 같이, 제 1 실시형태에 있어서, 백업 스토리지 매체(126)는 하나 이상의 디스크 어레이를 포함할 수 있다. 하나의 바람직한 실시형태에 있어서, 백업 스토리지 매체(126)는 복수의 ATA 또는 SATA 디스크를 포함한다. 이러한 디스크는 시중에서 쉽게 구할 수 있는 제품으로서, EMC, IBM 등의 제조사의 종래 저장 어레이 제품에 비해 비교적 저렴 수 있다. 또한, 분리가능한 매체(예컨대, 테이프)의 가격과 이러한 매체가 한정된 수명을 갖는다는 사실을 염두에 두는 경우, 이러한 매체는 가격 면에서 종래의 테이프 기반 백업 스토리지 시스템에 필적한다. 또한, 이러한 디스크는 테이프에 비해 고속으로 판독/기록이 가능하다. 예컨대, 단일 파이버 채널 연결을 통해 테이프의 백업 속도보다 확실히 빠른(예컨대, 10배 정도) 약 540 GB/hr으로 환산되는 적어도 150MB/s의 속도로 데이터를 디스크상에 백업할 수 있다. 또한, 일부 파이버 채널 연결은 병렬로 구현될 수 있기 때문에 더욱 속도가 증가한다. 본 발명에 실시형태에 따르면, 백업 스토리지 매체는 복수의 RAID(Redundant Array of Independent Disks) 방식을 구현하도록 구성될 수 있다. 예컨대, 제 1 실시형태에 있어서, 백업 스토리지 매체는 RAID-5 구현으로서 구성될 수 있다.

상기한 바와 같이, 본 발명에 의한 실시형태는 테이프 카트리지를 물리적 백업 스토리지 매체로서 교체하도록 디스크 어레이를 사용한 종래의 테이프 라이브러리 백업 시스템을 에뮬레이팅함으로써 "가상 테이프 라이브러리"를 제공한다. 종래의 테이프 라이브러리에 제공되는 물리적 테이프 카트리지는 "가상 카트리지"라는 용어에 의해 대체된다. "가상 테이프 라이브러리"라는 용어는 예컨대, 하나 이상의 디스크 어레이로서 소프트웨어 및/또는 물리적 하드웨어에서 구현될 수 있는 에뮬레이팅된 테이프 라이브러리를 의미하는 것으로 인식되어야 한다. 여기서는 주로 에뮬레팅된 테이프를 언급하고 있지만, 스토리지 시스템은 CD-ROM, DVD-ROM 등의 다른 스토리지 매체를 에뮬레이팅할 수도 있고, "가상 카트리지"라는 용어는 일반적으로 에뮬레이팅된 테이프 또는 에뮬레이팅된 CD 등의 에뮬레이팅된 스토리지 매체를 의미하는 것으로 인식되어야 한다. 제 1 실시형태에 있어서, 가상 카트리지는 실제로 하나 이상의 하드디스크에 대응한다.

따라서, 제 1 실시형태에 있어서, 소프트웨어 인터페이스는 테이프 라이브러리를 에뮬레이팅하도록 제공되어 백업/복구 애플리케이션에 있어서, 데이터가 테이프에 백업되는 것처럼 보이게 된다. 그러나, 실제 테이프 라이브러리는 이 디스크 어레이상에 데이터가 실제로 백업되도록 하나 이상의 디스크 어레이에 의해 대체된다. 이하, 스토리지 시스템(170)에 포함된 소프트웨어의 다양한 형태, 특성, 및 동작을 설명하기로 한다.

소프트웨어가 스토리지 시스템(170)에 "포함된다"라고 설명할 수도 있고, 스토리지 시스템 제어기(122)(도3)의 프로세서(127)에 의해 실행될 수 있다고 할 수도 있지만, 스토리지 시스템 제어기(122)상에서 모든 소프트웨어가 실행될 필요는 없다. 종합 풀 백업 애플리케이션 및 엔드 유저 복구 애플리케이션 등의 소프트웨어 프로그램은 호스트 컴퓨터 및/또는 유저 컴퓨터에서 실행될 수 있고, 스토리지 시스템 제어기, 호스트 컴퓨터, 및 유저 컴퓨터의 전체 또는 일부를 거쳐서 이 부분이 분배될 수 있다. 따라서, 스토리지 시스템 제어기가 컴퓨터 등의 포함된 물리적 엔티티일 필요는 없다. 스토리지 시스템(170)은 매체 서버(114) 또는 애플리케이션 서버(102) 등의 호스트 컴퓨터상에 존재하는 소프트웨어와 통신할 수 있다. 또한, 스토리지 시스템은 동일 또는 상이한 호스트 컴퓨터상에 존재하거나 이 호스트 컴퓨터에서 구동될 수 있는 몇몇 소프트웨어 애플리케이션을 포함할 수 있다. 또한, 스토리지 시스템(170)은 일부 실시형태에 있어서, 스토리지 시스템(170)은 분리된 장치로서 실시될 수 있을지라도 분리된 장치로 한정되지 않는다. 일례에 있어서, 스토리지 시스템(170)은 종래의 테이프 라이브러리 백업 시스템 "플러그 앤 플레이" 대체로서 작용하는 독립 유닛으로 제공될 수 있다(즉, 기존의 백업 절차 및 정책을 수정할 필요가 없음). 이러한 스토리지 시스템 유닛은 종래 백업 시스템을 포함하는 네트워킹된 컴퓨팅 환경에 사용되어 여분의 또는 추가적인 저장 용량을 제공할 수도 있다.

상기한 바와 같이, 제 1 실시형태에 의하면, 호스트 컴퓨터(120)[예컨대, 애플리케이션 서버(102) 또는 매체 서버(114)가 될 수 있음, 도1 참조]는 이 호스트 컴퓨터(120)를 스토리지 시스템(170)에 연결하는 네트워크 링크(예컨대, 파이버 채널 링크)(121)를 통해 백업 스토리지 매체(126)상에 데이터를 백업할 수 있다. 주로 에뮬레이팅된 매체상에 데이터를 백업하는 것에 대해 후술하겠지만, 이 원리는 에뮬레이팅된 매체로부터 백업 데이터를 복구하는 것에도 적용되는 것으로 인식되어야 한다. 호스트 컴퓨터(120)와 에뮬레이팅된 매체(134) 사이의 데이터 흐름은 상기한 바와 같이, 백업/복구 애플리케이션에 의해 제어될 수 있다. 백업/복구 애플리케이션의 관점에서는 데이터가 물리적 버전의 에뮬레이팅된 매체상에 실제로 백업된 것으로 보여질 수 있다.

도4에 도시된 바와 같이, 스토리지 시스템 소프트웨어(150)는 에뮬레이팅된 매체를 의미하고, 호스트 컴퓨터(120)상에 존재하는 백업/복구 애플리케이션(140)과 백업 스토리지 매체(126) 사이의 인터페이스를 제공하는 하나 이상의 논리적 추상층(logical abstraction layer)을 포함한다. 소프트웨어(150)는 백업/복구 애플리케이션(140)으로부터 테이프 포맷 데이터를 받아들여, 랜덤 액세스 디스크(예컨대, 하드디스크, 광 디스크 등)상에 저장하기에 적합한 데이터로 변환한다. 일례에 있어서, 이 소프트웨어(150)는 스토리지 시스템 제어기(122)의 프로세서(127)상에서 실행되고, 메모리(129)(도3)상에 저장될 수 있다.

제 1 실시형태에 의하면, 상기 소프트웨어(150)는 테이프, 테이프 드라이브, 및 테이프를 테이프 드라이브로/로부터 전송하는데 사용되는 로보틱메카니즘의 SCSI 에뮬레이션을 제공할 수 있는 가상 테이프 라이브러리(VTL)층(142)를 의미하는 층을 포함할 수 있다. 백업/복구 애플리케이션(140)은 예컨대, 화살표(144)로 표시된 SCSI 명령 등을 사용하여 VTL(142)와 통신(예컨대, 에뮬레이팅된 매체에 데이터를 백업 또는 기록)할 수 있다. 따라서, VTL은 다른 스토리지 시스템 소프트웨어 및 하드웨어와 백업/복구 애플리케이션 사이의 소프트웨어 인터페이스를 형성할 수 있어 에뮬레이팅된 스토리지 매체(134)를 백업/복구 애플리케이션에 제공하여, 에뮬레이팅된 매체가 종래 분리가능한 백업 스토리지 매체로서 백업/복구 애플리케이션으로 보여지게 한다.

파일 시스템층(146)으로 언급된 제 2 소프트웨어층은 에뮬레이팅된 스토리지 매체(VTL로 표현됨)와 물리적 백업 스토리지 매체(126) 사이의 인터페이스를 제공할 수 있다. 일례에 있어서, 파일 시스템(146)은 작은 운영시스템으로서 동작하여 화살표(148)로 표시된 SCSI 명령 등을 사용하여 백업 스토리지 매체(126)와 통신함으로써 백업 스토리지 매체(126)로/로부터 데이터를 판독 및 기록할 수 있다.

제 1 실시형태에 있어서, VTL은 일반적인 테이프 라이브러리 지원을 제공하고, 임의의 SCSI 매체 체인저(SCSI media changer)를 지원할 수 있다. 에뮬레이티드 테이프 장치는 IBM LTO-1, LTO-2 테이프 장치, Quantum SuperDLT320 테이프 장치, Quantum P3000 테이프 라이브러리 시스템, 또는 StorageTek L180 테이프 라이브러리 시스템 등을 포함할 수 있지만, 이것에 한정되지 않는다. VTL내의 각 가상 카트리지는 데이터가 저장됨에 따라 동적으로 늘어날 수 있는 파일이다. 이것은 고정된 크기를 갖는 종래의 테이프 카트리지와 반대이다. 하나 이상의 가상 카트리지는 도5를 참조하여 후술될 시스템 파일에 저장될 수 있다.

도5는 본 발명의 실시형태에 의한 시스템 파일(200)을 나타낸 파일 시스템 소프트웨어(146)내의 데이터 구조의 일례를 나타낸 도면이다. 이 실시형태에 있어서, 시스템 파일(200)은 헤더(202) 및 데이터(204)를 포함한다. 헤더(202)는 시스템 파일에 저장된 각 가상 카트리지를 식별하는 정보를 포함할 수 있다. 헤더(202)는 가상 카트리지가 기록방지되어 있는지 여부, 가상 카트리지의 생성/수정 날짜 등의 정보를 포함할 수 있다. 일례에 있어서, 헤더(202)는 각 가상 카트리지를 독자적으로 식별하고, 스토리지 시스템에 저장된 다른 가상 카트리지로부터 각 가상 카트리지를 구별하는 정보를 포함한다. 예컨대, 이 정보는 가상 카트리지의 이름, 및 식별번호(예컨대, 로보틱 메카니즘에 의해 테이프가 식별될 수 있도록 일반적으로 물리적 테이프에 제공되는 바코드에 대응함)를 포함할 수 있다. 헤더(202)는 각 가상 카트리지의 용량, 최종 수정된 날짜 등의 추가적인 정보를 포함할 수도 있다.

본 발명의 제 1 실시형태에 의하면, 헤더(202)의 크기는 시스템이 추적가능한 이러한 데이터의 독특한 세트의 수와 저장된 데이터의 형태(예컨대, 하나 이상의 호스트 컴퓨터 시스템으로부터 데이터 백업을 나타내는 가상 카트리지)를 나타내도록 극대화될 수 있다. 예컨대, 테이프 스토리지 시스템에 일반적으로 백업된 데이터는 다수의 시스템 및 유저 파일을 나타내는 대형 데이터 세트에 의해 일반적으로 특징지어진다. 데이터 세트가 크기 때문에, 이것에 대응하여 추적될 비연속 데이터 파일의 수는 적을 수 있다. 따라서, 제 1 실시형태에 있어서, 헤더(202)의 크기는 효과적으로 추적하기에 너무 많은 데이터를 저장하는 경우(즉, 헤더가 너무 큰 것)와 카트리지 식별자의 충분한 수를 저장할 공간이 모자라는 경우(즉, 헤더가 너무 작은 것) 사이의 절충을 통해 선택될 수 있다. 예시적인 제 1 실시형태에 있어서, 헤더(202)는 시스템 파일(200)의 최초 32MB를 활용한다. 그러나, 헤더(202)는 시스템의 필요, 및 이 시스템의 필요와 용량에 따른 특성에 의거한 여러가지 크기를 가질 수 있고, 헤더(202)를 위한 여러가지 크기를 선택할 수 있는 것으로 인식되어야 한다.

백업/복구 애플리케이션의 관점에서는 가상 카트리지는 속성과 특징이 모두 동일한 물리적 테이프 카트리지로서 보여진다. 즉, 백업 복구 애플리케이션에 있어서 가상 카트리지가 일련의 기록된 테이프로서 보여진다. 그러나, 하나의 바람직한 실시형태에 있어서, 가상 카트리지에 저장된 데이터는 백업 스토리지 매체(126)상에 일련의 포맷으로 저장되지 않는다. 오히려, 가상 카트리지상에 기록된 것으로 보여지는 데이터는 실제로 랜덤 액세스가 가능한 디스크 포맷 데이터로서 스토리지 시스템의 파일내에 저장된다. 메타데이터는 저장된 데이터를 가상 카트리지에 링크하여 백업/복구 애플리케이션이 카트리지 포맷으로 데이터를 판독 및 기록하는데 사용된다.

따라서, 바람직한 하나의 실시형태를 개괄하자면, 유저 및/또는 시스템 데이터("파일 데이터"를 의미함)는 호스트 컴퓨터(120)로부터 스토리지 시스템(170)에 의해 수신되고, 백업 스토리지 매체(126)를 이루는 디스크 어레이상에 저장된다. 스토리지 시스템의 소프트웨어(150)(도4) 및/또는 하드웨어는 이 파일 데이터를 하기에 보다 상세히 설명된 바와 같이, 시스템 파일의 형태로 백업 스토리지 매체(126)에 기록한다. 메타데이터는 스토리지 시스템 제어기에 의해 백업된 파일 데이터로부터 추출되어 백업된 유저 및/또는 시스템 파일의 속성을 추적한다. 예컨대, 각 파일에 대한 이 메타데이터는 파일명, 파일의 생성일 또는 최종 수정일, 파일에 대한 엔크립션 정보(encryption information), 및 기타 정보를 포함할 수 있다. 또한, 메타데이터는 가상 카트리지에 파일을 링크하는 각 파일마다 스토리지 시스템에 의해 생성될 수 있다. 이러한 메타데이터를 사용하여, 소프트웨어는 호스트 컴퓨터에 테이프 카트리지의 에뮬레이션을 제공하지만, 파일 데이터는 실제로 테이프 포맷으로 저장되지 않고, 오히려 후술하는 바와 같이, 시스템 파일에 저장된다. 일련의 카트리지 포맷보다는 오히려 시스템 파일에 데이터를 저장하는 것은 특정 파일을 찾기 위해 일련의 데이터를 통해 스캔할 필요가 없이 개개의 파일에 고속이며, 효율적인 랜덤 액세스를 가능하게 하는 장점을 가질 수 있다.

상기한 바와 같이, 제 1 실시형태에 의하면, 파일 데이터(즉, 유저 및/또는 시스템 데이터)는 시스템 파일로서 백업 스토리지 매체에 저장되고, 각 시스템 파일은 실제 유저 및/또는 시스템 파일인 데이터와 헤더를 포함한다. 각 시스템 파일(200)의 헤더(202)는 유저 및/또는 시스템 파일을 가상 카트리지에 링크하는 메타데이터를 포함한 테이프 디렉토리(206)를 포함한다. "메타데이터"라는 용어는 유저 및/또는 시스템 파일 데이터가 아닌 실제의 유저 및/또는 시스템 데이터의 속성을 나타내는 데이터를 의미한다. 일례에 의하면 테이프 디렉토리는 바이트 레벨 아래의 가상 카트리지상의 데이터 레이아웃을 규정할 수 있다. 제 1 실시형태에 있어서, 테이프 디렉토리(206)는 도6에 도시된 바와 같이, 테이블 구조를 갖는다. 상기 테이블은 저장된 정보 타입에 관한 칼럼(220)[예컨대, 데이터, 파일 마커(FM) 등], 바이트로 사용된 디스크 블럭의 크기에 대한 칼럼(222), 및 파일 데이터가 저장된 디스크 블럭의 수를 반영하는 칼럼(224)을 포함한다. 따라서, 테이프 디렉토리는 제어기가 백업 스토리지 매체(126)에 저장된 임의의 데이터 파일에 랜덤(일련의 반대) 액세스할 수 있게 한다. 예컨대, 도6에 도시된 바와 같이, 테이프 디렉토리는 파일의 데이터(226)가 시스템 파일(200)의 시작으로부터 하나의 블럭을 시작하는 것을 지시하기 때문에 데이터 파일(226)은 가상 테이프상에 신속히 배치될 수 있다. 이 하나의 블럭은 파일 마커(FM)에 대응하기 때문에 크기를 갖지 않는다. 파일 마커는 시스템 파일에 저장되지 않는다. 즉, 파일 마커는 제로 데이터(zero data)에 대응한다. 테이프 디렉토리는 파일 마커를 포함하는데, 이는 그것들이 종래의 테이프 및 백업/복구 애플리케이션에 의해 사용됨으로써 테이퍼 파일과 함께 파일 마커를 기록하고, 가상 카트리지를 볼때 파일 마커를 보고싶어하기 때문이다. 따라서, 파일 마커는 테이프 디렉토리내에서 추적을 행한다. 그러나, 파일 마커는 임의의 데이터를 나타내지 않기 때문에 시스템 파일의 데이터 섹션내에 저장되지 않는다. 따라서, 파일의 데이터(226)는 화살표(205)로 표시된 시스템 파일의 데이터 섹션의 처음부분에서 시작되고, 길이는 1024 바이트이다(즉, 하나의 디스크 블럭은 크기가 1024 바이트이다). 다른 파일 데이터는 데이터의 양, 즉, 데이터 파일의 크기에 따라 1024 바이트가 아닌 다른 블럭 크기로 저장될 수 있는 것으로 인식되어야 한다. 예컨대, 더 큰 데이터 파일은 효율을 위해 더 큰 블럭 크기를 사용하여 저장될 수 있다.

일례에 있어서, 테이프 디렉토리는 스토리지 시스템에 백업된 각 데이터 파일에 관련된 "파일 디스크립터"에 포함될 수 있다. 파일 디스크립터는 스토리지 시스템에 저장된 데이터 파일(204)에 관련된 메타데이터를 포함한다. 제 1 실시형태에 있어서, 파일 디스크립터는 대부분의 유닉스 기반 컴퓨터 시스템에 사용되는 테이프 아카이브(archive)(타르) 포맷과 같은 표준화된 포맷으로 구현될 수 있다. 각 파일 디스크립터는 유저 파일에 대응하는 이름, 유저 파일이 생성/수정된 날짜, 유저 파일의 크기, 및 유저 파일에 대한 액세스 제한 여부 등의 정보를 포함할 수 있다. 파일 디스크립터에 저장된 추가정보는 데이터가 복사된 디렉토리 구조를 설명하는 정보를 더 포함할 수 있다. 따라서, 파일 디스크립터는 하기에 보다 상세히 설명된 바와 같이, 대응하는 데이터 파일에 관한 검색가능한 메타데이터를 포함할 수 있다.

백업/복구 애플리케이션의 관점에서는 임의의 가상 카트리지는 파일 디스크립터에 대응하는 복수의 데이터 파일을 포함할 수 있다. 스토리지 시스템 소프트웨어의 관점에서는 데이터 파일이 예컨대, 특정 백업 작업에 링크될 수 있는 시스템 파일에 저장된다. 예컨대, 특성 시간에 하나의 호스트 컴퓨터에 의해 실행된 백업은 하나 이상의 가상 카트리지에 대응될 수 있는 하나의 시스템 파일을 생성할 수 있다. 따라서, 가상 카트리지는 임의의 크기일 수 있고, 가상 카트리지에 저장되는 유저 파일이 증가됨에 따라 동적으로 늘어날 수 있다.

상기 도3을 다시 참조하면, 스토리지 시스템(170)은 종합 풀 백업 소프트웨어 애플리케이션(240)을 포함할 수 있다. 제 1 실시형태에 있어서, 호스트 컴퓨터(120)는 에뮬레이팅된 매체(134)상에 데이터를 백업하여 하나 이상의 가상 카트리지를 형성한다. 몇몇 컴퓨터 환경에 있어서, "풀 백업", 즉, 네트워크내의 제 1 스토리지 시스템(도1)에 저장된 모든 데이터의 백업 복사는 주기적으로(예컨대, 매주) 달성될 수 있다. 이 처리는 일반적으로 복사될 데이터가 대용량이기 때문에 매우 시간이 많이 소요된다. 따라서, 대부분의 컴퓨팅 환경에 있어서, 추가적인 백업, 일명 증분적 백업은 연속적인 풀 백업, 예컨대, 매일의 풀 백업 중에 수행될 수 있다. 증분적 백업은 하나의 처리이므로 증분적 백업인지 풀 백업이던지간에 마지막 백업이 실행된 이후로 변화되는 데이터만이 백업된다. 일반적으로, 파일내의 많은 데이터가 자주 변경되지 않더라도 변경된 데

이터는 파일 기반으로 백업된다. 따라서, 증분적 백업은 풀 백업의 경우보다 작으므로 고속으로 달성된다. 대부분의 환경에서는 일반적으로 매주 한번씩 풀 백업을 실행하고, 증분적 백업은 일주일 동안 매일 실행하지만, 이러한 시간 프레임이 사용될 필요가 없다는 것을 인식해야 한다. 예컨대, 어떤 환경에서는 하루동안 몇번의 증분적 백업이 필요할 수 있다. 본 발명의 원리는 얼마나 자주 실행되는지와 무관하게 풀 백업(선택적인 증분적 백업)을 사용하는 모든 환경에 적용된다.

풀 백업 절차가 실행되는 동안, 호스트 컴퓨터는 복수의 데이터 파일로 이루어진 백업된 데이터를 포함하는 하나 이상의 가상 카트리지를 생성할 수 있다. 명확성을 위해, 후술되는 설명에서는 풀 백업이 단지 하나의 가상 카트리지를 생성하는 것으로 가정한다. 그러나, 풀 백업은 하나 이상의 가상 카트리지를 생성하고, 본 발명의 원리는 가상 카트리지의 개수에 한정되지 않는다는 것으로 인식되어야 한다.

제 1 실시형태에 의하면, 하나의 기존 풀 백업 데이터 세트와 하나 이상의 증분적 백업 데이터 세트로부터 종합 풀 백업 데이터 세트를 생성하는 방법이 제공된다. 이 방법은 주기적(예컨대, 매주) 풀 백업을 수행할 필요가 없기 때문에 유저의 상당한 시간과 네트워크 리소스를 절약할 수 있다. 또한, 당업자에게 자명한 바와 같이, 예컨대, 최근 버전의 파일이 증분적 백업에 존재하는 경우, 백업/복구 애플리케이션은 일반적으로 마지막 풀 백업에 의거한 파일을 복구하여 증분적 백업으로부터의 모든 변경을 적용하기 때문에 풀 백업에 의거한 복구 데이터와 하나 이상의 증분적 백업은 시간을 소비하는 처리가 될 수 있다. 따라서, 종합 풀 백업의 제공은 백업 복구 애플리케이션이 풀 백업과 하나 이상의 증분적 백업으로부터 중첩적으로 복구할 필요없이, 종합 풀 백업에만 의거하여 데이터 파일을 더 신속히 복구할 수 있게하는 추가적인 장점을 가질 수 있다. "가장 최근 버전"이라는 용어는 파일이 새로운 버전 번호를 갖는지와는 상관없이, 일반적으로 데이터 파일의 가장 최근 복사(즉, 데이터 파일이 저장된 가장 최근의 시간)를 의미하는 것으로 인식되어야 한다. "버전"이라는 용어는 몇가지 방법으로 수정될 수 있는, 또는 여러번 저장될 수 있는 동일한 파일의 복사를 의미한다.

도7은 종합 풀 백업 절차를 개략적으로 나타낸 도면이다. 호스트 컴퓨터(120)는 최초의 시간, 예컨대, 주말에 풀 백업(230)을 실행할 수 있다. 호스트 컴퓨터(120)는 계속적인 증분적 백업(232a, 232b, 232c, 232d, 232e)을 예컨대, 일주일 동안 매일 실행할 수 있다. 이어서, 스토리지 시스템(170)은 후술하는 바와 같이 종합 풀 백업 데이터 세트(234)를 생성할 수 있다.

제 1 실시형태에 의하면, 스토리지 시스템(170)은 종합 풀 백업 애플리케이션(240)(도3)으로서 여기서 언급된 소프트웨어 애플리케이션을 포함할 수 있다. 종합 풀 백업 애플리케이션(240)은 스토리지 시스템 제어기(122)(도2) 또는 호스트 컴퓨터(120)상에서 구동될 수 있다. 종합 풀 백업 애플리케이션은 종합 풀 백업 데이터 세트(234)의 생성에 필요한 소프트웨어 명령과 인터페이스를 포함한다. 일례에 있어서, 종합 풀 백업 애플리케이션은 풀 백업 데이터 세트(230)와 증분적 백업 데이터 세트(232) 각각의 메타데이터 표현의 논리적 병합을 수행하여 종합 풀 백업 데이터 세트(234)를 포함하는 새로운 가상 카트리지를 생성할 수 있다.

예컨대, 도8에 도시된 바와 같이, 기존 풀 백업 데이터 세트는 유저 파일(F1, F2, F3, 및 F4)을 포함할 수 있다. 제 1 증분적 백업 데이터 세트(232a)는 유저 파일 F2의 수정된 버전인 F2', 및 F3의 수정된 버전인 F3'을 포함할 수 있다. 제 2 증분적 백업 데이터 세트(232b)는 유저 파일 F1의 수정된 버전인 F1', F2의 더욱 수정 버전인 F2'', 및 새로운 유저 파일인 F5를 포함할 수 있다. 따라서, 종합 풀 백업 데이터 세트(234)는 풀 백업 데이터 세트(230)와 두개의 증분적 데이터 세트(232a, 232b)의 논리적 병합으로부터 형성되어 각 유저 파일(F1, F2, F3, F4, 및 F5)의 최종 버전을 포함한다. 따라서, 도8에 도시된 바와 같이, 종합 풀 백업 데이터 세트는 유저 파일 F1', F2'', F3', F4, 및 F5를 포함한다.

도3 및 도4에 도시된 바와 같이, 파일 시스템 소프트웨어(146)는 에뮬레이팅된 매체(134)에 저장된 각 유저 파일에 관한 메타데이터를 저장한 논리적 메타데이터 캐시(242)를 생성할 수 있다. 논리적 메타데이터 캐시는 물리적 데이터 캐시일 필요는 없지만, 대신에 스토리지 매체(126)에 저장된 데이터의 검색가능 컬렉션일 수 있다. 다른 예에 있어서, 논리적 메타데이터 캐시(242)는 데이터베이스로서 구현될 수 있다. 메타데이터가 데이터베이스에 저장된 경우, 종래 데이터베이스 명령(예컨대, SQL 명령)은 풀 백업 데이터 세트와 하나 이상의 증분적 백업 데이터 세트의 논리적 병합을 수행하여 종합 풀 백업 데이터 세트를 생성할 수 있다.

상기한 바와 같이, 에뮬레이팅된 매체(134)에 저장된 각 데이터 파일은 데이터 파일에 관련하여 메타데이터를 포함한 파일 디스크립터를 포함하고, 백업 스토리지 매체(126)상의 파일의 위치를 포함할 수 있다. 제 1 실시형태에 있어서, 호스트 컴퓨터(120)상에서 구동되는 백업/복구 애플리케이션은 에뮬레이팅된 매체(134)상에 스트리밍 테이프 포맷으로 데이터를 저장한다. 도9는 이 테이프 포맷을 나타낸 데이터 구조(250)의 예를 도시한 도면이다. 상기한 바와 같이, 시스템 파일 데이터 구조는 데이터 파일에 대한 파일 디스크립터, 파일의 생성 및/또는 수정일, 보안정보, 파일의 출처인 호스트 시스템의 디렉토리 구조뿐만 아니라 기타 가상 카트리지에 파일을 링크하는 정보와 같은 데이터 파일에 관한 정보를 가질 수 있

는 헤더를 포함한다. 이러한 헤더는 호스트 컴퓨터, 제 1 스토리지 시스템 등으로부터 백업(복사)된 실제 유저 및 시스템 파일인 데이터(254)와 관련된다. 시스템 파일 데이터 구조는 다음 헤더를 블럭 경계로 적절히 정렬할 수 있는 패드(256)를 선택적으로 포함할 수도 있다.

도9에 도시된 바와 같이, 제 1 실시형태에 있어서, 헤더 데이터는 논리적 메타데이터 캐시(242)에 배치되어 다른 일련의 테이프 데이터 포맷에 대한 빠른검색과 랜덤 엑세스를 가능하게 한다. 스토리지 시스템 제어기(122)상에 파일 시스템 소프트웨어(148)를 사용함으로써 구현된 논리적 메타데이터 캐시의 사용은 에뮬레이팅된 매체(134)에 저장된 선형, 일련의 테이프 데이터 포맷을 백업 스토리지 매체(126)를 구성하는 물리적 디스크상에 저장된 랜덤 액세스 데이터 포맷으로 변환할 수 있게 한다. 논리적 메타데이터 캐시(242)는 데이터 파일에 대한 파일 디스크립터를 포함하는 헤더(252), 데이터 파일로의 액세스를 제어하는데 사용될 수 있는 보안정보, 및 후술하는 바와 같은 포인터(256)를 가상 카트리지와 백업 스토리지 매체(126)상의 데이터 파일의 실제 위치에 저장한다. 제 1 실시형태에 있어서, 논리적 메타데이터 캐시는 풀 백업 데이터 세트(230)와 각 증분적 데이터 세트(232)에 백업된 모든 데이터 파일에 관한 데이터를 저장한다.

제 1 실시형태에 의하면, 종합 풀 백업 애플리케이션 소프트웨어(240)는 논리적 메타데이터 캐시에 저장된 정보를 사용하여 종합 풀 백업 데이터 세트를 생성한다. 이어서, 이 종합 풀 백업 데이터 세트는 종합 풀 백업 애플리케이션(240)에 의해 생성된 종합 가상 카트리지에 링크된다. 백업/복구 애플리케이션에 있어서 종합 풀 백업 데이터 세트는 이 종합 가상 카트리지상에 저장되는 것처럼 보인다. 상기한 바와 같이, 종합 풀 백업 데이터 세트는 기존의 풀 백업 데이터 세트와 증분적 백업 데이터 세트의 논리적 병합을 수행함으로써 생성될 수 있다. 이러한 논리적 병합은 각각의 기존 풀 백업 데이터 세트와 증분적 백업 데이터 세트에 포함된 각각의 데이터 파일의 비교, 및 도8을 참조하여 설명된 최종 수정된 버전의 각 유저 파일의 혼합의 생성을 포함할 수 있다.

제 1 실시형태에 의하면, 도10에 도시된 바와 같이, 종합 가상 카트리지(260)는 다른 가상 카트리지, 특히, 기존의 풀 백업 데이터 세트와 증분적 백업 데이터 세트를 포함한 가상 카트리지상의 데이터 파일의 위치를 포인팅하는 포인터를 포함한다. 상기 도8에 관하여 주어진 예를 고려하면, 종합 가상 카트리지(260)는 가상 카트리지(262)상의 기존 풀 백업 데이터 세트내의 유저 파일(F4)[기존 풀 백업 데이터 세트는 가장 최근 버전의 유저 파일(f4)을 포함하기 때문]의 위치와 예컨대, 가상 카트리지(264)상의 증분적 데이터 세트(232a)내의 유저 파일(F3')의 위치를 포인팅[화살표(268)로 표시됨]하는 포인터(266)을 포함한다.

종합 가상 카트리지는 포인터(266)가 포인팅하는 데이터를 포함하는 모든 가상 카트리지의 식별번호를 포함한 리스트(270)도 포함한다. 이 종속 카트리지 리스트(270)는 실제 데이터의 위치 추적과 종속 가상 카트리지가 삭제되는 것을 방지하기 위해 중요할 수 있다. 이 실시형태에 있어서, 종합 풀 백업 데이터 세트는 실제 유저 파일을 포함하지 않고, 백업 스토리지 매체(126)상의 유저 파일의 위치를 나타내는 포인터의 세트를 포함한다. 따라서, 실제 유저 파일(다른 가상 카트리지상에 저장된)의 삭제를 방지할 수 있다. 이것은 데이터를 포함한 가상 카트리지의 기록[종속 카트리지 리스트(270)]을 유지하고, 각 가상 카트리지의 덮어쓰기(over-written) 또는 삭제를 방지함으로써 부분적으로 달성될 수 있다. 종합 가상 카트리지는 종합 가상 카트리지의 크기, 백업 스토리지 매체(126)상의 종합 가상 카트리지의 위치와 같은 카트리지 데이터(272)를 포함할 수도 있다. 또한, 종합 가상 카트리지는 식별번호 및/또는 이름(274)을 가질 수 있다.

다른 실시형태에 의하면, 종합 가상 카트리지는 포인터와 실제 저장된 유저 파일의 조합을 포함할 수 있다. 도11에 도시된 바와 같이, 일례에 있어서, 종합 가상 카트리지는 가상 카트리지(262)상의 기존 풀 백업 데이터 세트(230)내의 데이터 파일(도9를 참조하여 설명한 바와 같은 가장 최근 버전)의 위치를 포인팅하는 포인터(266)를 포함한다. 종합 가상 카트리지는 화살표(280)로 표시된 증분적 데이터 세트(232)로부터 복사된 실제 데이터 파일을 포함하는 데이터(278)를 포함할 수도 있다. 이러한 방식으로, 증분적 백업 데이터 세트는 종합 풀 백업 데이터 세트(276)가 생성된 이후에 삭제될 수 있기 때문에 저장 공간이 절약된다. 상기 종합 가상 카트리지는 모든 유저 파일의 복사가 아닌 전체 또는 일부 포인터를 포함하는 종합 가상 카트리지에 비해 작다.

종합 풀 백업은 포인터와 저장된 파일 데이터의 조합을 포함하고, 상기 예에 한정되지 않는다는 것이 인식되어야 한다. 예컨대, 종합 풀 백업은 어떤 증분적 및/또는 풀 백업에 저장된 다수의 파일에 대한 데이터 파일에 대한 포인터를 포함할 수 있고, 다른 기존의 풀 및/또는 증분적 백업으로부터 복사되어 저장된 파일 데이터를 포함할 수 있다. 또한, 대안으로서, 종합 풀 백업은 어떠한 포인터도 포함하지 않고, 적합한 풀 및/또는 증분적 백업으로부터 복사된 가장 최근 버전의 실제 파일 데이터를 포함하는 모든 관련 증분적 백업 및 이전의 풀 백업에 의거하여 생성될 수 있다.

제 1 실시형태에 있어서, 종합 풀 백업 애플리케이션 소프트웨어는 각각의 기존 풀 백업 데이터 세트와 증분적 백업 데이터 세트에 대한 유저 및 시스템 파일 메타데이터를 비교하여 가장 최근 버전의 데이터 파일 각각이 위치된 곳을 결정할 수 있게 하는 디퍼런싱 알고리즘(differencing algorithm)을 포함할 수 있다. 예컨대, 디퍼런싱 알고리즘은 다른 백업 세트내

의 동일한 데이터 파일의 상이한 버전 사이에서 생성일 및/또는 수정일 등을 비교하여 가장 최근 버전의 데이터 파일을 선택하는데 사용될 수 있다. 그러나, 유저는 종종 파일내의 임의의 데이터를 실제로 변경하지 않고, 유저 파일을 열고, 파일을 저장할 수 있다(따라서, 그 수정의 테이퍼를 변경함). 따라서, 시스템은 시스템 또는 유저 파일내의 데이터를 분석하여 데이터가 실제로 변경되었는지를 결정할 수 있는 더욱 향상된 디퍼런싱 알고리즘을 구현할 수 있다. 이러한 디퍼런싱 알고리즘의 변형과 다른 형태의 비교 알고리즘은 당업자에게 자명할 것이다. 또한, 상기한 바와 같이, 메타데이터가 데이터베이스 포맷으로 저장되는 경우, SQL 명령 등의 데이터베이스 명령은 논리적 병합을 수행하는데 사용될 수도 있다. 본 발명은 종합 풀 백업 데이터 세트를 정확히 생성할 수 있도록 가장 최근 또는 최종 버전의 각 유저 파일이 전체 비교된 기존 백업 세트로부터 선택될 수 있게 하는 모든 알고리즘에 적용될 수 있다.

당업자에게 자명한 바와 같이, 종합 풀 백업 애플리케이션은 호스트 컴퓨터가 물리적 풀 백업을 실행할 필요없이 풀 백업 데이터 세트가 생성되고 이용가능하게 한다. 데이터를 백업 스토리지 시스템으로 전송하는 프로세서 부담으로 인해 호스트 컴퓨터에 부담을 주지 않도록 하는 것뿐만 아니라, 종합 풀 백업 애플리케이션이 스토리지 시스템에서 실행되는 실시형태에 있어서, 네트워크 대역폭의 활용을 상당히 감소시킨다. 도7에 도시된 바와 같이, 제 1 종합 풀 백업 데이터 세트(234)와 일련의 증분적 백업 데이터 세트(236)를 사용하여 추가적인 종합 풀 백업 데이터 세트가 생성된다. 이것은 빈번히 수정되지 않고 빈번히 복사되지 않을 파일 또는 대상에 상당한 시간적 이익을 제공할 수 있다. 그 대신, 종합 풀 백업 데이터 세트는 단지 1회 복사된 파일에 대한 포인터를 유지할 수 있다.

도3을 참조하여 상기한 바와 같이, 스토리지 시스템은 엔드 유저 복구 애플리케이션(300)으로서의 소프트웨어 애플리케이션을 포함할 수도 있다. 따라서, 다른 실시형태에 의하면, 엔드 유저가 IT 스태프의 간섭없이, 기존 백업/복구 절차 및/또는 정책을 변경할 필요없이 백업 데이터를 찾아서 복구하는 방법이 제공된다. 전형적인 백업 스토리지 시스템에 있어서, 호스트 컴퓨터(120)상에서 구동되는 백업/복구 애플리케이션은 IT 스태프에 의해 제어되고, 엔드 유저가 IT 스태프에 의한 간섭없이 백업된 데이터에 엑세스하는 것은 불가능하거나 매우 어려울 수 있다. 본 발명의 실시형태에 의하면, 스토리지 시스템 소프트웨어는 엔드 유저가 예컨대, 백업 스토리지 매체(126)와의 웹 기반 또는 다른 인터페이스를 통해 자신의 파일을 찾아서 복구하는 것을 제공한다.

종합 풀 백업 애플리케이션(240)과 마찬가지로 엔드 유저 복구 애플리케이션(300)은 스토리지 시스템 제어기(122) 또는 호스트 컴퓨터(120)상에서 구동될 수 있다는 것이 인식되어야 한다. 엔드 유저 복구 애플리케이션은 인증된 유저가 논리적 메타데이터 캐시를 검색하여 백업 스토리지 매체(126)로부터 백업된 파일을 찾고, 선택적으로 복구하게 하는데 필요한 소프트웨어 명령과 인터페이스를 포함한다.

제 1 실시형태에 의하면, 유저 컴퓨터(136)상에서 설치 및/또는 실행되는 유저 인터페이스를 포함하는 소프트웨어가 제공된다. 유저 인터페이스는 유저가 백업 스토리지 매체상의 파일을 찾게 하는 모든 형태의 인터페이스가 될 수 있다. 예컨대, 유저 인터페이스는 그래픽 유저 인터페이스, 웹 기반, 또는 텍스트 인터페이스 등이 될 수 있다. 유저 컴퓨터는 예컨대, 이더넷 연결과 같은 네트워크 연결(138)을 통해 스토리지 시스템(170)에 연결된다. 이 네트워크 연결(138)을 통해 유저 컴퓨터(136)의 오퍼레이터는 스토리지 시스템(170)에 저장된 데이터에 액세스 가능하다.

일례에 있어서, 엔드 유저 복구 애플리케이션(300)은 유저 인증 및/또는 인증 특징을 포함한다. 예컨대, 유저는 유저명과 패스워드를 사용하는 유저 컴퓨터상의 유저 인터페이스를 통해 로그인을 요청받을 수 있다. 유저 컴퓨터는 적절한 유저 입증 메카니즘을 사용하여 유저가 스토리지 시스템으로 액세스 했는지를 결정할 수 있는 스토리지 시스템(예컨대, 엔드 유저 복구 애플리케이션)으로 유저명과 패스워드를 전송할 수 있다. 유저 입증 메카니즘에 포함될 수 있지만 이것에 한정되지 않는 몇가지 예로서는 Microsoft Active Directory server, Unix "yellow pages" server, 또는 Lightweight Directory Access Protocol 등이 있다. 로그인/유저 입증 메카니즘은 엔드 유저 복구 애플리케이션과 통신하여 유저 권한을 전환할 수 있다. 예컨대, 몇몇 유저는 자기가 생성한 파일만을 검색할 수 있게 될 수도 있고, 또는 소정의 권한을 갖거나, 오너(owner)로서 식별될 수도 있다. 예컨대, 시스템 오퍼레이터 또는 관리자와 같은 다른 유저들은 백업된 파일 전체에 대하여 액세스가 허용될 수도 있다.

제 1 실시형태에 의하면, 엔드 유저 복구 애플리케이션은 논리적 메타데이터 캐시를 사용하여 백업 스토리지 매체상에 백업된 전체 데이터 파일에 대한 정보를 얻는다. 엔드 유저 복구 애플리케이션은 예컨대, 백업시간, 백업날짜, 유저이름, 오리지널 유저 컴퓨터 디렉토리 구조(파일이 백업된 경우에 얻어질 수 있음), 또는 다른 파일 특성 등에 의해 분류된 유저 파일의 계층적 디렉토리 구조를 유저 인터페이스를 통해 유저에게 제공한다. 일례에 있어서, 유저에게 제공되는 디렉토리 구조는 유저에게 주어진 권한에 따라 바뀔 수 있다. 엔드 유저 복구 애플리케이션은 브라우징 요청(즉, 유저 인터페이스를 통해 유저가 디렉토리 구조를 브라우징하여 원하는 파일을 찾음)을 받거나, 유저가 이름, 날짜 등에 의해 파일을 검색할 수 있다.

제 1 실시형태에 의하면, 유저는 스토리지 시스템으로부터 백업된 파일을 복구할 수 있다. 예컨대, 유저가 원하는 파일을 찾으면, 상기한 바와 같이, 유저는 네트워크 연결(138)을 통해 스토리지 시스템으로부터 상기 파일을 다운로드할 수 있다. 일례에 있어서, 이러한 다운로드 절차는 당업자에게 공지된 바와 같이, 웹 기반 다운로드에 필적하는 방식으로 구현될 수 있다.

뷰잉/다운로드에 관한 허가를 가진 엔드 유저가 파일에 액세스 가능하게 하고, 유저 인터페이스를 통해 이 액세스를 가능하게 함으로써 엔드 유저 복구 애플리케이션은 유저가 자신의 파일을 백업 정책 또는 절차를 변경하지 않고 검색 및 복구하게 할 수 있다.

다른 실시형태에 의하면, 유저가 백업 스토리지 매체(126)상에 저장된 백업 데이터 세트의 뷰(view)가 부착된 네트워크를 "마운팅"할 수 있는 방법 및 메카니즘이 제공된다. 이것은 유저가 자신의 컴퓨터에 연결된 임의의 다른 로컬 또는 네트워크 드라이브상의 데이터를 뷰잉하고 액세스하는 것과 마찬가지로, 마운팅된 데이터 세트내의 데이터를 뷰잉하고 액세스할 수 있게 한다. 따라서, 예컨대, 유저는 매체 서버(114)(도1)를 통한 복구 처리를 실행하지 않고 애플리케이션 서버[예컨대, 시스템의 제 1 스토리지 장치(106)(도1)가 실패한 경우]로 데이터를 유효하게 복구할 수 있다. 상기한 바와 같은 마운팅 절차를 사용한 애플리케이션 서버로의 데이터의 복구는 볼륨 복구가 용이한 전형적인 매체 서버에 비해 수십 배 빠른 속도로 이루어질 수 있다. "마운팅"이란 용어는 네트워크 드라이브 등의 네트워크 구성요소 또는 데이터 볼륨을 호스트 컴퓨터의 운영 시스템에서 이용가능하게 하는 것을 의미하는 것으로 인식되어야 한다. 데이터 볼륨은 예컨대, 단일 데이터 파일 또는 시스템 파일, 복수의 파일, 또는 복수의 파일을 포함하는 디렉토리 구조 등을 포함할 수 있다. 커먼 마운팅 프로토콜(common mounting protocol)은 NFS(network file system) 또는 CIFS(common internet file system) 쉐어링(sharing)을 포함한다. 이러한 프로토콜은 호스트 컴퓨터가 리모트 리소스(remote resource)가 호스트 컴퓨터상에 국부적으로 wprhd되는 것으로 보여지게 하는 인터페이스를 통해 네트워크 연결을 거쳐 다른 컴퓨터상의 리소스에 액세스할 수 있게 한다.

도12는 본 발명의 제 1 실시형태에 의한 볼륨 마운트를 수행하는 방법을 나타낸 순서도이다. 제 1 단계(290)에 있어서, 유저는 데이터 볼륨을 선택하여 마운팅하고 백업 스토리지 시스템 제어기(122)에 볼륨 마운트 요청을 전달한다(도3). 일반적으로, 유저는 백업된 정보의 전체적인 및 정확한 표현을 캡쳐할 수 있도록 풀 백업 데이터 세트(증분적 백업 데이터 세트가 아닌)로부터 데이터를 복구하기를 원할 수 있다. 현재 풀 백업 데이터 세트가 존재하지 않는 경우(예컨대, 네트워크 매니저가 풀 백업을 매주 실행함에 따라 유저가 주중에 데이터를 복구하길 원해도 현재 풀 백업을 이용할 수 없는 경우), 종합 풀 백업이 생성되어 선택된 데이터의 복구에 사용될 수 있다.

제 1 실시형태에 의하면, 백업 스토리지 시스템(170)은 데이터 볼륨 마운트와 복구 절차를 수행하는 방법을 제어하고 구현할 수 있는 볼륨 복구 애플리케이션(310)(도13)인 소프트웨어 애플리케이션을 포함할 수 있다. 종합 풀 백업 및 엔드 유저 복구 애플리케이션과 유사한 볼륨 복구 애플리케이션(310)은 호스트 컴퓨터 및/또는 유저 컴퓨터상에서 실행될 수 있고, 그 일부는 스토리지 시스템 제어기, 호스트 컴퓨터, 및 유저 컴퓨터의 전체 또는 일부에 분배될 수 있다.

상기 도12를 다시 참조하면, 볼륨 마운팅이 요청된 후, 볼륨 복구 애플리케이션은 현재의 풀 백업 데이터 세트가 이용 가능한지를 조회할 수 있다(단계 292). 이용이 불가한 경우, 볼륨 복구 애플리케이션은 종합 풀 백업 애플리케이션(240)과 통신하여 종합 풀 백업 처리를 수행하고, 현재의 백업 데이터 세트를 생성할 수 있다(단계 294). 볼륨 복구 애플리케이션은 정규 풀 백업 데이터 세트 또는 종합 풀 백업 데이터 세트를 익스포팅(exporting)하고, 요청된 볼륨 마운팅을 NFS 또는 CIFS 쉐어에 따라 수행할 수 있다. 특히, 볼륨 복구 애플리케이션은 논리적 메타데이터 캐시(242)를 조회하여 단계 290에서 식별되어 선택된 풀 백업 볼륨을 나타내는 적절한 메타데이터를 찾는다.

제 1 실시형태에 의하면, 마운트 요청(단계 290)은 볼륨 복구 애플리케이션이 하나 이상의 파일 디스크립터 구조를 생성하여 NFS 또는 CIFS 쉐어에 따른 마운팅에 대한 볼륨의 익스포팅을 용이하게 한다(단계 296). 도14는 볼륨 복구 애플리케이션에 의해 생성될 수 있는 파일 디스크립터 구조(320)의 제 1 실시형태를 나타낸 도면이고, 파일 디스크립터(320)는 테이프 포맷에 있어서 시스템 파일[예컨대, 시스템 파일(322), 도15 참조]에 대응된다. 상기한 바와 같이, 파일 디스크립터는 스토리지 시스템에 저장된 시스템 파일과 데이터 파일에 대응하는 검색가능한 메타데이터를 포함한다. 파일 디스크립터(320)는 예컨대, 마운팅될 볼륨에 포함된 데이터 파일에 대한 파일 퍼미션(액세스 제어 파일)(324)과 파일명(322) 등의 정보를 포함하는 복수의 필드를 포함할 수 있다. 또한, 파일 디스크립터는 데이터 파일의 소스 데이터의 위치(즉, 스토리지 매체(126)상에 데이터 파일이 저장된 위치를 식별하기 위한), 데이터 파일의 길이(328)에 대한 하나 이상의 포인터(326), 및 링크된 리스트 파일 디스크립터 구조내의 다음 엔트리(entry)(예컨대, 다음 데이터 파일)에 대한 포인터(330)를 포함한다. 예컨대, 참조번호 331에 의해 표시된 "다음" 필드가 널(null)인 경우, 데이터 파일이 파일 디스크립터(320)에 의해 표현된 시스템 파일에 알려진 가장 최근 데이터 파일(예컨대, 가장 최근 링크된 리스트 엔트리임)이라는 것을 나타낸다. 마운팅

될 데이터 볼륨내에 포함된 각 시스템 파일은 도14에 도시된 바와 같은 파일 디스크립터 구조에 의해 표현된다. 요청된 볼륨내의 각 시스템 파일이 생성된 파일 디스크립터(320)를 갖는 경우, 파일 디스크립터는 NFS 또는 CIFS 요청에 대답하는 관련 데이터 파일을 찾아 익스포팅하는데 사용될 수 있다.

상기한 바와 같이, 제 1 실시형태에 있어서, 파일 디스크립터는 대부분의 유닉스 기반 컴퓨터 시스템에 사용되는 테이프 아카이브(타르) 포맷 등의 표준화된 포맷에 따라 구현될 수 있다. 도15는 테이프(예컨대, 타르) 데이터 스트림의 세그먼트에 따른 테이프 포맷으로 기록된 전형적인 시스템 파일(332)을 나타낸 도면이다. 도16은 시스템 파일(332)에 대한 대응 파일 디스크립터(340)를 나타낸 도면이다. 도15에 도시된 바와 같이, 테이프 포맷으로 기록된 파일은 시스템 파일(332)에 저장된 실제 데이터(338)와 헤더(336)를 포함한다. 데이터(338)는 하나 이상의 데이터 파일에 대응할 수 있다. 도시된 예에 있어서, 시스템 파일(332)의 길이는 1032 바이트이지만, 상기 파일은 파일의 크기와 기록된 포맷에 따라 임의의 길이를 가질 수 있다.

파일(332)에 대한 파일 디스크립터(340)는 헤더(336)에 포함된다. 도16에 도시된 바와 같이, 그리고, 도14에 도시된 일반적인 예와 마찬가지로, 파일 디스크립터(340)는 파일명(341), 보안정보(344), 시스템 파일에 알려진 각 데이터의 저장된 데이터에 대한 포인터(342), 대응 데이터 파일의 길이(346), 및 도시된 예에 있어서 널(null)(348)인 시스템 파일에 알려진 다음 데이터 파일을 식별하는 "다음" 엔트리를 포함한다.

상기 도12를 다시 참조하면, 마운팅된 데이터 볼륨내의 파일에 대한 전체 파일 디스크립터가 생성된 경우, 볼륨 복구 애플리케이션은 생성된 파일 디스크립터에 의거한 파일 시스템을 유저가 특정한 마운트 포인트에 NFS 또는 CIFS 쉐어에 따라 익스포팅한다(단계 298). 이 포인트에서, 마운트가 완료되고(단계 299), 마운팅된 데이터 볼륨은 후술하는 바와 같이, 유저가 데이터를 판독 및/또는 기록하는데 이용가능하다.

제 1 실시형태에 의하면, NFS 또는 CIFS 판독 동작[즉, 유저가 마운팅된 데이터 볼륨내의 데이터를 뷰잉(viewing)하기 원함]은 파일 특정을 매칭시키기 위한 파일 디스크립터를 통해 검색함으로써 서비스된다. 제 1 실시형태에 의하면, 유저는 자신이 직접 파일 디스크립터를 실제로 검색할 필요가 없다는 것이 인식되어야 한다. 대신, 볼륨 복구 애플리케이션은 예컨대, 전형적인 디렉토리 구조 포맷내에서 유저에게 데이터를 제공하는 유저 인터페이스를 포함할 수 있다. 볼륨 복구 애플리케이션은 특정 파일에 대한 유저 요청을 논리적 메타데이터 캐시에 액세스하는 검색 명령으로 변환하고, 매칭 시스템 파일에 대한 파일 디스크립터(320)를 검색하는 소프트웨어를 포함할 수 있다. 파일이 찾아지는 경우, 유저 컴퓨터로의 데이터 전송은 링크된 리스트를 폴로우잉(following)[즉, 실제 데이터를 찾기 위해 파일 디스크립터에 저장된 포인터를 폴로우잉]함으로써 달성되어 요청한 유저로 보내질 수 있는 파일 데이터를 위해 버퍼를 생성한다.

다른 실시형태에 의하면, 또한, 유저가 마운팅된 볼륨에 새로운 데이터를 기록하는 것을 위해서 메카니즘이 제공될 수 있다. 상기한 바와 같이, 마운팅된 볼륨 데이터는 유저에게 보통의 네트워크 드라이브 또는 다른 네트워크-저장된 데이터로서 보여질 수 있다. 그러나, 실제로는, 오리지널 마운팅된 볼륨 데이터는 일반적으로 적어도 다른 백업 데이터 세트가 생성될 때까지는 보호될 필요가 있는 실제 백업 데이터이다. 따라서, 유저가 오리지널 백업 데이터를 실제로 수정할 수 있게 하는 것은 바람직하지 않을 수 있다. 유저가 마운팅된 볼륨에 대응하는 데이터를 수정할 수 있게 되어 있는 동안 백업 데이터의 수정을 방지하기 위해, 후술되는 바와 같이, 다른 스토리지 매체로의 기록으로 전환하는 메카니즘이 제공된다.

도17은 본 발명의 제 1 실시형태에 의한 기록 요청을 처리하는 방법을 나타낸 순서도이다. 첫번째 단계(350)에 있어서, 유저는 NFS 또는 CIFS 기록 동작(일반적으로 데이터 파일을 에디팅 또는 뷰잉하는 동안 "저장" 옵션을 선택하는 것에 의함)을 요청한다. 볼륨 복구 애플리케이션은 이용가능한 저장 공간을 찾고, 그 공간에 데이터를 기록하고, 및 새로 기록된 데이터를 참조하기 위한 적합한 파일 디스크립터를 업데이트함으로써 기록 요청을 실행한다.

제 1 실시형태에 의하면, 볼륨 복구 애플리케이션은 데이터를 기록하기 위한 저장 공간이 이미 할당되었는지의 여부를 조회하고(단계 352), 할당되어 있지 않은 경우, 저장 공간을 할당한다(단계 354). 저장 공간은 백업 스토리지 매체(126)에 할당될 수 있다(도13). 할당된 저장 공간은 기록 데이터만을 홀딩하기 위해 특별히 표시될 수 있다(관련된 메타데이터는 선택적임).

도18은 백업 스토리지 매체(126)에 저장된 NFS 또는 CIFS 기록 데이터의 일례를 나타낸 도면이다. 기록된 데이터(360)는 볼륨 복구 애플리케이션에 의해 서비스된 기록 명령의 결과로서 발생한 저장된 데이터에 대응하는 예컨대, 두개의 기록된 부분인 W1(362), W2(364)를 포함한다. 예컨대, W1 및 W2는 마운팅된 데이터 볼륨내에 포함된 수정된 데이터 파일에 대응할 수 있다. 두개의 기록 요청에 대응하여 도시되긴 했지만, 본 발명의 원리는 기록 요청의 수에 한정되지 않고 적용될 수 있으며, 파일은 기록 요청의 수에 따라 적합하게 변경될 수 있다는 것이 인식되어야 한다. 기록된 데이터(360)는 오리지

널 데이터[예컨대, 파일(332)]와 새로 기록된 데이터(360) 사이의 자기 표시 관계를 형성하는 메타데이터를 포함하는 헤더(366)도 포함한다. 특히, 도19를 더 참조하여 후술하는 바와 같이, 헤더는 기록된 데이터 부분(W1, W2)이 오리지널 데이터와 관련하여 논리적으로 존재하는 곳을 나타내는 오프셋 정보를 포함할 수 있다.

도19는 두개의 기록 요청이 서비스된 이후의 시스템 파일 레이아웃의 일례를 나타낸 도면이다. 오리지널 시스템 파일(332)은 백업 스토리지 매체(126)(도13)에 저장되고, 상기한 마운팅 절차를 통해 유저에게 제공된다. 도19에 도시된 시스템 파일(332)은 데어터 포맷내에 있고, 데이터 부분(338)은 복수의 데이터 파일(예컨대, 유저 파일)을 포함할 수 있다. 데이터는 오프셋 제로 바이트[포인트(370)]에서 시작되고, 나중에 포인트(372)에서 종료된다. 기록된 파일(360)은 유저의 요청에 대응하여 파일(332)에 데이터를 기록한다. 예컨대, 유저는 시스템 파일(332)내에 포함된 두개의 데이터 파일을 수정할 수 있고, 결과적으로, 기록된 파일(360)은 W1 및 W2를 포함한다. 상기한 바와 같이, 이 기록된 파일(360)은 오리지널 백업 데이터를 변경하지 않도록 스토리지 매체상의 파일(332)로부터 분리되어 저장될 수 있다. 논리적으로 수정된 시스템 파일(380)이 도시되고, 기록 요청을 통해 사용자에 의한 변경[즉, 기록된 파일(360)]을 포함하는 파일(332)을 나타낸다. 즉, 수정된 시스템 파일(380)에 있어서, W1 및 W2(사용자 수정 데이터 파일)는 백업된 데이터를 제거하지 않고 오리지널 시스템 파일(332)의 데이터 부분내에 포함된 오리지널 데이터 파일을 대체하는데 사용될 수 있다.

도19에 도시된 바와 같이, 수정된 시스템 파일은 오리지널 시스템 파일(332)과 기록된 파일(360)의 논리적 병합(summation)에 대응한다. 도시된 바와 같이, 오리지널 시스템 파일 데이터(338)는 오리지널 파일내의 오프셋 제로에서 시작된다. 오프셋 64(참조번호 384)에서, 수정된 데이터의 제 1 부분(W1)이 시작하고, 9바이트가 추가된 오프셋 73(참조번호 386)에서 종료된다. 따라서, 유저의 기록 요청에 의한 유저 수정 데이터 파일인 W1은 오리지널 시스템 파일(332)내의 오프셋 64에 위치된 오리지널 데이터 파일을 대체하는데 사용될 수 있다. W1은 기록된 파일(360)내의 오프셋 제로(390)로부터 존재하고, 기록된 파일(360)내의 오프셋 9(392)에서 종료하기 때문에 W1의 길이는 9바이트가 된다. 수정된 파일내 W1의 시작 위치(도시된 예에 있어서 오프셋 64)는 헤더(366)에 저장된 정보, 즉, 기록된 파일(360)과 오리지널 파일(332) 사이의 상대적 관계에 의해 결정된다. W2 부분도 수정된 파일(380)내에 포함되고, 오프셋 1032(파일의 오리지널 엔드, 참조번호 372)에서 시작하고, 논리적으로 파일을 100 바이트 연장한다. 또한, W2의 길이는 헤더(366)에 위치된 정보로부터 결정된다. 파일의 새로운 종료 포인트는 참조번호 388로 표시된다.

수정된 파일은 논리적으로 생성되고, 유저 수정 버전의 오리지널 파일로 표현되지만, 파일(360)에 의해 표현된 새로 기록된 데이터는 오리지널 파일(332)의 일부로서 실제로 저장되지 않는다. 대신, 상기한 바와 같이, 새로 기록된 데이터는 데이터를 기록하기 위해 식별된 스토리지 매체상의 특정 위치에 저장된다. 이러한 방식으로, 일반적인 로컬 또는 네트워크 드라이브에서와 마찬가지로, 유저가 마운팅된 볼륨에 외관상 기록이 가능한 반면, 오리지널 백업 데이터의 보전이 유지된다.

수정된 파일(380)은 수정된 파일을 나타내는 파일 디스크립터를 포함한 헤더(382)를 포함한다. 도20은 이러한 파일 디스크립터(400)의 예를 나타낸 도면이다. 파일 디스크립터(400)는 수정된 파일(380)의 파일명을 식별하는 이름 필드(name field)(402), 및 수정된 파일(380)의 허용 속성을 식별하는 보안 필드(security field)(404)를 포함한다. 파일 디스크립터(400)는 각각의 오리지널 파일과 기록된 파일에 저장된 데이터를 캡쳐하기 위한 오리지널 파일(332)에 대한 포인터, 및 기록된 파일(360)에 대한 포인터를 포함하는 복수의 데이터 필드도 포함한다. 파일 디스크립터(400)에 주어진 포인터의 링크된 리스트를 연속적으로 폴로우잉함으로써, 수정된 파일(380)의 표시가 주어진다.

도19 및 도20에는 수정된 파일에 대한 파일 디스크립터의 일례가 도시되고 설명되어 있다. 제 1 데이터 필드(406)에 있어서, 도19에서 참조번호 408로 식별되는 오프셋 제로 바이트에 있는 수정된 파일(380)내의 제 1 데이터 파일 위치에 대한 포인터가 위치한다. 폴로우잉 필드(410)는 포인터(406)에 의해 위치가 특정된 데이터 파일의 길이를 표시한다. 도시된 예에 있어서, 도19에서 볼 수 있는 바와 같이, 길이는 64 바이트이다[제로 오프셋 포인트(408)와 64 바이트의 오프셋(384) 사이에서 데이터가 연장됨]. 다음 필드(412)는 도19에 도시된 바와 같이 수정된 파일(380)내의 다음 데이터 파일이 W1이라는 것을 표시한다. 따라서, 포인터(414)는 W1에 대응한 데이터의 위치는 제로 오프셋 포인트(390, 도19)에서 새로 기록된 파일(360)에 저장된다는 것을 표시한다. 길이 필드(416)는 도19에서 볼 수 있는 바와 같이, W1은 수정된 파일(380)내에서 오프셋 64(384)와 오프셋 73(386) 사이에서 연장되어 W1의 길이가 9 바이트라는 것을 표시한다. 다음 필드(418)는 수정된 파일(380)내의 다음 데이터 파일이 오리지널 시스템 파일(332)로부터의 데이터 파일이라는 것을 표시한다. 필드(420)내의 포인터는 다음 데이터 파일이 수정된 파일(380)내의 오프셋 73(도19의 참조번호 386)에 위치된다는 것을 표시한다. 필드(422)는 도19에 도시된 바와 같이, 데이터 파일의 길이가 959 바이트라는 것을 표시한다. 다음 필드(424)는 폴로우잉 데이터 파일이 W2라는 것을 표시한다. 또한, 필드(426)내의 포인터는 W2의 위치, 즉, 도19에 도시된 바와 같이, 새로 기록된 파일(360)의 오프셋 9를 표시한다. 필드(428)는 W2의 길이가 100 바이트라는 것, 다음 필드(430)는 널을 포함한다는 것, 도19에 도시된 바와 같이, W2가 수정된 파일(380)내의 최종 데이터 파일이라는 것을 표시한다. 따라서, 파일 디스크립터(400)는 수정된 파일(380)의 구조, 및 수정된 파일(380)에 포함된 데이터의 위치를 나타내는 "로드맵(roadmap)"을 포함한다.

상기한 볼륨 복구 애플리케이션, 및 방법은 일련의 테이프 포맷 데이터를 NFS 또는 CIFS 등의 랜덤 액세스 I/O 시스템에 적합한 형태로 표시한다. 파일 디스크립터(400)와 같은 링크된 리스트 파일 디스크립터는 특정 타르 스트림(tar stream) 내의 각 데이터 파일의 스토리지 매체상의 위치와 더불어 예컨대, 타르 스트림내의 다른 데이터 파일에 관련된 타르 스트림내의 각 데이터 파일의 위치를 기록함으로써 일련의 테이프 포맷 데이터를 랜덤 액세스가 가능한 데이터로 변환하는데 사용될 수 있다. 또한, 제 1 실시형태에 의하면, 볼륨 복구 애플리케이션은 백업/복구 애플리케이션이 상기한 보통의 방식으로 데이터에 액세스할 수 있도록, 변경된(즉, 기록된) 데이터 백을 테이프(예컨대, 타르) 포맷으로 표시하는 프로비젼(provision)을 포함할 수 있다. 제 1 실시형태에 의하면, 인스턴트 복구 애플리케이션은 파일 시스템 소프트웨어와 관련하여 상기한 방식으로 테이프 헤더, 패드, 데이터 및 파일 마커로 적절히 포맷된 가상 카트리지를 생성하는 설비를 포함한다. 다른 실시형태에 있어서, 볼륨 복구 애플리케이션은 파일 시스템 소프트웨어와 인터페이싱되어 새로 기록된 및 수정된 파일을 포함하는 상기한 바와 같은 가상 카트리지를 생성할 수 있다.

본 발명에 있어서, 종합 풀 백업 애플리케이션, 엔드 유저 복구 애플리케이션, 및 볼륨 복구 애플리케이션과 같은 소프트웨어의 용어가 주로 사용되었지만, 소프트웨어, 하드웨어 또는 펌웨어, 또는 그 조합으로 다른 형태가 선택적으로 구현될 수 있다는 것이 인식되어야 한다. 따라서, 본 발명의 실시형태는 스토리지 시스템의 프로세서에서 적어도 일부가 실행되어 상기한 바와 같은 종합 풀 백업 애플리케이션 및/또는 엔드 유저 복구 애플리케이션의 기능을 수행하는 경우, 컴퓨터 프로그램으로 엔코딩된 모든 컴퓨터 판독가능 매체(예컨대, 컴퓨터 메모리, 플로피 디스크, 콤팩트 디스크, 테이프 등)를 포함할 수 있다.

요컨대, 본 발명에 의한 실시형태는 종래의 테이프 백업 시스템을 에뮬레이팅하지만, 엔드 유저가 백업된 파일을 뷰잉 또는 복구하게 하고, 종합 백업을 생성할 수 있는 것과 같은 향상된 기능성을 제공할 수 있는 스토리지 시스템, 및 방법을 포함한다. 그러나, 본 발명에 의한 다양한 형태는 컴퓨터 데이터의 백업 이외에 사용될 수 있다. 본 발명에 의한 스토리지 시스템은 저장된 데이터가 하드 디스크 액세스 시간에 있어서, 연속적이지 않고 랜덤하게 액세스될 수 있는 대용량 데이터를 경제적으로 저장하는데 사용될 수 있고, 본 발명에 의한 실시형태는 종래 백업 스토리지 시스템 이외의 사용을 찾을 수 있다. 예컨대, 본 발명에 의한 실시형태는 영화와 음악의 폭넓은 선택을 의미하는 주문형 비디오 및/또는 주문형 오디오가 가능한 비디오 또는 오디오 데이터를 저장하는데 사용될 수 있다.

본 발명의 하나 이상의 실시형태의 몇가지 양상에 대한 상세한 설명에 의해 당업자는 다양한 변형, 수정, 및 개량을 할 수 있는 것이 인식되어야 한다. 이러한 변형, 수정, 및 개량은 이 상세한 설명의 일부로서 의도되었고, 본 발명의 사상내에서 의도된 것이다. 따라서, 상기 설명과 도면은 예시만을 위한 것이다.

(57) 청구의 범위

청구항 1.

백업 스토리지 시스템에 저장된 가장 최근 백업된 버전의 하나 이상의 데이터 파일에 대응하는 데이터 파일을 하나 이상 포함하는 데이터 볼륨을 호스트 컴퓨터상에 마운팅하는 단계, 및

상기 가장 최근 백업된 버전의 하나 이상의 데이터 파일을 보존하는 동안, 상기 백업 스토리지 시스템에 저장된 상기 가장 최근 백업된 버전의 하나 이상의 데이터 파일보다 더 최근의 제 2 버전의 하나 이상의 데이터 파일에 대응하는 데이터를 상기 백업 스토리지 시스템에 저장하는 단계를 포함하는 것을 특징으로 하는 방법.

청구항 2.

제 1 항에 있어서,

상기 가장 최근 백업된 버전의 하나 이상의 데이터 파일과 상기 제 2 버전의 하나 이상의 데이터 파일을 링크하는 단계를 더 포함하는 것을 특징으로 하는 방법.

청구항 3.

제 1 항에 있어서,

상기 가장 최근 백업된 버전의 하나 이상의 데이터 파일과 상기 제 2 버전의 하나 이상의 데이터 파일을 식별하는 데이터 구조를 생성하는 단계를 더 포함하는 것을 특징으로 하는 방법.

청구항 4.

제 3 항에 있어서,

상기 제 2 버전의 하나 이상의 데이터 파일은 상기 가장 최근 백업된 버전의 하나 이상의 데이터 파일의 수정된 버전인 것을 특징으로 하는 방법.

청구항 5.

제 1 항에 있어서,

상기 데이터 볼륨을 마운팅하는 단계는 NFS 마운팅과 CIFS 마운팅 중 어느 하나의 수행을 포함하는 것을 특징으로 하는 방법.

청구항 6.

제 1 항에 있어서,

상기 데이터 볼륨을 마운팅하는 단계는 상기 가장 최근 백업된 버전의 하나 이상의 데이터 파일에 관련된 메타데이터를 포함하는 파일 디스크립터의 생성을 포함하고, 상기 메타데이터는 상기 가장 최근 백업된 버전의 하나 이상의 데이터 파일의 백업 스토리지 매체상의 저장 위치를 식별하는 식별자를 포함하는 것을 특징으로 하는 방법.

청구항 7.

백업 데이터 세트를 저장하기 위한 백업 스토리지 매체, 및

제 1 항의 방법을 구현하는 명령의 세트를 실행하도록 구성된 하나 이상의 프로세서를 포함하는 제어기를 포함하는 것을 특징으로 하는 백업 스토리지 시스템.

청구항 8.

제 7 항에 있어서,

상기 백업 데이터 세트는 종합 풀 백업 데이터 세트인 것을 특징으로 하는 백업 스토리지 시스템.

청구항 9.

하나 이상의 프로세서에서 실행시 제 1 항의 방법을 구현하는 복수의 명령으로 엔코딩된 것을 특징으로 하는 컴퓨터 판독 가능 매체.

청구항 10.

제 9 항에 있어서, 상기 프로세서는 백업 스토리지 시스템에 포함되는 것을 특징으로 하는 컴퓨터 판독가능 매체.

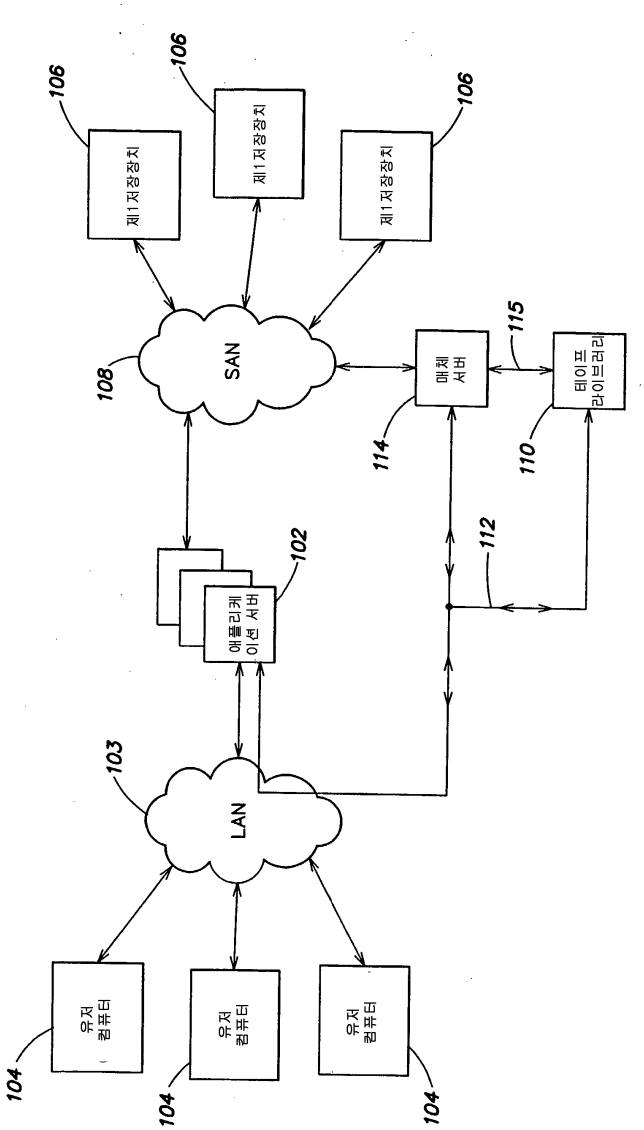
청구항 11.

하나 이상의 데이터 파일을 포함하는 백업 데이터 세트에 대응하는 시스템 파일을 독자적으로 식별하는 제 1 식별자, 및

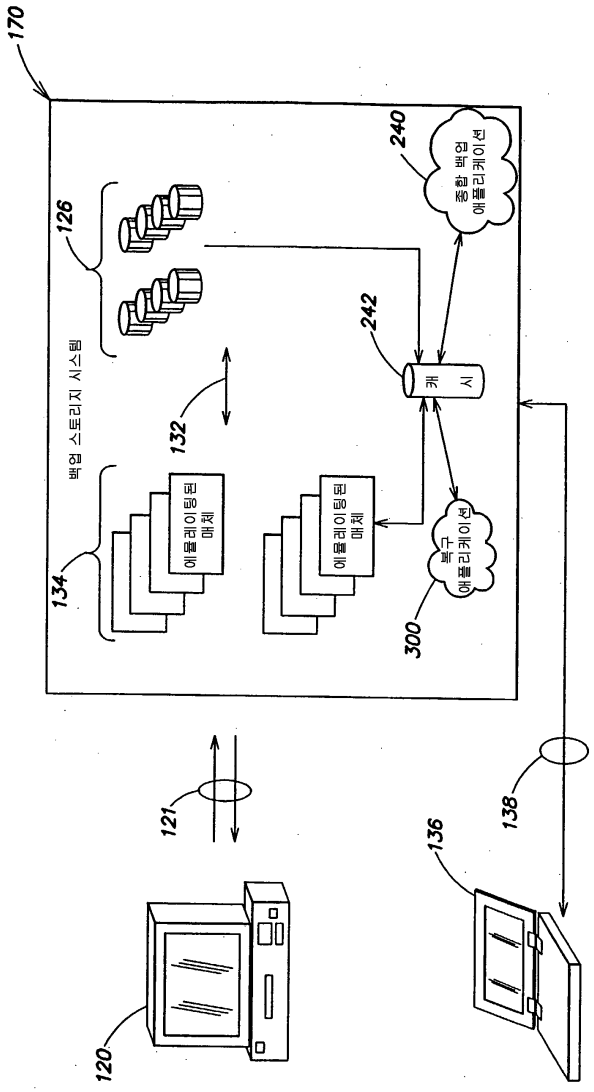
백업 데이터 세트의 가장 최근 버전의 하나 이상의 데이터 파일 각각이 저장된 스토리지 매체상의 개개의 저장 위치를 식별하는 하나 이상의 제 2 식별자를 포함하는 데이터구조가 저장된 것을 특징으로 하는 컴퓨터 판독가능 매체.

도면

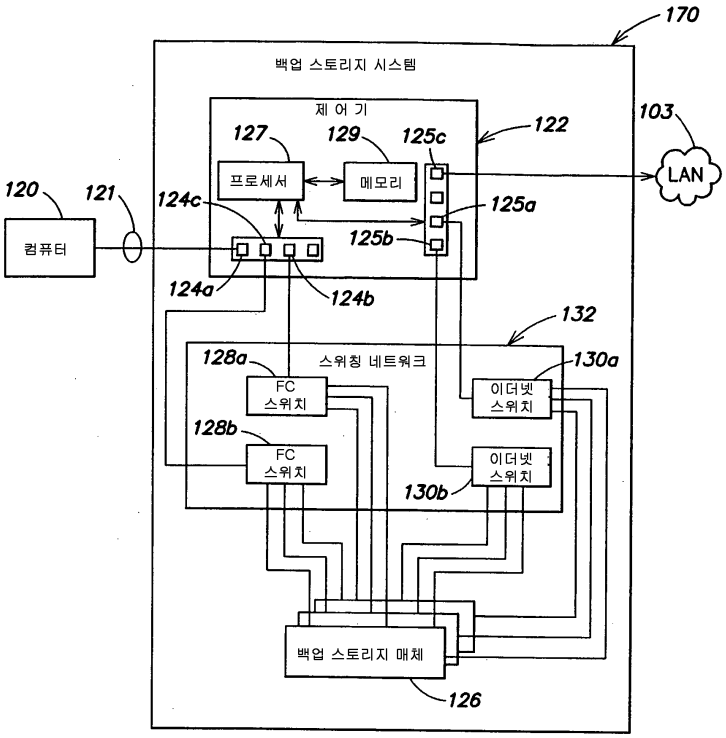
도면1



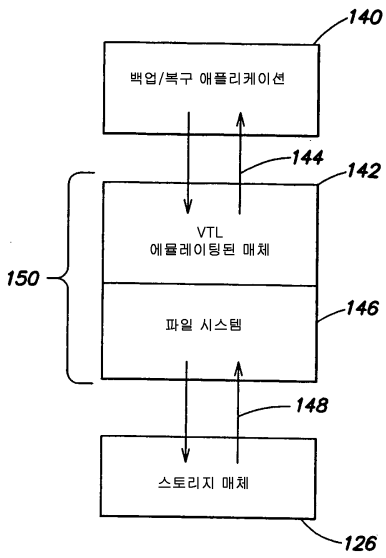
도면2


170
240
종합 백업
애플리케이션
126
백업 스토리지 시스템
242
캐 시
132
에플레이팅된
매체
에플레이팅된
매체
134
복구
애플리케이션
300
121
138
136
120

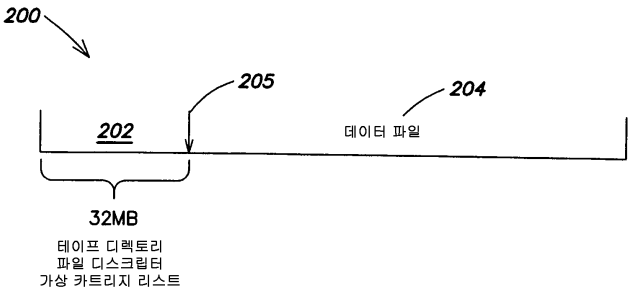
도면3


170
백업 스토리지 시스템
제 어 기
127
129
125c
122
103
LAN
프로세서
메모리
120
121
124c
125a
컴퓨터
125b
124a
124b
132
128a
스위칭 네트워크
130a
FC
스위치
이더넷
스위치
128b
FC
스위치
이더넷
스위치
130b
백업 스토리지 매체
126

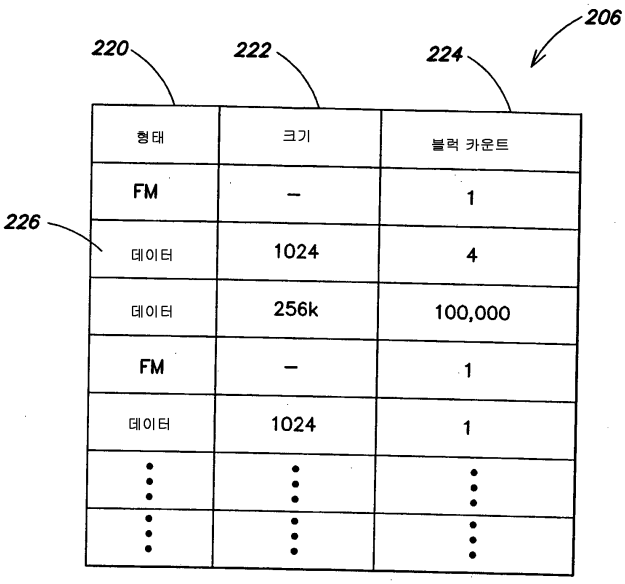
도면4


140
백업/복구 애플리케이션
144
142
VTL
에뮬레이팅된 매체
150
파일 시스템
146
148
스토리지 매체
126

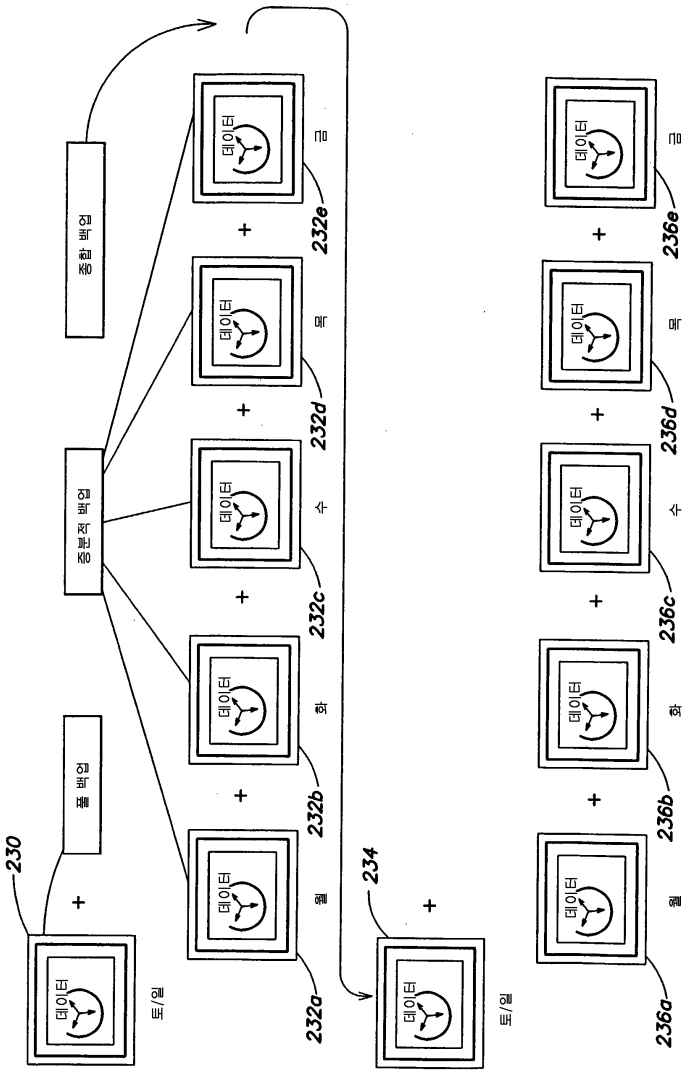
도면5


200
205
204
202
데이터 파일
32MB
테이프 디렉토리
파일 디스크립터
가상 카트리지 리스트

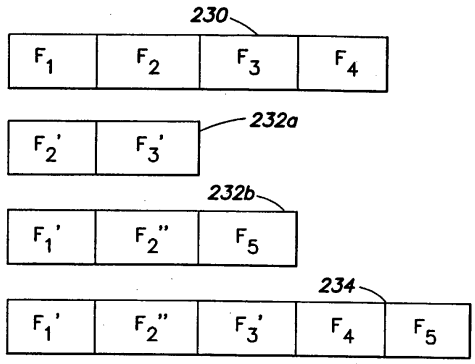
도면6


206
220
222
224
226
형태
크기
블럭 카운트
FM
–
1
데이터
1024
4
데이터
256k
100,000
FM
–
1
데이터
1024
1

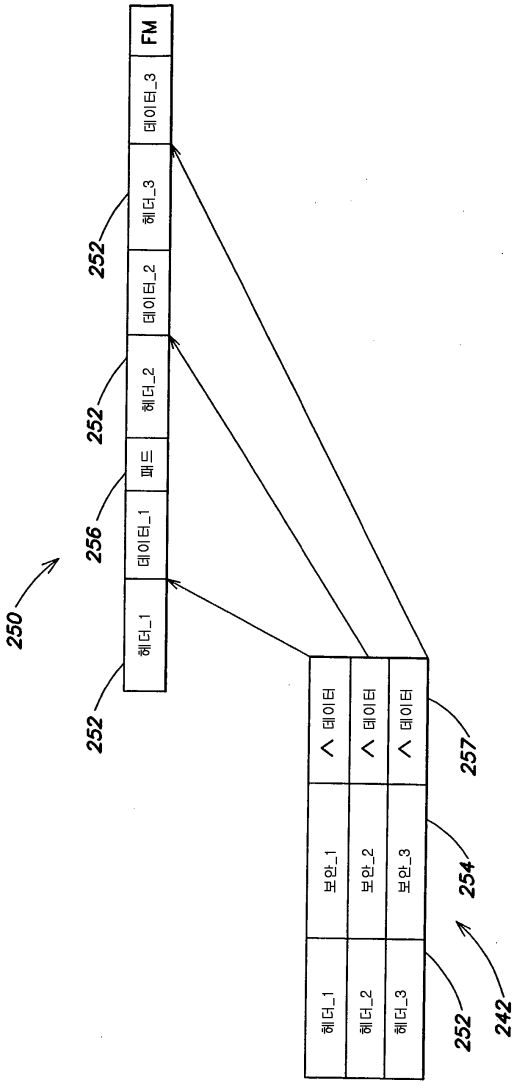
도면7


종합 백업
증분적 백업
풀 백업
데이터
토/일
월
화
수
목
금
232a
232b
232c
232d
232e
230
234
236a
236b
236c
236d
236e

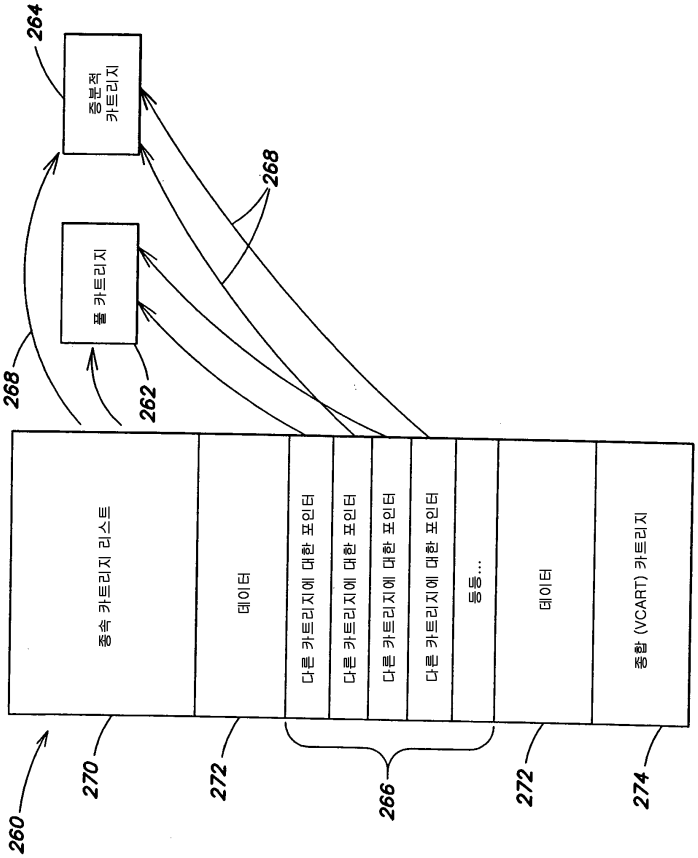
도면8


230
F1
F2
F3
F4
232a
F2'
F3'
232b
F1'
F2''
F5
234
F1'
F2''
F3'
F4
F5

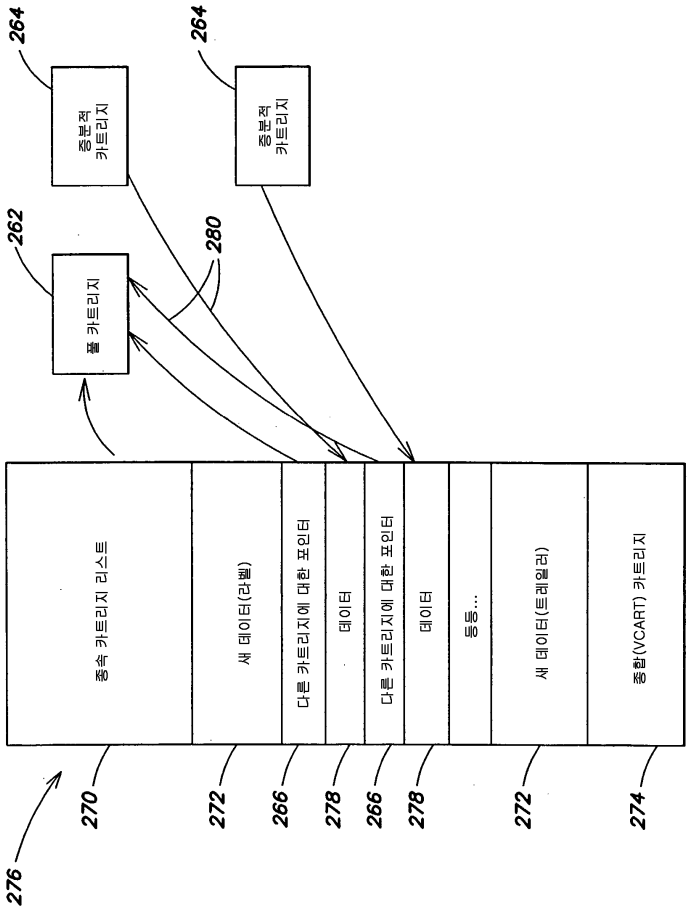
도면9


250
252
256
252
252
252
242
254
257
헤더_1
데이터_1
패드
헤더_2
데이터_2
헤더_3
데이터_3
FM
헤더_1
헤더_2
헤더_3
보안_1
보안_2
보안_3
데이터
데이터
데이터

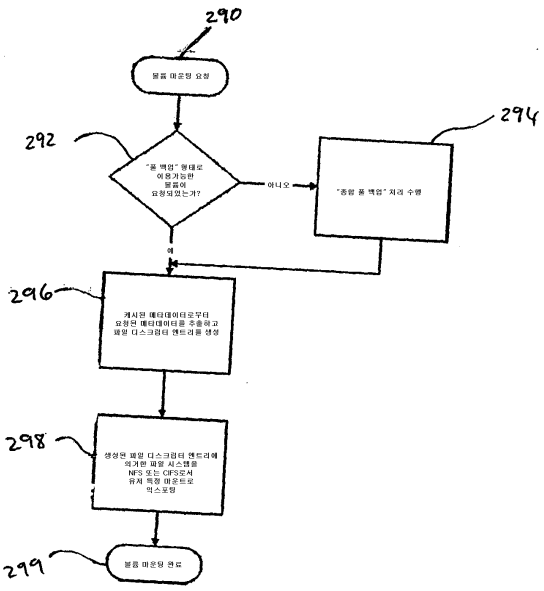
도면10


264
증분적 카트리지
268
풀 카트리지
268
262
종속 카트리지 리스트
데이터
다른 카트리지에 대한 포인터
다른 카트리지에 대한 포인터
다른 카트리지에 대한 포인터
다른 카트리지에 대한 포인터
등등...
데이터
종합 (VCART) 카트리지
260
270
272
266
272
274

도면11

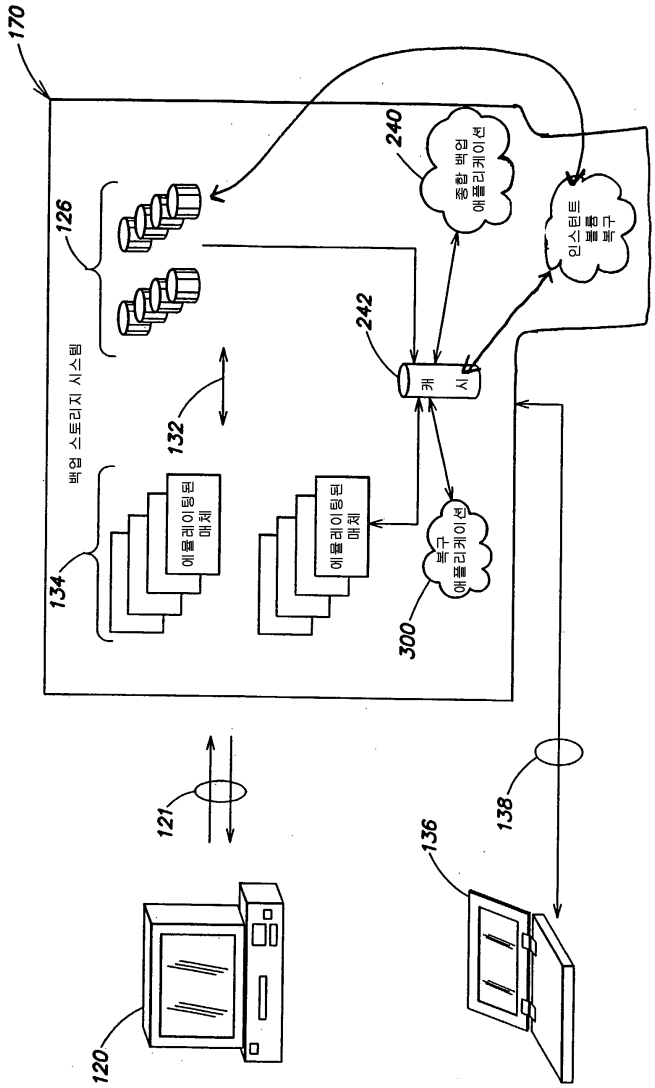


도면12

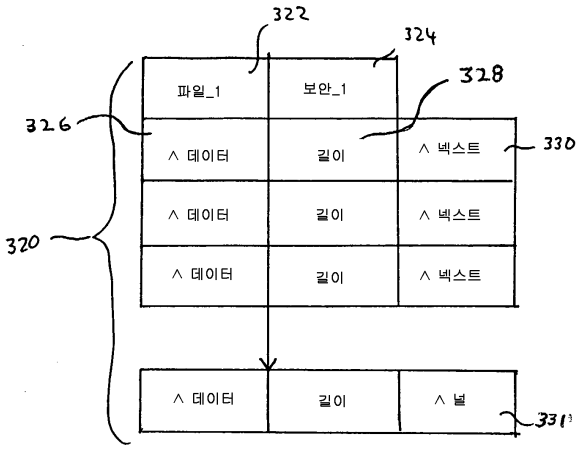


볼륨 마운팅 순서도

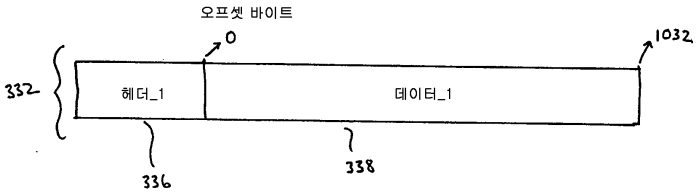
도면13


170
240
종합 백업 애플리케이션
인스턴트 복구
126
242
캐 시
132
백업 스토리지 시스템
에뮬레이팅된 매체
에뮬레이팅된 매체
134
복구 애플리케이션
300
121
138
136
120

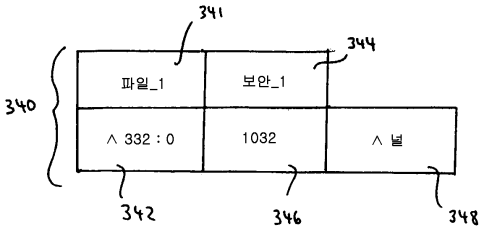
도면14


322
324
328
326
330
320
파일_1
보안_1
^ 데이터
길이
^ 넥스트
^ 데이터
길이
^ 넥스트
^ 데이터
길이
^ 넥스트
^ 데이터
길이
^ 널

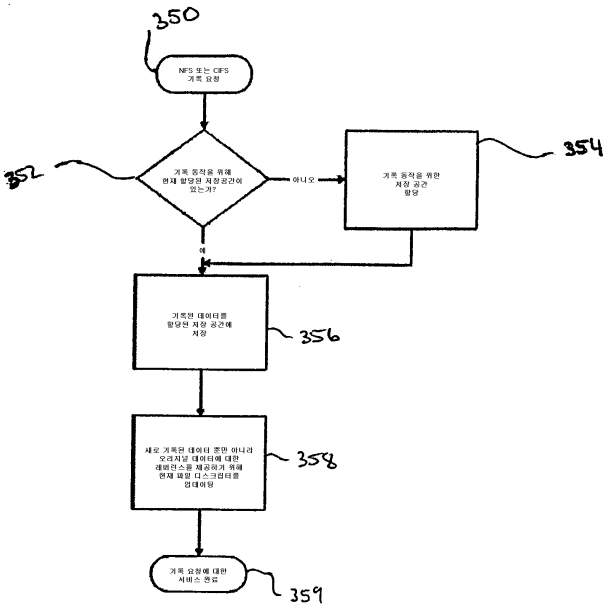
도면15



도면16

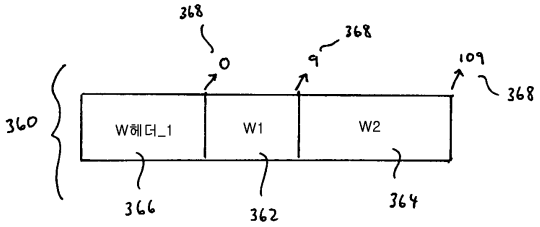


도면17



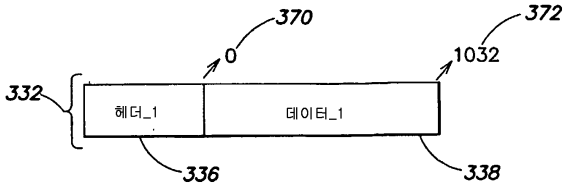
NFS 또는 CIFS 기록 요청 순서도

도면18

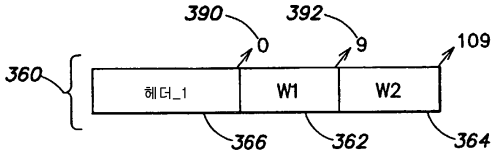


저장된 NFS 또는 CIFS 기록 데이터

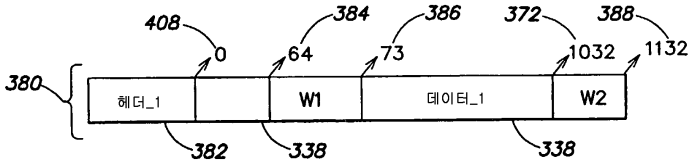
도면19



(+)



(=)



도면20