(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。Int. Cl.
G06F 15/173 (2006.01)

(11) 공개번호 10-2007-0028362
(43) 공개일자 2007년03월12일

(21) 출원번호 10-2006-7022721
(22) 출원일자 2006년10월30일
심사청구일자 없음
번역문 제출일자 2006년10월30일
(86) 국제출원번호 PCT/EP2005/052331
국제출원일자 2005년05월20일
(87) 국제공개번호 WO 2005/114372
국제공개일자 2005년12월01일

(30) 우선권주장 10/851,036 2004년05월21일 미국(US)

(71) 출원인 인터내셔널 비지네스 머신즈 코포레이션
미국 10504 뉴욕주 아몬크 뉴오차드 로드

(72) 발명자 코렐 스티븐
미국 오레곤주 97215 포틀랜드 사우스이스트 62번 애비뉴 222
시저 제임스 존
미국 오레곤주 97006 포틀랜드 노스웨스트 선다운 웨이 229
웨들레이크 마틴 브루스
미국 오레곤주 97124 힐스보로 노스이스트 텐스 애비뉴 2385

(74) 대리인 김창세
장성구
김원준

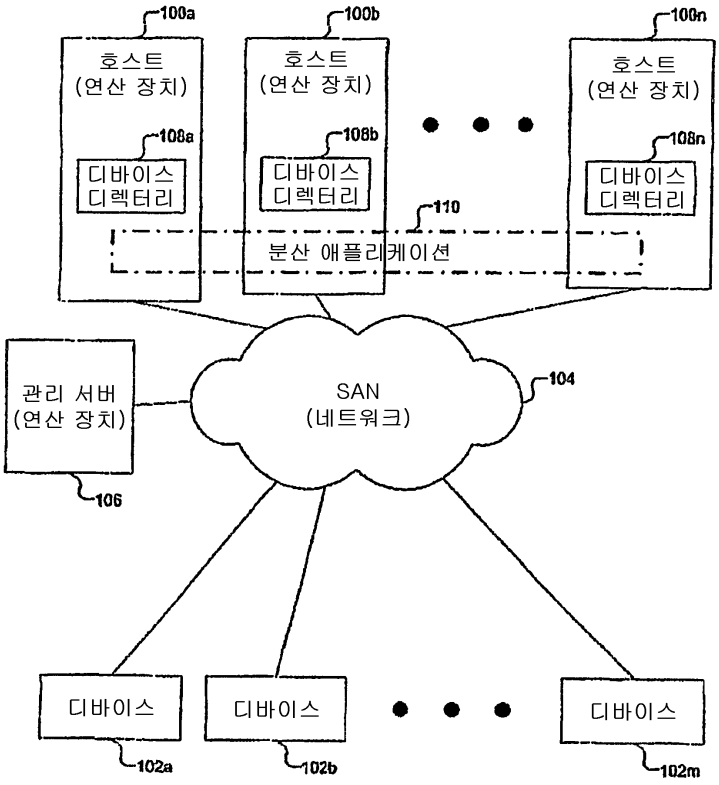
전체 청구항 수 : 총 23 항

(54) 디바이스 정보 저장 방법, 데이터 프로세싱 시스템 및컴퓨터 프로그램

(57) 요약

본 발명은 방법, 시스템 및 컴퓨터 프로그램에 관한 것으로, 복수의 디바이스에 대응하는 복수의 파일에 대한 복수의 참조는 연산 장치에서 실행되는 데이터 구조체에 저장되고, 연산 장치는 네트워크를 통해 복수의 디바이스에 결합된다. 분산 애플리케이션은 데이터 구조체에 대해 액세스할 수 있는데, 분산 애플리케이션은 데이터 구조체에 저장된 참조를 사용하여 디바이스에 대응하는 파일을 결정하고, 이 결정된 파일을 통해 디바이스에 대하여 데이터 전송 연산을 수행한다.

대표도



특허청구의 범위

청구항 1.

복수의 디바이스에 대응하는 복수의 파일에 대한 복수의 참조를 연산 장치에서 실행되는 데이터 구조체에 저장하는 단계 -상기 연산 장치는 네트워크를 통해 복수의 디바이스에 결합됨- 와,

분산 애플리케이션(distributed application)에 상기 데이터 구조체에 대한 액세스를 제공하는 단계 -상기 분산 애플리케이션은 상기 데이터 구조체에 저장된 참조(reference)를 사용하여 디바이스에 대응하는 파일을 결정하고, 상기 결정된 파일을 통해 상기 디바이스에 대한 데이터 전송 연산을 수행함- 를 포함하는

방법.

청구항 2.

제 1 항에 있어서,

상기 데이터 구조체는 디렉터리이고,

상기 파일은 디바이스 파일이며,

상기 참조는 상기 디바이스 파일에 대한 소프트링크(soft link)인

방법.

청구항 3.

제 1 항에 있어서,

상기 데이터 구조체는 레지스트리(registry)이고,

상기 레지스트리 내의 항목들(entries)은 상기 참조를 포함하는

방법.

청구항 4.

제 1 항 내지 제 3 항 중 어느 한 항에 있어서,

상기 네트워크에 추가되는 추가 디바이스에 대응하는 추가 파일에 해당하는 추가 참조를 결정하는 데 사용될 수 있는 정보를 다른 연산 장치로부터 수신하는 단계와,

상기 추가 참조를 포함하도록 상기 데이터 구조체를 갱신(update)하는 단계를 더 포함하는

방법.

청구항 5.

제 1 항 내지 제 3 항 중 어느 한 항에 있어서,

상기 네트워크에 추가되었던 추가 디바이스를 발견하는 단계와,

상기 추가 디바이스에 대응하는 추가 파일에 해당하는 추가 참조를 결정하는 단계와,

상기 추가 참조를 포함하도록 상기 데이터 구조체를 갱신하는 단계를 더 포함하는

방법.

청구항 6.

제 1 항 내지 제 5 항 중 어느 한 항에 있어서,

상기 네트워크는 SAN(storage area network)이고,

상기 분산 애플리케이션은 복수의 연산 장치를 통해 복수의 디바이스에 액세스할 수 있는

방법.

청구항 7.

제 1 항 내지 제 6 항 중 어느 한 항에 있어서,

상기 연산 장치는 제 1 연산 장치이고,

상기 데이터 구조체는 제 1 데이터 구조체이며,

상기 방법은,

상기 복수의 디바이스에 대응하는 상기 복수의 파일에 대한 상기 복수의 참조 중 적어도 하나를 제 2 연산 장치에서 실행되는 제 2 데이터 구조체에 저장하는 단계를 더 포함하되,

상기 제 2 연산 장치는 상기 네트워크를 통해 상기 복수의 디바이스에 결합되고, 상기 분산 애플리케이션은 상기 제 1 데이터 구조체 및 제 2 데이터 구조체를 통해 상기 복수의 디바이스에 액세스할 수 있는

방법.

청구항 8.

제 1 항 내지 제 7 항 중 어느 한 항에 있어서,

상기 데이터 구조체는 복수의 이종 운영 체제(heterogeneous operating system)에서 실행될 수 있고,

상기 복수의 디바이스는 이종인

방법.

청구항 9.

제 1 항 내지 제 8 항 중 어느 한 항에 있어서,

상기 데이터 구조체는 상기 연산 장치에서 국부적으로 실행되고,

상기 분산 애플리케이션은 상기 연산 장치에서 국부적으로 실행되는 상기 데이터 구조체에 액세스함으로써, 상기 연산 장치로부터 원격으로 실행되는 상기 데이터 구조체에 액세스하는 것에 비해 더 빠르게 상기 디바이스에 대해 상기 데이터 전송 연산을 개시할 수 있는

방법.

청구항 10.

제 1 항에 있어서,

상기 연산 장치 내의 운영 체제 및 드라이버는 상기 디바이스에 대해 상기 데이터 전송 연산을 수행하기 위한 정보에 대한 액세스를 상기 분산 애플리케이션에 직접 제공할 수 없는

방법.

청구항 11.

제 1 항에 있어서,

상기 파일은 디바이스 파일이고,

상기 디바이스는 가상 저장 디바이스이며,

상기 네트워크는 SAN이고,

상기 디바이스 파일은 상기 SAN을 통해 상기 가상 저장 디바이스까지의 경로를 나타내는

방법.

청구항 12.

네트워크를 통해 복수의 디바이스 및 분산 애플리케이션과 통신할 수 있는 데이터 프로세싱 시스템에 있어서,

연산 장치와,

상기 연산 장치에 결합된 메모리와,

상기 복수의 디바이스에 대응하는 복수의 파일에 대한 복수의 참조를 상기 연산 장치에서 실행되는 데이터 구조체에 저장하는 것을 제어하는 수단과,

상기 데이터 구조체에 대한 액세스를 상기 분산 애플리케이션에 제공하는 수단을 포함하되,

상기 분산 애플리케이션은 상기 데이터 구조체에 저장된 참조를 사용하여 디바이스에 대응하는 파일을 결정하고, 상기 결정된 파일을 통해 상기 디바이스에 대하여 데이터 전송 연산을 수행하는

데이터 프로세싱 시스템.

청구항 13.

제 12 항에 있어서,

상기 데이터 구조체는 디렉터리이고,

상기 파일은 디바이스 파일이며,

상기 참조는 상기 디바이스 파일에 대한 소프트링크인

데이터 프로세싱 시스템.

청구항 14.

제 12 항에 있어서,

상기 데이터 구조체는 레지스트리이고,

상기 레지스트리 내의 항목들은 상기 참조를 포함하는

데이터 프로세싱 시스템.

청구항 15.

제 12 항에 있어서,

상기 네트워크에 추가될 때, 또 다른 연산 장치와 통신할 수 있으며,

추가 디바이스에 대응하는 추가 파일에 해당하는 추가 참조를 결정하는 데 사용될 수 있는 정보를 또 다른 연산 장치로부터 수신하는 수단과,

상기 추가 참조를 포함하도록 상기 데이터 구조체를 갱신하는 수단을 더 포함하는

데이터 프로세싱 시스템.

청구항 16.

제 12 항에 있어서,

상기 네트워크에 추가되는 추가 디바이스에 응답하여, 상기 추가 디바이스를 발견하는 단계와, 상기 추가 디바이스에 대응하는 추가 파일에 해당하는 추가 참조를 결정하는 단계와, 상기 추가 참조를 포함하도록 상기 데이터 구조체를 갱신하는 단계를 수행하는 수단을 더 포함하는

데이터 프로세싱 시스템.

청구항 17.

제 12 항에 있어서,

상기 네트워크는 SAN이고,

상기 분산 애플리케이션은 상기 복수의 연산 장치를 통해 상기 복수의 디바이스에 액세스할 수 있는

데이터 프로세싱 시스템.

청구항 18.

제 12 항에 있어서,

상기 연산 장치는 제 1 연산 장치이고,

제 2 연산 장치는 상기 네트워크에 결합되며,

상기 데이터 구조체는 제 1 데이터 구조체이고,

상기 시스템은,

상기 복수의 디바이스에 대응하는 상기 복수의 파일에 대한 상기 복수의 참조 중 적어도 하나를 상기 제 2 연산 장치에서 실행되는 제 2 데이터 구조체에 저장하는 수단을 더 포함하되,

상기 제 2 연산 장치는 상기 네트워크를 통해 상기 복수의 디바이스에 결합되고, 상기 분산 애플리케이션은 상기 제 1 데이터 구조체 및 제 2 데이터 구조체를 통해 상기 복수의 디바이스에 액세스할 수 있는

데이터 프로세싱 시스템.

청구항 19.

제 12 항에 있어서,

상기 데이터 구조체는 복수의 이종 운영 체제에서 실행될 수 있고,

상기 복수의 디바이스는 이종인

데이터 프로세싱 시스템.

청구항 20.

제 12 항에 있어서,

상기 데이터 구조체는 상기 연산 장치에서 국부적으로 실행되고,

상기 분산 애플리케이션은 상기 연산 장치에서 국부적으로 실행되는 상기 데이터 구조체에 액세스함으로써, 상기 연산 장치로부터 원격으로 실행되는 상기 데이터 구조체에 액세스하는 것에 비해 더 빠르게 상기 디바이스에 대해 상기 데이터 전송 연산을 개시할 수 있는

데이터 프로세싱 시스템.

청구항 21.

제 12 항에 있어서,

상기 연산 장치 내의 운영 체제 및 드라이버는 상기 디바이스에 대해 상기 데이터 전송 연산을 수행하는 정보에 대한 액세스를 상기 분산 애플리케이션에 직접 제공할 수 없는

데이터 프로세싱 시스템.

청구항 22.

제 12 항에 있어서,

상기 파일은 디바이스 파일이고,

상기 디바이스는 가상 저장 디바이스이며,

상기 네트워크는 SAN이고,

상기 디바이스 파일은 상기 SAN을 통해 상기 가상 저장 디바이스까지의 경로를 나타내는

데이터 프로세싱 시스템.

청구항 23.

네트워크에 의해 복수의 디바이스에 결합된 연산 장치에서 실행하는 컴퓨터 프로그램에 있어서,

복수의 디바이스에 대응하는 복수의 파일에 대한 복수의 참조를 상기 연산 장치에서 실행되는 데이터 구조체에 저장하는 단계와,

분산 애플리케이션에 상기 데이터 구조체에 대한 액세스를 제공하는 단계 -상기 분산 애플리케이션은 상기 데이터 구조체에 저장된 참조를 사용하여 디바이스에 대응하는 파일을 결정하고, 상기 결정된 파일을 통해 상기 디바이스에 대하여 데이터 전송 연산을 수행함- 를 수행하기 위해 상기 연산 장치를 제어하는

컴퓨터 프로그램.

명세서

기술분야

본 발명은 디바이스 정보를 저장하고 저장된 정보에 대한 액세스를 제공하는 방법, 시스템 및 컴퓨터 프로그램에 관한 것이다.

배경기술

SAN(storage area network)은 복수의 저장 디바이스와 관련된 데이터 서버를 상호연결하는 특수한 목적의 네트워크이다. 소정의 구현예에서, SAN은 공유된 저장 디바이스의 고속 서브네트워크일 수 있다. 저장 디바이스는 데이터를 저장하는 복수의 디스크, 테이프 또는 다른 저장 매체를 포함하는 머신이다.

SAN은 복수의 저장 디바이스에 복수의 호스트를 결합할 수 있는데, 여기서 호스트는 파일 서버일 수 있다. 소정의 구현예에서, SAN은 IP 기반 네트워크와는 다른 저장 네트워크일 수 있다.

SAN은 IBM® z990 메인프레임과 같은 다른 컴퓨팅 리소스에 근접하여 클러스터링될 수도 있지만, 특정 SAN은 LAN 캐리어 기술을 사용함으로써 백업 및 기록 저장을 위한 원격 위치까지 확장할 수도 있다. SAN은 IBM의 광섬유 기반 ESCON®(optical fiber based Enterprise System Connection), 섬유 채널 기술(Fibre Channel technology) 등과 같은 통신 기술을 사용할 수 있다. SAN은 디스크 모니터링, 백업 및 복원, 데이터의 기록 및 검색, 하나의 저장 디바이스로부터 다른 저장 디바이스로의 데이터 이동 및 네트워크 내의 서로 다른 서버들 간의 데이터 공유를 지원할 수 있다. 특정 SAN은 서브네트워크와 NAS(network-attached stroage) 시스템을 통합할 수도 있다.

발명의 상세한 설명

데이터를 저장하고, 저장된 데이터에 대한 액세스를 제공하는 방법, 시스템 및 컴퓨터 프로그램이 제공된다. 복수의 디바이스에 대응하는 복수의 파일에 대한 복수의 참조는 연산 장치에서 실행되는 데이터 구조체에 저장된다. 연산 장치는 네트워크를 통해 복수의 디바이스에 결합된다. 분산 애플리케이션은 데이터 구조체에 액세스할 수 있는데, 이 분산 애플리케이션은 데이터 구조체에 저장된 참조를 사용하여 디바이스에 대응하는 파일을 결정하고, 이 결정된 파일을 통해 디바이스에 대하여 데이터 전송 연산을 수행한다.

소정의 실시예에서, 데이터 구조체는 디렉터리이고, 파일은 디바이스 파일이며, 참조는 디바이스 파일에 대한 소프트링크이다.

소정의 다른 실시예에서, 데이터 구조체는 레지스트리이고, 레지스트리 내의 항목들은 참조(reference)를 포함한다.

또 다른 실시예에서, 다른 연산 장치로부터 정보가 수신되는데, 이 정보는 네트워크에 추가되는 추가 디바이스에 대응하는 추가 파일에 해당하는 추가 참조를 결정하는 데 사용될 수 있다. 추가 참조를 포함하도록 데이터 구조체가 갱신된다.

다른 실시예에서, 네트워크에 추가되었던 추가 디바이스가 발견된다. 추가 디바이스에 대응하는 추가 파일에 해당하는 추가 참조를 결정한다. 추가 참조를 포함하도록 데이터 구조체가 갱신된다.

다른 실시예에서, 네트워크는 SAN이고, 분산 애플리케이션은 복수의 연산 장치를 통해 복수의 디바이스에 액세스할 수 있다.

또 다른 실시예에서, 연산 장치는 제 1 연산 장치이고, 데이터 구조체는 제 1 데이터 구조체이다. 제 2 연산 장치에서 실행되는 제 2 데이터 구조체는 복수의 디바이스에 대응하는 복수의 파일에 대한 복수의 참조 중 적어도 하나를 저장하고, 제 2 연산 장치는 네트워크를 통해 복수의 디바이스에 결합되며, 분산 애플리케이션은 제 1 및 제 2 데이터 구조체를 통해 복수의 디바이스에 액세스할 수 있다.

다른 실시예에서, 데이터 구조체는 복수의 이종 운영 체제에서 실행될 수 있고, 복수의 디바이스는 이종이다.

또 다른 실시예에서, 데이터 구조체는 연산 장치에서 국부적으로 실행되고, 분산 애플리케이션은 연산 장치에서 국부적으로 실행되는 데이터 구조체에 액세스함으로써, 연산 장치로부터 원격으로 실행되는 데이터 구조체에 액세스하는 것에 비해 더 빠르게 디바이스에 대하여 데이터 전송 연산을 개시할 수 있다.

다른 실시예에서, 연산 장치 내의 운영 체제 및 드라이버는 디바이스에 대하여 데이터 전송 연산을 수행하기 위한 정보에 대한 액세스를 분산 애플리케이션에 직접 제공할 수 없다.

다른 실시예에서, 파일은 디바이스 파일이고, 디바이스는 가상 저장 디바이스이며, 네트워크는 SAN이고, 디바이스 파일은 SAN을 통해 가상 저장 디바이스까지의 경로를 나타낸다.

본 발명의 실시예는 동일한 번호는 동일한 부품을 나타내는 첨부 도면을 예로서만 참조하여 이하에 설명된다.

실시예

이하의 설명에서는, 본 명세서의 일부분이며 몇몇 실시예를 도시한 첨부 도면에 대하여 언급한다. 다른 실시예가 이용될 수 있고, 구조 및 연산 상의 변화가 있을 수 있음은 물론이다.

도 1은 소정의 실시예가 구현되는 컴퓨팅 환경을 도시한다. 복수의 호스트(100a 내지 100n)는 SAN(104)과 같은 네트워크를 통해 복수의 디바이스(102a 내지 102m)에 결합된다. 소정의 실시예에서, 호스트(100a 내지 100n) 및 디바이스(102a 내지 102m)에 대해 연산을 수행할 수 있는 관리 서버(106)도 SAN(104)에 결합된다.

복수의 호스트(100a 내지 100n) 및 관리 서버(106)는 워크스테이션, 데스크탑 컴퓨터, 랩탑, 메인프레임, 전화 장치, 휴대용 컴퓨터, 서버, 블레이드(blade) 컴퓨터 등과 같은 임의의 유형의 연산 장치를 포함할 수 있다. 복수의 호스트(100a 내지 100n)는 복수의 디바이스 디렉터리(108a 내지 108n)를 포함할 수 있는데, 소정의 실시예에서, 적어도 하나의 호스트는 하나의 디바이스 디렉터리를 포함한다. 예컨대, 호스트(100a)는 디바이스 디렉터리(108a)를 포함할 수 있고, 호스트(100b)는 디바이스 디렉터리(108b)를 포함할 수 있으며, 호스트(100n)는 디바이스 디렉터리(108n)를 포함할 수 있다. 소정의 실시예에서, 디바이스 디렉터리(108a 내지 108n)는 파일 디렉터리이고, 복수의 디바이스(102a 내지 102m) 중 하나 이상에 대응하는 디바이스 파일에 대한 참조(reference)를 포함한다. 소정의 실시예에서, 호스트(100a 내지 100n)는 이종(heterogeneous)일 수 있고, 복수의 운영 체제를 실행할 수 있다.

디바이스(102a 내지 102m)는 디스크 드라이브, 테이프 드라이브, CDROM 드라이브 등처럼 당해 기술 분야에 알려져 있는 임의의 유형의 저장 디바이스를 포함할 수 있다. 디바이스(102a 내지 102m)는 SAN(104)을 통해 호스트(100a 내지 100n) 및 관리 서버(106)로부터 액세스될 수 있는 이종 저장 디바이스 그룹을 포함할 수 있다. 소정의 실시예에서, 복수의 디바이스(102a 내지 102m)는 복수의 호스트(100a 내지 100n) 사이에서 공유된다.

SAN(104)은 당해 기술에 알려져 있는 임의의 SAN을 포함할 수 있다. 소정의 실시예에서, SAN(104)은 인터넷, 인트라넷, LAN, WALN 등처럼 당해 기술에 알려져 있는 기타 네트워크(도시 생략)에 결합될 수 있다.

분산 애플리케이션(110)은 복수의 호스트(100a 내지 100n) 중 하나 이상 내의 소프트웨어 구성 요소를 실행하거나 이들과 상호작용할 수 있다. 분산 애플리케이션(110)은 복수의 호스트(100a 내지 100n) 중 하나 이상과 상호작용하거나 이들

내에서 실행할 수 있다. 소정의 실시예에서, 분산 애플리케이션(110)은 SAN(104)에서 복수의 호스트 및 디바이스를 사용하는 임의의 SAN 애플리케이션을 포함할 수 있다. 분산 애플리케이션(110)은 재난 복구 애플리케이션, 데이터 상호교환 애플리케이션, 데이터 저장(vaulting) 애플리케이션, 데이터 보호 애플리케이션 등을 포함할 수 있다.

분산 애플리케이션(110)은 복수의 이종 디바이스(102a 내지 102m) 및 호스트(100a 내지 100n) 내의 이종 호스트 운영 체제와 상호작용해야 하므로, 분산 애플리케이션(110)은 SAN(104)에서 디바이스(102a 내지102m)의 사용을 관리하거나 허용하는 데 있어서 호스트 운영 체제, 클러스터 관리자, 논리적 볼륨 관리자 등에 전적으로 의존할 수 없다. 추가적으로, 디바이스(102a 내지 102m)가 호스트(100a 내지 100n) 사이에서 공유되면, 호스트 운영 체제, 클러스터 관리자 등은 디바이스(102a 내지 102m)를 관리하는 데 필요한 정보를 가질 수 없다. 또한, 호스트 운영 체제, 호스트 버스 어댑터 드라이버 또는 저장 디바이스 드라이버의 모든 조합에 적합한 지원을 처리하는 것은 애플리케이션 공급자에게 부담이 될 수 있다. 따라서, 디바이스(102a 내지 102m)의 관리는 능력 부족 및 호스트 운영 체제, 호스트 버스 어댑터 드라이버, 저장 디바이스 드라이버 등의 복수의 조합에 적합한 지원 생성과 관련된 비용 때문에, 호스트 운영 체제, 클러스터 관리자, 논리적 볼륨 관리자 등을 통해 가능할 수 있다.

도 1은 디바이스(102a 내지 102m)에 관련된 정보가 디바이스 디렉터리(108a 내지 108n)에 저장되는 실시예를 도시하는데, 디바이스 디렉터리(108a 내지 108n)는 분산 애플리케이션(110)에 액세스할 수 있다. 추가적으로, 디바이스 디렉터리(108a 내지 108n)는 디바이스 디렉터리가 독립적 운영 체제(operating system neutral)이며 디바이스에 관련된 정보를 분산된 애플리케이션(110)에 인터페이싱하기에 적합한 형태로 저장하는 방식으로 구현된다. 소정의 실시예는 호스트(100a 내지 100n) 및 디바이스(102a 내지 102m)가 클러스터로 분류되는 컴퓨팅 환경에서 구현될 수 있다. 분산 애플리케이션(110)은 클러스터 기반 운영 체제를 통해 실행할 수 있고, 디바이스(102a 내지 102m)에 액세스하는 디바이스 디렉터리(108a 내지 108n)를 사용할 수 있다.

도 2는 호스트(200)가 호스트(100a 내지 100n) 중 임의의 호스트임을 나타내는 호스트(200)의 블록도를 도시한다. 호스트(200)는 시스템 소프트웨어(202), 디바이스 디렉터리(204)를 포함하고, 분산 애플리케이션(110)과 상호작용할 수 있는데, 소정의 실시예에서, 분산 애플리케이션(110)은 하나 이상의 호스트(100a 내지 100n)에서 구현될 수 있다. 호스트(200)에 포함된 시스템 소프트웨어(202)는 호스트(200)의 운영체제, 호스트(200)에서 실행하는 다양한 드라이버, 호스트(200)에서 실행하는 클러스터 관리자, 호스트(200)에서 실행하는 논리적 볼륨 관리자 등을 포함할 수 있다. 디바이스 디렉터리(204)는 디바이스 디렉터리(108a 내지 108n) 중 임의의 디렉터리를 나타낼 수 있다. 예컨대, 소정의 실시예에서 호스트(200)가 호스트(100a)를 나타내면, 디바이스 디렉터리(204)는 디바이스 디렉터리(108a)를 나타낸다.

디바이스 디렉터리(204)는 복수의 디바이스 파일 링크(206a 내지 206p)를 포함하는데, 디바이스 파일 링크(206a 내지 206p)는 디바이스(102a 내지 102m)에 대응하는 디바이스 파일에 대한 참조이고, 디바이스 파일은 분산 애플리케이션(110)에 의해 사용되어, 디바이스 파일에 대응하는 디바이스에 대해 데이터 전송 연산을 수행할 수 있다. 예컨대, 소정의 실시예에서 "x"라는 이름의 디바이스 파일이 디바이스(102m)에 대응하면, 디바이스 파일 링크(206n)는 디바이스 파일 "x"에 대한 소프트링크(softlink)일 수 있다. 소프트링크는 SAN(104) 내의 디바이스 파일 "x"의 위치를 나타낼 수 있다. 예컨대, 소프트링크는 "/dev/home/x"으로서 나타날 수 있는데, "x"라는 이름의 파일은 "dev"의 "home" 디렉터리에 저장되고, "dev"는 SAN(104)에 결합되는 연산 장치(100a 내지 100n), 관리 서버(106), 디바이스(102a 내지 102m) 또는 파일 "x"을 저장할 수 있는 기타 구성 요소 중 임의의 것으로 나타날 수 있다. 소정의 실시예에서, 디바이스 파일은 호스트(100a 내지 100n)와 같은 호스트 상에 존재하고, SAN을 통해 저장 디바이스(102a 내지 102m)와 같은 저장 디바이스까지의 경로 또는 가능한 경로 세트를 식별하는데, 저장 디바이스는 저장 서버에 의해 제공되는 가상 디스크를 포함할 수 있다. 특정 운영 체제에서, 디바이스 파일은 애플리케이션이 디바이스 파일을 개방, 판독 또는 기록함으로써 대응하는 디바이스를 사용할 수 있게 한다. 예컨대, 디바이스 파일에 기록함으로써, 분산 애플리케이션(110)과 같은 애플리케이션은 드라이버 및 SAN(104)을 통해 저장 디바이스(102a 내지 102m)에 기록한다. 애플리케이션은 대응하는 디바이스 파일에 대해 연산을 수행함으로써, 디바이스의 SAN 주소와 같은 디바이스에 대한 특정 정보도 획득할 수 있다. 소정의 실시예에서, 디바이스 파일에 대한 링크는 파일이 실제 디바이스 파일을 대신하여 디바이스 파일의 프록시(proxy)로서 실행하는 운영 체제 기능이다. 애플리케이션은 링크를 개방할 수 있고, 이 링크에서 디바이스 파일에 대한 링크 지점 상의 연산과 유사한 연산을 수행할 수 있다. 애플리케이션은 디바이스 파일에 대한 링크 지점이 무엇인지를 결정하는 운영 체제도 필요로 할 수 있다.

소정의 실시예에서, 디바이스 디렉터리(204)는 디바이스 파일 링크(206a 내지 206p)를 포함하는 파일 디렉터리이다. 다른 실시예에서, 디바이스 디렉터리(204)는 디바이스(102a 내지 102m)와 관련된 정보에 대한 참조를 저장할 수 있는 임의의 데이터 구조체일 수 있다. 소정의 실시예에서, 특정 디바이스에 대한 디바이스 파일 링크와 관련된 식별자와 같은 추가 필드는 디바이스 디렉터리(204)에 포함된다.

일 실시예에서, 분산 애플리케이션(110)은 디바이스 디렉터리(204)에 저장된 디바이스 파일 링크(206a 내지 206p)를 통해 디바이스(102a 내지 102m)에 대응하는 디바이스 파일에 액세스함으로써, 디바이스(102a 내지 102m)에 대해 I/O 연산과 같은 데이터 전송 연산을 수행한다. 소정의 실시예에서, 분산 애플리케이션(110)이 디바이스 파일 링크를 이용하려 하기 전에, 디바이스 디렉터리(204)가 파일 링크(206a 내지 206p)로 생성되어 배치된다. 디바이스 디렉터리(204)는 호스트(200)에 국부적으로 저장되므로, 분산 애플리케이션(110)은 디바이스(102a 내지 102m)에 대한 참조가 호스트(200)에서 국부적으로 이용가능하지 않은 구현예에 비해 더 빠른 디바이스(102a 내지 102m)에 대해 데이터 전송 연산을 초기화할 수 있다. 만일 디바이스(102a 내지 102m)에 대한 참조가 디바이스 디렉터리(204)에 국부적으로 저장되지 않으면, SAN(104)에서 디바이스의 개수가 증가함에 따라, 디바이스(102a 내지 102m)를 검색하는 데 걸리는 시간도 상당히 증가한다. 또한, SAN(104)에서 중복 경로(redundant path)의 개수가 증가하면, 완전히 검색해야할 디바이스 파일의 개수도 증가할 수 있으므로, 디바이스(102a 내지 102m)를 검색하는 데 걸리는 시간이 증가한다.

추가적으로, 소정의 실시예에서 디바이스 디렉터리(204)는 독립적 운영 체제이며, 즉, 디바이스 디렉터리는 복수의 운영 체제의 파일 시스템에 저장될 수 있다. 호스트(100a 내지 100n)가 이종 운영 체제를 갖는 실시예에서 디바이스 디렉터리(204)가 독립적 운영 체제이면, 분산 애플리케이션(110)은 디바이스 디렉터리(204)에 액세스할 수 있다.

도 3은 소정의 실시예에 따라서, 분산 애플리케이션(110)이 디바이스 디렉터리(204)를 사용하여 SAN(104)에서 복수의 디바이스에 액세스하는 방법을 설명하는 블록도를 도시한다.

분산 애플리케이션(110)은 디바이스에 대해 데이터 전송 연산을 수행할 필요가 있다. 소정의 실시예에서, 분산 애플리케이션(110)은 호스트(200) 내의 디바이스 디렉터리(204)를 통해 디바이스 파일 링크(206a 내지 206p)에 액세스한다. 소정의 실시예에서, 디바이스 파일 링크(206a 내지 206p)는 디바이스(102a 내지 102p)에 대응하는 디바이스 파일(300a 내지 300p)을 참조할 수 있다. 소정의 실시예에서, 디바이스(102a 내지 102p)는 도 1에 도시된 디바이스(102a 내지 102m)의 서브세트일 수 있다. 예컨대, 디바이스 파일 링크(206a)는 디바이스 파일(300a)을 참조할 수 있고, 디바이스 파일 링크(206p)는 디바이스 파일(300p)을 참조할 수 있다. 소정의 실시예에서, 디바이스 파일(300a 내지 300p)은 SAN(104)을 통해 저장 디바이스(102a 내지 102p)까지의 특정 개별 경로 또는 저장 디바이스(102a 내지 102p)까지의 경로의 선택을 나타낼 수 있다. 소정의 실시예에서 저장 디바이스(102a 내지 102p)는 IBM Enterprise Storage Server®과 같은 저장 서버에 의해 제공되는 가상 저장 디바이스를 포함할 수 있다.

소정의 실시예에서, 분산 애플리케이션(110)은 디바이스 파일 링크(206a, 206p)와 같은 디바이스 파일 링크를 결정할 수 있다. 분산 애플리케이션(110)은 디바이스 파일(300a, 300p)에 대해 개방(302a, 304a), 폐쇄(302b, 304b), 갱신(302c, 304c), 판독(도시 생략), 기록(도시 생략), 첨부(도시 생략) 등과 같은 다양한 연산을 수행할 수 있다. 예컨대, 분산 애플리케이션(110)은 디바이스(102a)에 대해 데이터 전송 연산을 초기화하는 디바이스 파일(300a)을 개방(302a)하기 위해 디바이스 파일 링크(206a)를 사용할 수 있다.

따라서, 도 3은 분산 애플리케이션(110)이 디바이스 디렉터리(204)를 사용함으로써 SAN(104) 내의 디바이스(102a 내지 102p)에 액세스하는 실시예를 도시한다.

도 4는 본 발명의 소정의 실시예에 따라서, 디바이스 디렉터리(204)를 생성하고 이 디바이스 디렉터리(204)를 사용함으로써 SAN(104) 내의 디바이스에 대해 I/O 연산을 수행하는 연산을 도시한다. 도 4에 설명된 연산은 도 1에 도시된 컴퓨팅 환경에서 구현될 수 있다.

제어는 디바이스 디렉터리(204)가 호스트(200)와 같은 호스트에서 생성되는 블록(400)에서 시작한다. 디바이스 디렉터리(204)는 디바이스 디렉터리(108a 내지 108n) 중 임의의 디렉터리를 나타낼 수 있고, 호스트(200)는 대응하는 호스트(100a 내지 100n)를 나타낼 수 있다. 호스트 내의 디바이스 디렉터리(204)의 생성은 호스트 또는 관리 서버(106)에 의해 수행될 수 있다. 소정의 실시예에서 분산 애플리케이션(110)은 디바이스 디렉터리(204)를 생성할 수 있다.

디바이스 디렉터리(204)가 디바이스 파일 링크(206a 내지 206p)와 같은 디바이스 파일 링크로 배치되거나 갱신될 필요가 있는지 여부를 판단한다(블록 402). 예컨대, 디바이스 디렉터리(204)가 비어있거나, 호스트(200) 내의 프로세스가 디바이스 디렉터리(204)에 없는 디바이스에 대한 액세스를 요청하면, 디바이스 디렉터리(204)가 배치(populated)되거나 갱신될 필요가 있다. 소정의 실시예에서, 디바이스 발견이 수행될 필요가 있을 때 또는 다른 호스트나 관리 서버(106)가 디바이스 디렉터리(204)로의 갱신을 포함할 수 있는 메시지 전송을 시작할 때, 디바이스 디렉터리(204)는 주기적인 간격으로 배치되거나 갱신될 필요가 있다. 만일 디바이스 디렉터리가 배치되거나 갱신될 필요가 있으면, 디바이스 디렉터리(204)는 블

록(404a, 404b, 404c) 중 하나 이상에 설명된 연산의 수행 및 블록(406)에 설명된 후속하는 연산의 수행에 의해 배치되거나 갱신될 수 있다. 소정의 실시예에서, 프로세스는 디바이스 디렉터리(204)가 갱신되거나 존재할 필요가 있다고 판단할 때까지 블록(402)에서 대기할 수 있다.

소정의 실시예에서, 호스트(200)에서 실행하는 분산 애플리케이션(110)은 디렉터리(204)를 배치하거나 갱신하라는 메시지를 관리 서버(106)로부터 수신할 수 있다(블록 404a). 호스트(200)에서 실행하는 분산 애플리케이션(110)은 SAN(104) 내의 관심대상 중 하나 이상의 디바이스(102a 내지 102m)도 발견할 수 있다(블록 404b). 호스트(200)에서 실행하는 분산 애플리케이션(110)은 SAN(104) 내의 다른 호스트로부터 디바이스 디렉터리(204)를 배치하고 갱신하라는 하나 이상의 메시지도 수신할 수 있다(블록 404c). 예컨대, 소정의 실시예에서 분산 애플리케이션(110)을 실행하는 호스트(100a)는 디바이스 디렉터리(108a)를 배치하고 갱신하라는 메시지를 호스트(100b)로부터 수신할 수 있다. 호스트(100a)에 의해 수신된 메시지는 수신하는 호스트(100a)가 관심대상 중 대응하는 디바이스를 찾을 수 있게 하는 정보를 포함할 수 있다. 예컨대, 이 정보는 디바이스의 WWPN(World Wide Port Name)을 포함할 수 있는데, 여기서 수신하는 호스트(100a)는 저장 드라이버와 관련된 디바이스의 WWPN을 사용하여 관심대상의 디바이스 파일을 찾을 수 있다. 이어서, 수신하는 호스트(100a)의 디바이스 디렉터리(108a)에서 대응하는 디바이스 파일에 대한 링크가 생성될 수 있다.

분산 애플리케이션(110)은 블록(404a, 404b, 404c)에서 수행되는 디바이스 발견 또는 수신되는 메시지에 기초하여 대응하는 디바이스에 대한 디바이스 파일 링크로 디바이스 디렉터리(204)를 배치하거나 갱신할 수 있다(블록 406). 예컨대, 소정의 실시예에서, 분산 애플리케이션(110)은 디바이스(102a 내지 102p)에 대응하는 디바이스 파일(300a 내지 300p)을 참조하는 디바이스 파일 링크(206a 내지 206p)로 디바이스 디렉터리(204)를 배치하거나 갱신할 수 있다. 따라서, 소정의 실시예에서 디바이스 디렉터리(204)의 배치 및 갱신은 분산 애플리케이션(110)에 의해 수행될 수 있다. 다른 소정의 실시예에서, 분산 애플리케이션(110)과 다른 애플리케이션이 디바이스 디렉터리를 배치하거나 갱신할 수 있다.

분산 애플리케이션(110)은 선택된 디바이스에 대해 연산이 수행될 것인지의 여부를 판단한다(블록 408). 만일 연산이 수행된다면, 분산 애플리케이션(110)은 디바이스 디렉터리(204)로부터 선택된 디바이스에 대응하는 디바이스 파일에 액세스함으로써 선택된 디바이스에 대해 연산을 수행하고(블록 410), 제어는 디바이스 디렉터리(204)를 배치하거나 갱신하는 블록(402)으로 돌아간다. 예컨대, 소정의 실시예에서, 분산 애플리케이션(110)은 디바이스 디렉터리(204) 내의 디바이스 파일 링크(206p)를 사용함으로써 디바이스(102p)에 대응하는 디바이스 파일(300p)의 개방(304a)을 수행할 수 있다.

만일 분산 애플리케이션(110)이 선택된 디바이스에 대해 어떠한 연산도 수행되지 않았다고 판단하면(블록 408), 제어는 디바이스 디렉터리(204)를 배치하거나 갱신하는 블록(402)으로 돌아간다. 소정의 실시예에서, 블록(402, 404a, 404b, 404c, 406, 408, 410)에 설명된 프로세스는 호스트(200)에서 반복하여 실행될 수 있다. 다른 실시예에서, 예외, 에러 상태, 셧다운 또는 호스트(200)의 재부팅은 도 4에 설명된 프로세스를 종료할 수 있다.

따라서, 도 4는 디바이스(102a 내지 102p)에 대응하는 디바이스 파일(300a 내지 300p)에 대한 참조를 포함하는 디바이스 디렉터리(204)가 생성, 배치 및 갱신되는 실시예를 설명한다. 디바이스 디렉터리(204)가 위치하는 호스트는 분산 애플리케이션(110)이 디바이스 디렉터리(204)를 사용함으로써 디바이스(102a 내지 102p)에 대해 연산을 수행하게 한다. 디바이스 디렉터리(204)는 호스트(200)에 국부적으로 저장되므로, 분산 애플리케이션(110)은 디바이스에 대한 참조가 호스트(200)로부터 멀리 위치하는 구현예에 비해 더 빨리 디바이스에 액세스할 수 있다. 따라서, 소정의 실시예에서 호스트 운영 체제와 같은 시스템 소프트웨어(202)는 디바이스 파일을 관리할 수 있고, 분산 애플리케이션은 디바이스 디렉터리(204)를 관리할 수 있다.

도 5는 소정의 실시예에 따라서, 분산 애플리케이션(110)이 디바이스 디렉터리(204)를 사용하여 SAN(104) 내의 디바이스(102a 내지 102m)에 대해 I/O 연산을 수행하게 하기 위해 호스트(200)와 같은 호스트에서 수행되는 연산을 도시한다.

제어는 연산 장치(200)가 복수의 디바이스(102a 내지 102p)에 대응하는 복수의 파일(300a 내지 300p)에 대한 참조(206a 내지 206p)를 연산 장치(200)에서 실행되는 데이터 구조체(204)에 저장하는 블록(500)에서 시작하는데, 연산 장치(200)는 네트워크(104)를 통해 복수의 디바이스(102a 내지 102p)에 결합된다.

연산 장치(200)는 분산 애플리케이션(110)이 데이터 구조체(204)에 액세스하는 것을 허용하는데, 여기서 분산 애플리케이션(110)은 데이터 구조체(204) 내의 저장된 참조를 사용하여 디바이스에 대응하는 파일을 결정하고, 결정된 파일을 통해 디바이스에 대하여 데이터 전송 연산을 수행한다(블록 502).

따라서, 도 5는 호스트(200)와 같은 연산 장치가 분산 애플리케이션(110)이 디바이스 디렉터리(204)를 사용하여 데이터 전송 연산을 수행하게 하는 방법을 도시한다.

소정의 실시예에서, SAN 내의 디바이스에 대한 지식은 호스트에 국부적으로 캐싱되어(cashed), 후보 또는 사용된 디바이스는 관리자 또는 분산 애플리케이션(110)에 신속하게 액세스 및 인식할 수 있다. 소정의 실시예에서, 지정된 디렉터리는 SAN 환경 내의 복수의 호스트를 통해 분산 애플리케이션에 의해 공유되는 디바이스를 관리하는 플랫폼-독립형 기술, 공급자-독립형 기술로서 사용되는데, 지정된 디렉터리를 호스트에 국부적으로 저장함으로써 적절한 디바이스에 대한 스캔 시간 및 복잡성이 감소한다. 호스트가 중대한 연산을 수행하지 않으면, 지정된 디렉터리는 디바이스에 대한 참조로 갱신될 수 있는데, 여기서 중대한 연산은 가능한 한 빨리 완료되어야하는 연산이다. 소정의 실시예에서, 분산 애플리케이션(110)은 디바이스 디렉터리(108a 내지 108n)에 정보를 캐싱하는데, 캐싱된 정보는 액세스될 수 있는 디바이스 또는 분산 애플리케이션(100)이나 다른 애플리케이션에 의해 이미 사용 중인 디바이스와 관련될 수 있다. 소정의 실시예에서, 디바이스 디렉터리(108a 내지 108n)의 위치는 분산 애플리케이션(110)에 의해 지정될 수 있다. 다른 소정의 실시예에서, 디바이스 디렉터리는 임의의 저장 디바이스 공급자 또는 범용 디바이스 파일을 저장하는 임의의 호스트 시스템 소프트웨어에 의해 사용되지 않는다. 다른 소정의 실시예에서, 관리자는 관리 서버(106)를 사용하여 디바이스 디렉터리(108a 내지 108n)를 수동으로 구성할 수 있다. 디바이스 디렉터리(108a 내지 108n)를 구성하는 자동화 스크립트도 관리 서버(106)에서 실행될 수 있다. 추가적으로, 다른 실시예에서는 관리 서버(106) 외에 호스트(100a 내지 100n)에서 관리할 수 있다. 예컨대, 관리자는 선택된 호스트에 로그온하고, 분산 애플리케이션(110)에 의해 사용되는 디바이스 디렉터리(204)에 새로운 링크를 추가하며, 선택된 호스트에서 분산 애플리케이션(110)이 사용할 수 있는 디바이스를 분산 애플리케이션(110)에게 알려줄 수 있다.

소정의 실시예에서, 분산 애플리케이션(110)은 중대한 연산 동안 디바이스(102a 내지 102m)를 스캔하는 데 필요한 시간을 감소시키는 것이 가능해진다. 다른 소정의 실시예에서, 호스트(100a 내지 100n), 분산 애플리케이션(110), 또는 관리 서버(106)는 중대한 연산이 프로세싱되지 않으면 SAN(104) 내의 디바이스를 스캐닝함으로써 디바이스 디렉터리(108a 내지 108n)를 생성, 배치 또는 갱신할 수 있다.

소정의 실시예에서, 분산 애플리케이션(110)은 일반적인 방식으로 디바이스(102a 내지 102m)와 상호작용한다. 다른 소정의 실시예에서, 분산 애플리케이션(110)은 분산 애플리케이션이 설계, 테스트 또는 기록되었던 시간에 이용가능하지 않았던 디바이스(102a 내지 102m)를 사용할 수 있다. 다른 실시예에서는, 분산 애플리케이션(110)에 의해 도 1 내지 도 5에 설명된 연산 이외의 추가 연산이 이용되어 디바이스를 배치할 수 있다. 예컨대, 분산 애플리케이션(110)은 다른 디바이스 위치를 검색하거나, 특정 공급자 디바이스 또는 드라이버를 사용 또는 선호하는 데 적합해지거나, 호스트에서 운영 체제와 상호작용하여 데이터 전송을 위한 디바이스를 결정할 수 있다. 소정의 실시예에서, 디바이스(102a 내지 102m)는 시스템 소프트웨어(202)를 사용하지 않고 디바이스 디렉터리에서 사용하기 위해 라벨링된다. 복수의 호스트(100a 내지 100n) 중 각 호스트 상의 시스템 소프트웨어가 서로 다를 수 있으므로, 각 호스트 상의 시스템 소프트웨어에 의해 디바이스(102a 내지 102m)를 라벨링하는 것은 충돌을 야기할 수 있고, 분산 애플리케이션(110)에 의한 디바이스(102a 내지 102m)의 사용을 방해할 수 있다.

설명된 기술은 소프트웨어, 펌웨어, 마이크로-코드, 하드웨어 및/또는 이들의 임의의 조합을 포함하는 방법, 장치 또는 제조물로서 구현될 수 있다. 여기에 설명된 용어 "제조물"은 프로그램 인스트럭션, 회로소자에서 실행되는 코드 및/또는 로직(예컨대, 집적 회로 칩, PGA(Programmable Gate Array), ASIC 등) 및/또는 컴퓨터 판독가능한 매체(예컨대, 하드 디스크 드라이브, 플로피 디스크, 테이프와 같은 자기 저장 매체), 광학 저장장치(예컨대, CD-ROM, DVD-ROM, 광디스크 등), 휘발성 메모리 디바이스 및 비휘발성 메모리 디바이스(예컨대, EEPROM(Electrically Erasable Programmable Read Only Memory), ROM(Read Only Memory), PROM(Programmable Read Only Memory), RAM(Random Access Memory), DRAM(Dynamic Random Access Memory), SRAM(Static Random Access Memory), 플래시, 펌웨어, 프로그램가능한 로직 등)를 지칭한다. 컴퓨터 판독가능한 매체 내의 코드는 프로세서와 같은 정보 처리 장치에 의해 액세스 및 실행될 수 있다. 소정의 실시예에서, 실시예가 이루어지는 코드는 전송 매체를 통해서 또는 네트워크를 거쳐 파일 서버로부터 추가로 액세스될 수 있다. 이러한 경우에, 코드가 실행되는 제조물은 네트워크 전송선, 무선 전송 매체, 공간을 통한 신호 전달, 전파, 적외선 신호 등과 같은 전송 매체를 포함할 수 있다. 물론, 당업자는 실시예의 범주를 벗어나지 않으면서 다수의 변경이 이루어질 수 있으며, 제조물이 당해 기술에 알려진 임의의 정보 저장 매체를 포함할 수 있음을 알 것이다. 예컨대, 제조물은 정보 처리 장치에 의해 실행될 때 연산이 수행되게 하는 인스트럭션이 자체 내에 저장된 저장 매체를 포함한다.

도 6은 소정의 실시예가 구현될 수 있는 컴퓨터 아키텍처(600)의 블록도를 도시한다. 소정의 실시예에서, 호스트(100a 내지 100n) 및 관리 서버(106)는 컴퓨터 아키텍처(600)를 따라서 구현될 수 있다. 컴퓨터 아키텍처(600)는 프로세서 또는 회로소자(602), 메모리(604)(예컨대, 휘발성 메모리 디바이스) 및 저장장치(606)를 포함할 수 있다. 컴퓨터 아키텍처(600)의 특정 구성 요소는 호스트(100a 내지 100n) 및 관리 서버(106)에서 발견될 수도 발견되지 않을 수도 있다. 저장장치(606)는 비휘발성 메모리 디바이스(예컨대, EEPROM, ROM, PROM, RAM, DRAM, SRAM, 플래시, 펌웨어, 프로그램

가능한 로직 등), 자기 디스크 드라이브, 광디스크 드라이브, 테이프 드라이브 등을 포함할 수 있다. 저장장치(606)는 내부 저장 디바이스, 부속형 저장 디바이스 및/또는 네트워크 액세스가능한 저장 디바이스를 포함할 수 있다. 저장장치(606) 내의 프로그램은 메모리(604)에 로딩되어 프로세서(602)에 의해 실행될 수 있다. 소정의 실시예에서, 회로소자(602)는 메모리(604)와 통신할 수 있고, 연산을 수행할 수 있다. 추가적으로, 아키텍처는 SAN(104)과 같은 네트워크와 통신할 수 있게 하는 네트워크 카드(608)를 포함할 수 있다. 아키텍처는 키보드, 터치스크린, 펜, 음성 인식 입력 등과 같은 적어도 하나의 입력 디바이스(610) 및 디스플레이 디바이스, 스피커, 프린터 등과 같은 적어도 하나의 출력 디바이스(612)도 포함할 수 있다.

도 4 및 도 5의 연산 중 적어도 몇몇은 순차적으로뿐만 아니라 병렬적으로도 수행될 수 있다. 다른 실시예에서, 연산 중 몇몇은 서로 다른 순서로 수행, 수정 또는 제거될 수 있다.

또한, 다수의 소프트웨어 부품 및 하드웨어 부품은 설명하기 위해 별도의 모듈로 설명되었다. 이러한 부품은 더 적은 개수의 부품으로 집적되거나 더 많은 개수의 부품으로 분할될 수 있다. 추가적으로, 특정 부품에 의해 수행되는 것으로 설명된 소정의 연산은 다른 부품에 의해서도 수행될 수 있다.

도 1 내지 도 6에 도시되거나 지칭된 데이터 구조체 및 부품은 특정 유형의 정보를 구비하는 것으로 설명된다. 다른 실시예에서, 데이터 구조체 및 부품은 서로 다르게 구성될 수 있고, 도면에 도시되거나 지칭된 것보다 더 적/많은 서로 다른 필드 또는 서로 다른 기능을 구비할 수 있다.

따라서, 상술한 실시예에 대한 설명은 예시하고 설명하기 위한 것이다. 개시된 형태로 실시예를 제한하거나 사용하려는 것은 아니다. 이상의 개시 내용으로 보아 다수의 변경 및 수정이 가능하다.

IBM, ESCON 및 Enterprise Storage Server는 IBM사의 상표로 등록되었다.

도면의 간단한 설명

도 1은 소정의 실시예에 따른, 컴퓨팅 환경의 블록도를 도시한다.

도 2는 소정의 실시예에 따른, 디바이스 디렉터리를 포함하는 호스트의 블록도를 도시한다.

도 3은 소정의 실시예에 따라서, 분산 애플리케이션이 디바이스 디렉터리를 사용하여 SAN의 복수의 디바이스에 액세스하는 방법을 설명하는 블록도를 도시한다.

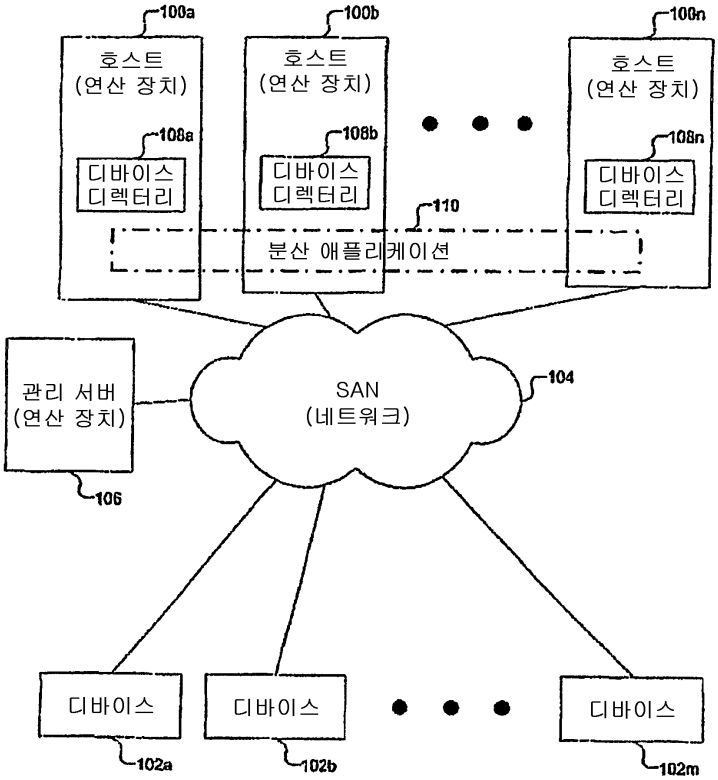
도 4는 소정의 실시예에 따라서, 디바이스 디렉터리를 생성하고, 이 디바이스 디렉터리를 사용하여 SAN의 디바이스에 대해 입/출력(I/O) 연산을 수행하는 연산을 도시한다.

도 5는 소정의 실시예에 따라서, 분산 애플리케이션이 디바이스 디렉터리를 사용하여 SAN의 디바이스에 대해 I/O 연산을 수행하게 하기 위해 호스트에서 수행되는 연산을 도시한다.

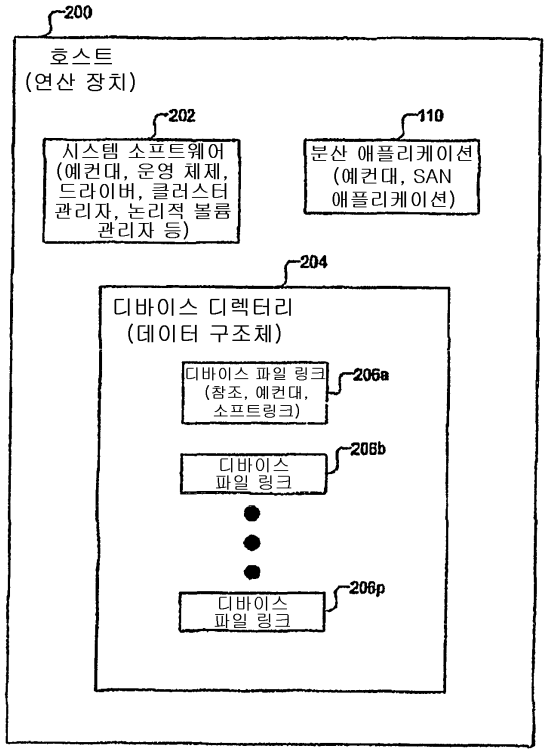
도 6은 소정의 실시예가 구현되는 컴퓨팅 아키텍처를 도시한다.

도면

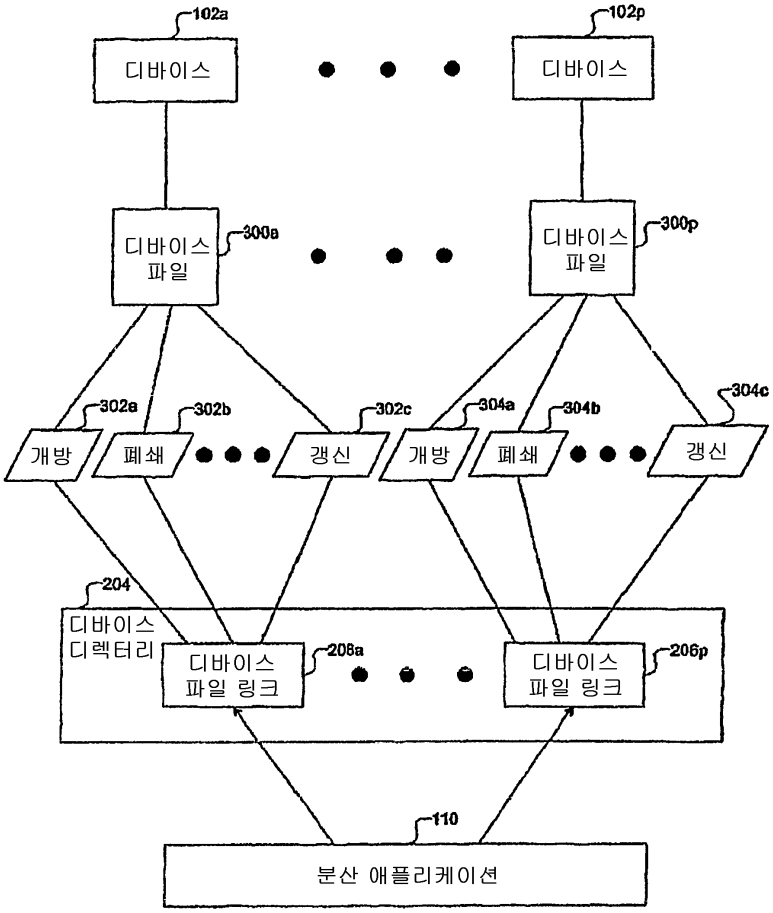
도면1


100a
호스트
(연산 장치)
108a
디바이스
디렉터리
100b
호스트
(연산 장치)
108b
디바이스
디렉터리
100n
호스트
(연산 장치)
108n
디바이스
디렉터리
110
분산 애플리케이션
관리 서버
(연산 장치)
106
SAN
(네트워크)
104
디바이스
102a
디바이스
102b
디바이스
102m

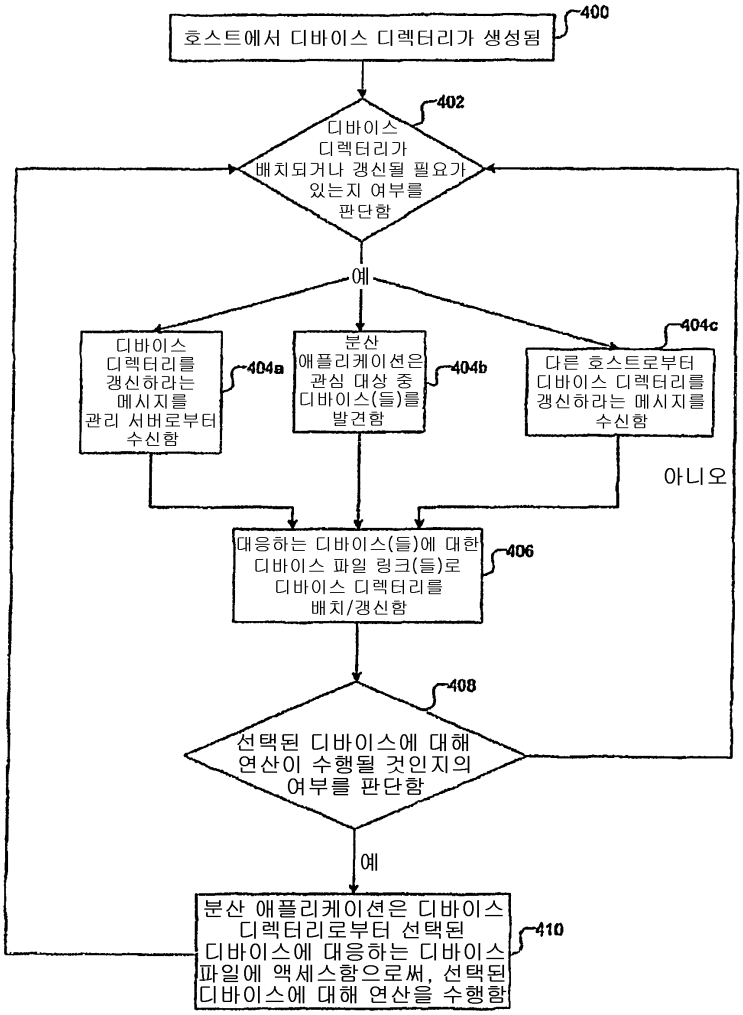
도면2


200
호스트
(연산 장치)
202
시스템 소프트웨어
(예컨대, 운영 체제,
드라이버, 클러스터
관리자, 논리적 볼륨
관리자 등)
110
분산 애플리케이션
(예컨대, SAN
애플리케이션)
204
디바이스 디렉터리
(데이터 구조체)
디바이스 파일 링크
(참조, 예컨대,
소프트링크)
206a
디바이스
파일 링크
206b
디바이스
파일 링크
206p

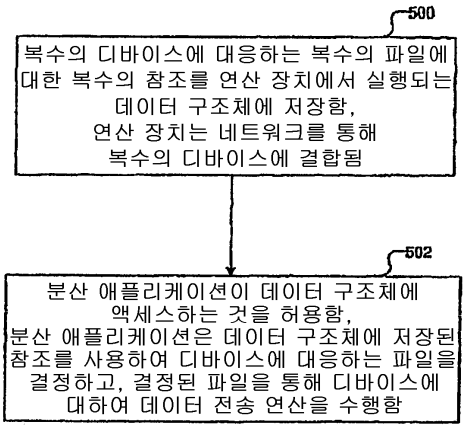
도면3


102a
디바이스
102p
디바이스
디바이스 파일
300a
디바이스 파일
300p
302a
개방
302b
폐쇄
302c
갱신
304a
개방
304b
폐쇄
304c
갱신
204
디바이스 디렉터리
디바이스 파일 링크
208a
디바이스 파일 링크
206p
110
분산 애플리케이션

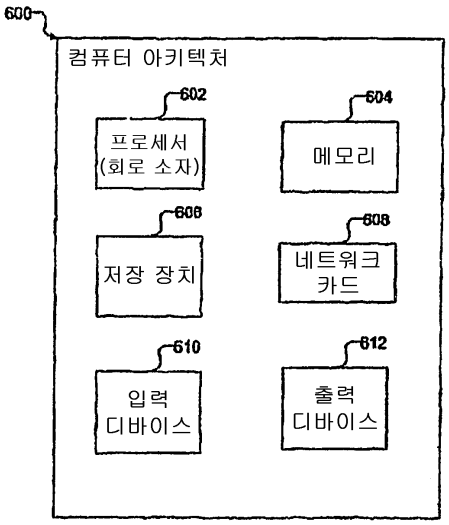
도면4


400
호스트에서 디바이스 디렉터리가 생성됨
402
디바이스 디렉터리가 배치되거나 갱신될 필요가 있는지 여부를 판단함
예
404a
디바이스 디렉터리를 갱신하라는 메시지를 관리 서버로부터 수신함
404b
분산 애플리케이션은 관심 대상 중 디바이스(들)를 발견함
404c
다른 호스트로부터 디바이스 디렉터리를 갱신하라는 메시지를 수신함
아니오
406
대응하는 디바이스(들)에 대한 디바이스 파일 링크(들)로 디바이스 디렉터리를 배치/갱신함
408
선택된 디바이스에 대해 연산이 수행될 것인지의 여부를 판단함
예
410
분산 애플리케이션은 디바이스 디렉터리로부터 선택된 디바이스에 대응하는 디바이스 파일에 액세스함으로써, 선택된 디바이스에 대해 연산을 수행함

도면5


500
복수의 디바이스에 대응하는 복수의 파일에 대한 복수의 참조를 연산 장치에서 실행되는 데이터 구조체에 저장함, 연산 장치는 네트워크를 통해 복수의 디바이스에 결합됨
502
분산 애플리케이션이 데이터 구조체에 액세스하는 것을 허용함, 분산 애플리케이션은 데이터 구조체에 저장된 참조를 사용하여 디바이스에 대응하는 파일을 결정하고, 결정된 파일을 통해 디바이스에 대하여 데이터 전송 연산을 수행함

도면6


600
컴퓨터 아키텍처
602
프로세서
(회로 소자)
604
메모리
606
저장 장치
608
네트워크
카드
610
입력
디바이스
612
출력
디바이스