(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl.
H04L 12/24 (2006.01)
H04L 29/06 (2006.01)
H04L 12/28 (2006.01)

(45) 공고일자 2007년02월09일
(11) 등록번호 10-0680626
(24) 등록일자 2007년02월02일

(21) 출원번호 10-2005-7008956
(22) 출원일자 2005년05월18일
심사청구일자 2005년11월10일
번역문 제출일자 2005년05월18일
(86) 국제출원번호 PCT/EP2003/050895
국제출원일자 2003년11월25일
(65) 공개번호 10-2005-0086732
(43) 공개일자 2005년08월30일
(87) 국제공개번호 WO 2004/057798
국제공개일자 2004년07월08일

(30) 우선권주장 02102852.7 2002년12월20일 유럽특허청(EPO)(EP)

(73) 특허권자 인터내셔널 비지네스 머신즈 코포레이션
미국 10504 뉴욕주 아몬크 뉴오차드 로드

(72) 발명자 라이쉬 크리스토프
독일 겔링겐 70839 프리드리히-샤페르트-스트라쎄 21

(74) 대리인 김태홍
신정건

심사관 : 신성길

전체 청구항 수 : 총 10 항

(54) 비신뢰 서버 환경에서 ＳＡＮ 관리용 보안 시스템 및 방법

(57) 요약

본 발명에 따르면, 복수의 서버가 하나의 광채널 어댑터를 공유하는 서버 환경에서 스토리지 영역 네트워크(SAN)를 운영하는 방법 및 시스템이 제공된다. SAN 관리 서버는 스토리지 시스템 내의 영역과 보안을 관리하고, 광채널 네트워크는 스토리지 장치로의 접속을 제공하며, 복수의 운영 체제 이미지는 상기 서버 환경에서 동작한다. 또한, 신뢰 SAN 관리 클라이언트 유닛은 상기 SAN 관리 서버에 접속되고 광채널 어댑터는 상기 신뢰 SAN 관리 클라이언트 유닛을 인증하도록 구성됨으로써, 상기 신뢰 SAN 관리 클라이언트 유닛은 상기 운영 체제 이미지 각각에 대신하여 상기 광채널 네트워크에서 명령을 발생하도록 구성된다.

대표도

도 1

특허청구의 범위

청구항 1.

복수의 서버가 하나의 광채널 어댑터를 공유하는 서버 환경에서 스토리지 영역 네트워크(SAN)를 운영하는 시스템에 있어서,

SAN 관리(SM; SAN Management) 서버;

스토리지 장치로의 접속을 제공하는 광채널 네트워크; 및

상기 서버 환경에서 실행되는 복수의 운영 체제 이미지를 포함하고,

상기 SM 서버에 접속된 신뢰 SAN 관리(SM; SAN Management) 클라이언트 유닛과, 광채널 어댑터(FC 어댑터)를 특징으로 하며,

상기 신뢰 SM 클라이언트 유닛은, 각각의 상기 운영 체제 이미지(OS 이미지) 대신에 광채널 네트워크에서 명령을 내리도록 구성되는 것인, 스토리지 영역 네트워크 운영 시스템.

청구항 2.

제1항에 있어서, 상기 SM 서버는 제1 명령 세트와 제2 명령 세트를 구별하도록 구성되어, 상기 제1 명령 세트는 상기 SAN과 함께 상기 SM 클라이언트에 의해 처리되고, 상기 제2 명령 세트는 상기 SAN으로의 액세스 없이 상기 OS 이미지에 의해 처리되는 것인, 스토리지 영역 네트워크 운영 시스템.

청구항 3.

제1항에 있어서, 상기 SM 클라이언트는 제1 명령 세트와 제2 명령 세트를 구별하도록 구성되어, 상기 제1 명령 세트는 상기 SAN과 함께 상기 SM 클라이언트에 의해 처리되고, 상기 제2 명령 세트는 상기 SAN으로의 액세스 없이 상기 OS 이미지에 의해 처리되는 것인, 스토리지 영역 네트워크 운영 시스템.

청구항 4.

제1항 내지 제3항 중 어느 한 항에 있어서, 상기 서버 환경은 가상 서버를 포함하는 것인, 스토리지 영역 네트워크 운영 시스템.

청구항 5.

제1항 내지 제3항 중 어느 한 항에 있어서, 상기 서버 환경은 분할된 서버를 포함하는 스토리지 영역 네트워크 운영 시스템.

청구항 6.

청구항 6은(는) 설정등록료 납부시 포기되었습니다.

제1항 내지 제3항 중 어느 한 항에 있어서, 상기 광채널 어댑터(FC 어댑터)는 상기 신뢰 SM 클라이언트 유닛을 인증하도록 구성된 것인, 스토리지 영역 네트워크 운영 시스템.

청구항 7.

청구항 7은(는) 설정등록료 납부시 포기되었습니다.

제1항 내지 제3항 중 어느 한 항에 있어서, 상기 FC 어댑터와 상기 SAN은 상기 비신뢰 OS 이미지의 액세스를 최소의 필요 명령 세트로 제한할 수 있는 것인, 스토리지 영역 네트워크 운영 시스템.

청구항 8.

제1항 내지 제3항 중 어느 한 항에 있어서, 상기 FC 어댑터와 상기 서버의 가상층은 상기 비신뢰 OS 이미지의 액세스를 최소의 필요 명령 세트로 제한할 수 있는 것인, 스토리지 영역 네트워크 운영 시스템.

청구항 9.

청구항 9은(는) 설정등록료 납부시 포기되었습니다.

제1항 내지 제3항 중 어느 한 항에 있어서, 상기 서버 부하를 적게 유지하도록 단 하나의 SM 클라이언트가 제공되는 것인, 스토리지 영역 네트워크 운영 시스템.

청구항 10.

청구항 10은(는) 설정등록료 납부시 포기되었습니다.

제8항에 있어서, 리던던시를 제공하도록 하나 이상의 백업 SM 클라이언트가 제공되는 것인, 스토리지 영역 네트워크 운영 시스템.

청구항 11.

청구항 11은(는) 설정등록료 납부시 포기되었습니다.

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 SAN으로부터의 메시지를 수신하기 위해 상기 SM 클라이언트만이 등록되고, 상기 SM 클라언트는 상기 SM 서버에만 상기 메시지를 포워딩하도록 구성되는 것인, 스토리지 영역 네트워크 운영 시스템.

청구항 12.

청구항 12은(는) 설정등록료 납부시 포기되었습니다.

제1항 내지 제3항 중 어느 한 항에 있어서, 상기 FC 어댑터는, 상기 SM 클라이언트뿐만 아니라 비신뢰 OS 이미지에게도 등록할 필요없는 SAN에 의해 생성된 모든 메시지를 전송하도록 구성되는 것인, 스토리지 영역 네트워크 운영 시스템.

청구항 13.

청구항 13은(는) 설정등록료 납부시 포기되었습니다.

제1항 내지 제3항 중 어느 한 항에 있어서, 상기 FC 어댑터는, 원래 메시지를 상기 비신뢰 OS 이미지에 전송하는 것에 추가하여, 상기 SM 클라이언트에게 등록할 필요없는 SAN에 의해 발생된 모든 메시지의 복사복을 포워딩하도록 구성되는 것인, 스토리지 영역 네트워크 운영 시스템.

청구항 14.

제1항 내지 제3항 중 어느 한 항에 있어서, 상기 서버는 두개의 에이전트 클래스(two classes of agents), 즉, 상기 SM 클라이언트와 원격 액세스 서버(RA 서버)가 구비된 것인, 스토리지 영역 네트워크 운영 시스템.

청구항 15.

청구항 15은(는) 설정등록료 납부시 포기되었습니다.

제14항에 있어서, 상기 SM 서버는 상기 RA 서버에 액세스하기 위한 인증 데이터를 보유하는 저장소가 구비된 것인, 스토리지 영역 네트워크 운영 시스템.

청구항 16.

청구항 16은(는) 설정등록료 납부시 포기되었습니다.

제14항에 있어서, 상기 SM 클라이언트는 상기 RA 서버를 액세스하기 위한 인증 데이터를 보유하는 저장소가 구비된 것인, 스토리지 영역 네트워크 운영 시스템.

청구항 17.

청구항 17은(는) 설정등록료 납부시 포기되었습니다.

제1항 내지 제3항 중 어느 한 항에 있어서, 상기 SM 클라이언트와 상기 FC 어댑터는, 각각의 비신뢰 OS 이미지에 의한 자원의 사용을 빌링하는데 사용되는 신뢰 정보를 수집하도록 구성된 것인, 스토리지 영역 네트워크 운영 시스템.

청구항 18.

청구항 18은(는) 설정등록료 납부시 포기되었습니다.

제1항 내지 제3항 중 어느 한 항에 있어서, 상기 SM 프레임워크는 상기 액세스 권한을 설정하기 위해서 방화벽 제어 애플리케이션과 통신하도록 되어 있는 것인, 스토리지 영역 네트워크 운영 시스템.

청구항 19.

청구항 19은(는) 설정등록료 납부시 포기되었습니다.

제1항 내지 제3항 중 어느 한 항에 있어서, 상기 SM 클라이언트는 상기 SM 서버로부터 상기 RA 서버로의 요청에 대해 라우터 기능을 하도록 되어 있는 것인, 스토리지 영역 네트워크 운영 시스템.

청구항 20.

청구항 20은(는) 설정등록료 납부시 포기되었습니다.

제1항 내지 제3항 중 어느 한 항에 있어서, 상기 RA 서버는 기존의 텔넷/sshd 서버에 의해 형성되는 것인, 스토리지 영역 네트워크 운영 시스템.

청구항 21.

복수의 운영 체제가 하나의 광채널 어댑터(Fibre Channel Adapter; FC 어댑터)를 공유하는 서버 환경에서 스토리지 영역 네트워크(SAN)를 운영하는 방법에 있어서,

상기 광채널 어댑터로의 통신 경로를 구비한 적어도 하나의 SAN 관리(SM; SAN Management) 서버와 적어도 하나의 SAN 관리(SM; SAN Management) 클라이언트를 이용하여 SAN 관리 소프트에 의해 상기 SAN을 관리하는 단계; 및

상기 SM 서버가 내린 요청을 적어도 두 그룹으로 분리하는 단계를 포함하며,

제1 그룹은, 신뢰 경로에 대응하여 동일한 어댑터를 공유하는 다른 운영 체제 대신에 상기 SM 클라이언트을 위하여 상기 SAN과 상기 광채널 어댑터에 의해 처리되고,

제2 그룹은 상기 광채널 어댑터 및 상기 SAN과의 요청의 송수신없이 다른 운영 체제들에 의해 처리되는 것인, 스토리지 영역 네트워크 운영 방법.

청구항 22.

제21항에 있어서, 상기 SAN 및 FC 어댑터에서 생성된 비요청 메시지에 포함된 모든 정보를, 상기 SM 클라이언트에 의해 상기 SAN 관리자에게 라우팅하는 단계를 더 포함하는, 스토리지 영역 네트워크 운영 방법.

청구항 23.

제21항에 있어서, 상기 HBA_API 결합 요청을 사용하여 상기 방화벽을 변형하는 단계를 더 포함하는, 스토리지 영역 네트워크 운영 방법.

청구항 24.

청구항 24은(는) 설정등록료 납부시 포기되었습니다.

제21항 내지 제23항 중 어느 한 항에 있어서, 다른 운영 체제 이미지에 의해 도청되거나 변형될 수 없도록 상기 SM 클라이언트로부터 상기 어댑터로의 상기 통신 경로를 운영하는 단계를 더 포함하는 스토리지 영역 네트워크 운영 방법.

청구항 25.

청구항 25은(는) 설정등록료 납부시 포기되었습니다.

제21항 내지 제23항 중 어느 한 항에 있어서, 상기 어댑터와 SAN에서 생성된 개별 운영 체제 이미지를 빌딩하는데 관련된 모든 정보를, 상기 신뢰 경로 상에서 상기 SAN 클라이언트를 통해서만 액세스하는 단계를 더 포함하는, 스토리지 영역 네트워크 운영 방법.

청구항 26.

청구항 26은(는) 설정등록료 납부시 포기되었습니다.

제21항 내지 제23항 중 어느 한 항에 있어서, 상기 제1 그룹으로부터 요청을 실행하기 위해 상기 SM 서버가 상기 SM 클라이언트에게 인증 데이터를 제공하는 단계를 더 포함하는 스토리지 영역 네트워크 운영 방법.

청구항 27.

청구항 27은(는) 설정등록료 납부시 포기되었습니다.

제21항 내지 제23항 중 어느 한 항에 있어서, 상기 제2 그룹으로부터 요청을 실행하기 위해 상기 SM 서버가 다른 OS 이미지들에게 인증 데이터를 제공하는 단계를 더 포함하는 스토리지 영역 네트워크 운영 방법.

청구항 28.

청구항 28은(는) 설정등록료 납부시 포기되었습니다.

제21항 내지 제23항 중 어느 한 항에 있어서, 상기 SAN에서 제한된 명령 세트만을 실행할 수 있도록 상기 OS 이미지를 운영하는 단계를 더 포함하는 스토리지 영역 네트워크 운영 방법.

청구항 29.

청구항 29은(는) 설정등록료 납부시 포기되었습니다.

제21항 내지 제23항 중 어느 한 항에 따른 방법을 컴퓨터가 수행하도록 하는 컴퓨터 판독가능 프로그램 수단을 포함하는, 컴퓨터 판독가능한 기록매체.

명세서

기술분야

본 발명은 스토리지 영역 네트워크에 관한 것이다. 구체적으로는, 본 발명은 복수의 서버가 하나의 광 채널 어댑터를 공유하는 서버 환경에서 스토리지 영역 네트워크를 운영하는 방법 및 시스템에 관한 것이다.

배경기술

광 채널은, 수십 킬로미터만큼 멀리 떨어질 수 있는 입출력(I/O) 장치와 호스트 시스템을 상호접속하는데 사용되는 고속, 풀 듀플렉스(full-duplex), 직렬 통신 기술이다. 이것은, SCSI 및 PCI에서 볼 수 있는 수율 및 신뢰성과 같은 종래 I/O 인터페이스의 최선의 특징을, 이더넷 및 토큰 링에서 볼 수 있는 접속성과 스케일가능성과 같은 네트워킹 인터페이스의 최선의 특징과 결합한다. 이것은 기존 명령의 전달을 위한 전송 메커니즘을 제공하고, 상당한 양의 프로세싱이 하드웨어에서 수행될 수 있도록 함으로써, 고성능을 달성하는 아키텍처를 제공한다. 이것은 SCSI 및 IP 등의 레거시 프로토콜 및 드라이버와 함께 동작할 수 있어서, 기존의 인프라구조에 용이하게 도입될 수 있다.

광채널은 정보의 소스와 사용자 간에 정보를 전달한다. 이 정보는 명령, 제어, 파일, 그래픽 비디오 및 음성을 포함할 수 있다. 광채널 접속은, I/O 장치, 호스트 시스템 및 이들을 상호접속하는 네트워크 상에 존재하는 광채널 포트 사이에 설정된다. 이 네트워크는 광채널 포트를 상호접속하는데 사용되는 스위치, 허브, 브리지 및 리피터와 같은 요소로 구성된다.

광채널 아키텍처에는 3개의 광채널 토폴로지가 정의되어 있다. 이들은 포인트-대-포인트(Point-to-Point), 스위치 패브릭(Switched Fabric), 및 중재 루프(Arbitrated Loop)이다.

또한, 광채널 스위치(또는 스위치 패브릭)는 통상 구역화(Zoning)로 불리는 기능을 포함한다. 이들 기능들은 사용자가 스위치 포트를 포트 그룹으로 분할할 수 있게 한다. 포트 그룹, 즉, 구역(zone) 내의 포트는 동일 포트 그룹(구역) 내의 다른 포트들과만 통신할 수 있다. 구역화를 사용함으로써, 한 그룹의 호스트 및 장치로부터의 I/O는 임의의 다른 그룹에서 완전히 분리될 수 있기 때문에, 그룹 간의 임의의 간섭 가능성을 방지한다.

이것은 "소프트 구역화(soft zoning)"라고도 불린다. 이러한 소프트 구역화에서는, 사용자가 노드의 월드 와이드 네임(World Wide Name) - 월드 와이드 포트 네임(WWPN) 또는 월드 와이드 노드 네임(WWNN) - 에 따라, 노드를 구역에 할당하는 식으로 운영된다. 네임 서버는 이 정보를 캡처하며, 이것은 스위치 내에 임베디드된 기능이다. 그 다음, 포트가 어느 노드와 접속할 수 있는지를 알기 위해 네임 서버와 통신할 때마다, 그 네임 서버는 이 포트의 구역(zone) 내에 있는 노드에만 응답할 것이다.

표준 광채널 장치 드라이버는 이러한 방식으로 네임 서버와 통신하기 때문에, 이러한 유형의 구역화는 대부분의 상황에 적합하다. 그러나, 허용된 접속 리스트에 있지 않은 노드를 액세스하려는 장치 드라이버가 설계될 가능성이 있다. 이런 일이 발생하면, 스위치는 이러한 위반을 방지하지도 검출하지도 못한다.

이러한 경우를 방지하기 위해서, 스위치는, 소프트 구역화에 추가하여, "하드 구역화(hard zoning)"로 불리는 메커니즘을 선택적으로 구현한다. 이러한 하드웨어 구역화에서는, 프레임의 전송이 허용된다면 각 프레임의 소스 및 목적지 어드레스에만 기초하여 스위치 네트워크가 결정을 내린다.

광채널 스토리지 영역 네트워크(SAN)는 스토리지 장치를 호스트 서버에 접속하는 네트워크이다. 이들은 네트워킹 인프라 구조로서 광채널 기술에 기반하여 구축된다. SAN이 종래의 상호접속 방식과 구별되는 점은, 호스트 서버와 스토리지 간의 임의-대-임의(any-to-any) 접속에 추가하여 모든(또는 대부분의) 스토리지가 하나의 큰 "스토리지 영역"에 통합되어 중앙형(단순화된) 관리를 할 수 있다는 기본 개념이다.

광채널 SAN은, 개방 시스템과 스토리지(즉, 비-z시리즈)를 z시리즈 시스템 및 스토리지와 같은 네트워크로 상호접속할 수 있도록 허용하는 잠재 능력을 가진다. 이것은, 개방 부착(open attachment) 및 z시리즈 부착(zSeries attachment) 양자 모두에 대한 프로토콜이 광채널 아키텍처의 FC-4 레이어에 매핑되기 때문에 가능하다.

광채널 부착에서, LUN은, 호스트가 부착되는 ESS(IBM 기업 스토리지 서버)가 어떤 것인지에 관계없이, 호스트의 광채널 어댑터(어댑터의 월드 와이드 고유 식별자, a. k. a. 월드 와이드 포트 네임을 통해)에 친화력을 갖는다. 따라서, 하나의 광채널 호스트가 ESS 상의 복수의 광채널 포트에 대한 액세스를 가질 수 있는 스위치 패브릭 구성에서, 광채널 호스트에 의해 액세스될 수 있는 LUN 세트는 ESS 포트들 각각에 동일하다.

이 구현의 결과 중 하나는, SCSI에서와는 달리, 광채널을 사용하면, 동일한 광채널 포트로의 패브릭을 통해 ESS에 부착된 호스트가 동일한 LUN을 "관측"하지 못할 수도 있다. 이것은, 각각의 광채널 호스트마다 LUN 마스킹이 상이할 수 있기 때문이다. 즉, 각 ESS는 어느 호스트가 어느 LUN에 액세스를 갖는지를 정의할 수 있다.

다른 방법은, 각 호스트로부터의 각각의 광채널 포트가 ESS 상의 하나의 광채널 포트에 부착되도록 제한되는 방식으로, 스위치 내에 구역을 생성하는 것이다. 이렇게 함으로써, 호스트가 하나의 경로를 통해서만 LUN을 관측할 수 있도록 한다.

광채널 사양의 세부사항은 이하와 같은 표준들에 나타낸다: 광채널 물리 및 시그널링 인터페이스(FC-PH), ANSI X3.230-1994; 광채널 2세대 물리 인터페이스(FC-PH-2), ANSI X3.297-1997; 광채널 3세대 물리 인터페이스(FC-PH-3), ANSI X3.303-199X, 개정 9.4 및 광채널 중재 루프(FC-AL), ANSI X3.272-1996. 추가로 관련된 표준들에는 FC-FS, FC-GS-3이 있다.

광채널에 관한 추가 정보는, 광채널 컨설턴트 - 포괄적 소개(Robert W. Kembel,1998)와 광채널 컨설턴트 중재 루프(Robert W. Kembel, 1996)에 개시되어 있다.

Barry Stanley Barnett등에 의해 미국 뉴욕주 아몽크(Armonk)의 인터내셔널 비즈니스 머신스 코퍼레이션에 양도된, 발명이 명칭이 "Method and system for end-to-end problem determination and fault isolation for storage area networks"이고 2000년 12월 22일 출원된 EP 1 115 225 A2는 스토리지 영역 네트워크(SAN)에서 문제점 판정 및 고장 분리를 위한 방법 및 시스템을 개시한다. 멀티벤더 호스트 시스템, FC 스위치, 및 스토리지 주변장치의 복잡한 구성은 통신 아키텍처(CA)를 통해 SAN에 접속된다. 통신 아키텍처 요소(CAE)는 네트워크 서비스 프로토콜을 통해 호스트 컴퓨터 상의 통신 아키텍처 관리자(CAM)에 성공적으로 등록된 네트워크 접속 장치이고, CAM은 SAN에 대한 문제점 판정(PD) 기능을 포함하여 SAN PD 정보 테이블을 유지한다(SPDIT). CA는 SPDIT에서 저장된 정보를 통신할 수 있는 모든 네트워크 접속 요소를 포함한다. CAM은 SAN 토폴로지 맵을 사용하고 SPDIT는 SAN 진단 테이블(SDT)를 생성하는데 사용된다. 특정 장치에서 고장 컴포넌트는 동일 네트워크 접속 경로를 따라 장치가 에러를 생성하게 하는 에러를 생성할 수 있다. CAM이 에러 패킷 또는 에러 메시지를 수신함에 따라, 에러는 SDT에 저장되고 각 에러는 SDT 내의 다른 에러들과 시간 및 공간적으로 비교되어 분석된다. CAE가 에러를 생성하는 후보인 것으로 판정되면, CAE는 가능한 경우 대체용으로 보고된다.

일본의 SawaoIwatani, Kawasaki에 의해 2001년 2월 9일 출원되고 2001년 12월 20일에 공개된 발명의 명칭이 "Storage area network management system, method, and computer-readable medium"인 미국특허출원번호 제2001/0054093 Al호는, 하나의 소스로부터 종래 분산된 보안 시스템을 통합하여 관리하고 SAN에서 보안 관리를 자동화하는 스토리지 영역 네트워크(SAN)의 통합 관리 메커니즘을 개시한다. 통합 관리 메커니즘은 SAN을 통합하여 관리하고 SAN의 호스트 컴퓨터와 스토리지 장치의 액세스 관계가 통합 관리 메커니즘을 사용하여 관리되도록 구성된다. 호스트 컴퓨터로부터 액세스가 시도되는 스토리지 장치 영역, 이 스토리지를 액세스할 때 사용되는 광채널 어댑터, 및 호스트 버스 어댑터

(HBA) 등의 통합 관리 메커니즘 상의 액세스 경로가 구성된다. 구성된 액세스 경로 정보에 따라, 통합 관리 메커니즘은 개별 스토리지 설정, 구역화 설정 및 호스트 컴퓨터의 SAN 관리 메커니즘, 스위치의 구역화 설정 메커니즘, 및 스토리지 장치의 스토리지 관리 메커니즘에 대한 액세스가능 영역 허용을 설정한다.

발명의 상세한 설명

이로부터, 본 발명의 목적은, 복수의 서버가 하나의 광채널 어댑터를 공유하여 개선된 보안 메커니즘을 갖는 서버 환경에서 스토리지 영역 네트워크를 운영하는 방법 및 시스템을 제공하는 것이다.

상기 목적은 독립항들에 개시된 방법 및 시스템에 의해 달성된다. 본 발명의 다른 실시예가 종속항에서 설명되며 후술하는 명세서에서 교시된다.

본 발명에 따르면, 복수의 서버가 하나의 광채널 어댑터를 공유하는 서버 환경에서 스토리지 영역 네트워크(SAN)를 운영하는 방법 및 시스템이 제공된다. SAN 관리 서버는 스토리지 시스템 영역과 보안, 및/또는 에러 검출 및 SAN 구성을 관리하고, 광채널 네트워크는 스토리지 장치로의 접속과 상기 서버 환경에서 실행되는 복수의 운영 체제 이미지를 제공한다. 또한, 신뢰 SAN 관리 클라이언트 유닛이 상기 SAN 관리 서버와 광채널 어댑터에 접속되어, 상기 신뢰 SAN 관리 클라이언트 유닛이, 상기 각각의 운영 체제 이미지 대신에, 상기 광 채널 네트워크에 명령을 내리도록 구성된다.

본 발명의 바람직한 실시예에서, 서버 환경은 가상 서버 및/또는 분할된 서버를 포함한다.

바람직하게는, SAN 관리 서버는 제1 명령 세트와 제2 명령 세트를 구별하도록 구성되어, 제1 명령 세트는 상기 SAN과 함께 SM 클라이언트에 의해 처리되고, 상기 제2 명령 세트는 상기 SAN으로의 액세스 없이 상기 OS 이미지에 의해 처리된다. 광채널 어댑터(FC 어댑터)는 상기 신뢰 SAN 관리 클라이언트 유닛을 인증하도록 구성되는 것이 바람직하다.

FC 어댑터와 상기 SAN은, 비신뢰 OS 이미지의 액세스를 최소 필요의 명령 세트로 제한할 수 있다. 대안으로서, FC 어댑터와, 가상 서버의 가상 층은 비신뢰 OS 이미지의 액세스를 최소 필요의 명령 세트로 제한할 수 있다.

본 발명의 다른 실시예에서, 서버 부하를 작게 유지하도록 단 하나의 SM 클라이언트가 제공된다. 선택적으로, 리던던시를 제공하도록 하나 이상의 백업 SM 클라이언트가 제공된다.

유익하게도, SAN으로부터의 메시지를 수신하는기 위해 SM 클라이언트만이 등록되고, 이 SAM 클라이언트는 상기 메시지를 상기 SM 서버에만 포워딩하도록 구성된다. 선택적으로는, FC 어댑터는, SM 클라이언트 뿐만 아니라 비신뢰 OS 이미지에게도 등록할 필요가 없는 모든 메시지를 전송하도록 구성된다.

본 발명의 다른 실시예에서, 서버는 2개의 에이전트 클래스, 즉, SM 클라이언트 및 원격 액세스 서버(RA 서버)가 구비된다. 바람직하게는, 서버가 RA 서버를 액세스하기 위한 인증 데이터(authorization data)를 유지하는 저장소가 구비되어 있다.

바람직하게는, 단지 SM 클라이언트와 FC 어댑터가 각각의 비신뢰 OS 이미지에 의한 자원의 사용을 빌링하는데 사용되는 정보를 수집하도록 구성된다. 유익하게도, SM 프레임워크는 액세스 권한을 설정하기 위해서 방화벽 제어 애플리케이션과 통신할 수 있다.

본 발명의 다른 실시예에서, SM 클라이언트는 SM 서버에서 RA 서버로의 요청에 대한 라우터로서 동작할 수 있다. 바람직하게는, 기존 텔넷/sshd 서버는 RA 서버를 형성한다.

더욱이, 본 발명은 복수의 운영 체제 이미지가 하나의 광채널 어댑터를 공유하고, 광채널 어댑터로의 통신 채널을 갖는 적어도 하나의 SAN 관리 서버 및 적어도 하나의 SAN 관리 클라이언트와 더불어 SAN 관리 소프트웨어에 의해 SAN이 관리되는 서버 환경에서, 스토리지 영역 네트워크(SAN)를 운영하는 방법으로서 구현될 수 있다. SAN 관리 서버가 내린 요청은 적어도 2개의 그룹으로 분리된다. 즉, 제1 그룹은 신뢰 경로에 대응하며 동일 어댑터를 공유하는 다른 운영 체제들 대신에 SM 클라이언트를 위하여 SAN과 광 채널 어댑터에 의해 처리되고, 제2 그룹은, FC 어댑터 및 SAN과의 요청의 송수신 없이, 다른 운영 체제들에 의해 처리된다.

본 방법의 바람직한 실시예에서, SAN과 FC 어댑터에서 발생된 비요청 메시지에 포함된 모든 정보는 SAN 관리 클라이언트에 의해 SAN 관리자에 라우팅된다. 바람직하게는, HBA_API 결합 요청이 방화벽을 수정하는데 사용된다.

선택적으로, SAN 관리 클라이언트로부터 어댑터로의 통신 경로는 다른 운영 체제 이미지에 의해 수정되거나 도청될 수 없도록 동작한다. 다른 바람직할 실시예에서, 개개의 운영 체제 이미지를 빌링하는데 관련된 모든 정보는 어댑터에 생성되고, SAN 클라이언트에 의해 신뢰 경로 상에서 SAN을 통해 SAN 관리자에 라우팅된다.

유익하게도, SM 서버는, 제1 그룹으로부터 요청을 실행하기 위해 인증 데이터를 SM 클라이언트에 제공한다. 선택적으로, SM 서버와 SM 클라이언트는, 제2 그룹으로부터의 요청을 실행하기 위해 인증 데이터를 다른 OS 이미지들에 제공한다. 바람직하게는, OS 이미지들은 SAN에서 제한된 명령 세트만을 실행할 수 있도록 동작한다.

많은 수의 운영 체제들(> 256, 4000 가상 서버가 2004년에 가능)이 공유된 FC 어댑터들을 사용하여 SAN에 참여하여야 한다. 이러한 유형의 시나리오에서 어댑터들은, 비용 및 관리능력 때문에 공유된다. 256개의 FC 어댑터들을 구비한 서버는, 예를 들면 어댑터로부터 스위치로의 256개의 케이블과 스위치 네트워크에서 256개의 포트를 요구한다.

서버 호스팅 환경에서, 각각의 OS 이미지는 전용 데이터(private data)를 요구하며, 이는 다른 OS 이미지(예를 들면, 서버 구성), 공유 판독 기입 데이터, 예를 들면 공유 데이터베이스와 공유 판독 전용 데이터, 예를 들면, 동일 어댑터에 의해 OS 미미지에 의해 공유되는 LUN에 대하여 미리설치된 운영 체제 이미지(유닉스 시스템에서 /usr)에 의해 액세스되지 않을 수 있어서, 완전히 SCSI 호환 동작을 보장할 수 없다(예를 들면, SAM-2에서 정의되는 바와 같은 예약/릴리스, NACA 핸들링, 큐잉 규칙).

각각의 "비신뢰 운영 체제 이미지"는 잠재적으로 위험한 엔티티, 예를 들어 서로 경쟁하는 두 회사 또는 해커에 의해 소유된다. 여기서 소유라는 단어는, 실제로 루트 액세스를 가지며, 모든 SM 클라이언트 및 기타 소프트웨어를 포함한 운영 체제 이미지의 모든 부분을 변경할 수 있는 엔티티를 의미한다. OS 이미지 소유자는 하드웨어에 대한 액세스권을 갖지 않으며 그 자신을 위해 하드웨어를 변형할 수 없고, 이를 행하기 위해서는 머신 소유자가 필요하다. 머신 소유자(IT 부서/ASP/ISP)는, SAN 내의 모든 하드웨어, 매핑 및 정책에 대한 전체 제어권을 갖는다. 머신 소유자는 SAN에서의 자원(아마도, 디스크 제어기에서는 LUN)을 운영 체제 이미지에 할당한다. 머신 소유자는, 운영 체제 이미지의 다운을 방지하기 위해 SAN 내의 에러 검출 및 고장 분리(fault isolation)를 수행할 도구가 필요하다. 머신 소유자는, 자신의 하드웨어를 다른 서브넷들(그룹 다중 서버 또는 LPAR)과 독립하여 관리될 수 있는 서브넷 및 여러 엔티티로 분할하기를 원할 수 있다. 공유 어댑터는 서브넷을 확장(span)하지 않지만, LPAR는 확장할 수 있다.

각각의 OS 이미지는 개별 서버(블레이드 서버) 상에서 또는 가상 서버로서 동작한다. 가상 서버 환경은 예를 들면 인텔 기반 서버 상의 LPAR-z시리즈 펌웨어, z시리즈 VM 운영 체제 또는 VM웨어에 의해 생성될 수 있다.

SAN 관리자 사용자 인터페이스에서 SAN 자원의 프리젠테이션은, SM 서버 사용자 인터페이스에 의해 제공된 프리젠테이션과는 완전히 상이한 종류의 프리젠테이션을 얻지 않고, 머신 사용자가 OS 이미지를 가상 서버로부터 물리 서버로 이동할 수 있도록 하는 방식으로 행해져야 한다.

각 OS 이미지의 액세스는 OS 이미지에 속하지 않은 FC 어댑터 또는 다른 엔티티에서 방화벽에 의해 제한될 수 있다(다르게는, OS 이미지에 대하여 완전한 제어를 갖는 루트 액세스 권한을 구비한 사용자가 이를 우회할 능력을 가짐).

각각의 비신뢰 OS 이미지에서 FC 대 SCSI 매핑은 FC 장치 구동기 매핑과 구성 인터페이스 MAP_IF에 의해 구성될 수 있다. 예를 들면, z시리즈 상의 광채널의 특정 버전에 대한 MAP_IF는 프록 파일 시스템(proc-file-system)이라 하는 리눅스형 구성 방법에 의해 구현되어 상기 매핑을 설정 및 조회한다. SM 서버는 MAP_IF에 의해 보고된 정정 데이터에 의존하지 않아야 한다(액세스 실행이 방화벽에 의해 행해지기 때문에, OS는 협력하거나 어떤 데이터도 보지 못할 수 있다). 빌링 측정은 비신뢰 OS 이미지에서 행해질 수 없다(루트 사용자가 데이터를 자유롭게 변형하기 때문에). 따라서, 어댑터 또는 다른 엔티티는 필요한 측정 데이터를 제공하여야 한다.

WWPN의 개수가 어댑터 당 지원된 OS 이미지 개수보다 적을 수 있더라도 다른 실시예는 여러 WWPN를 제공할 수 있다.

실시예

도 1을 참조하면, 본 발명에 따른 시스템(100)의 제1 실시예를 나타내는 블록도가 도시되어 있다. 시스템(100)은 복수의 컴퓨터 시스템(104, 105 및 106)을 포함하며, 이들 모두는 WWPN 1(World Wide Port Name; 월드 와이드 포트 네임)을 갖는 하나의 공통 광채널 어댑터(112)를 통해 SAN(110)에 액세스할 수 있다. 명확성을 위해서, 단지 3개의 컴퓨터를 나타낸다. 컴퓨터 시스템의 개수는 수백 또는 심지어 수천일 수 있다. 이러한 많은 개수의 컴퓨터 시스템을 갖는 경우, 시스템 내에는 하나 이상의 광채널 어댑터가 공유될 수 있으며, 이와 같은 경우에도 복수의 컴퓨터 시스템이 하나의 동일한 광채널 어댑터를 공유할 수 있다.

각 컴퓨터 시스템(104, 105 및 106)과 광채널 어댑터 간의 게이트웨이로서 방화벽(114, 115 및 116)이 각각 제공되며, 이 방화벽들은 각 컴퓨터 시스템이 SAN(110)의 허용된 부분에만 액세스할 수 있도록 보장하기 위한 것이다. 방화벽(114, 115 및 116)은 광채널 어댑터에 통합되거나 또는 이에 부착될 수 있다. 반면에, 모든 컴퓨터 시스템(104, 105 및 106)은 이더넷 네트워크 등의 네트워크(120)에 접속된다.

또한, 시스템(100)은 네트워크(120)에 접속된 2개 이상의 컴퓨터 시스템(122 및 123)을 포함하여, 상기 컴퓨터 시스템들은 운영 체제(OS; 131) 상에서 동작하는 SAN 관리 서버(130; SM 서버), 및 SAN 관리 클라이언트(132; SM 클라이언트)를 각각 호스팅한다. SM 서버(130)는 예를 들면 티볼리(Tivoli) 스토리지 네트워크 관리자(http://www. tivoli. com/products/index/storagenetmgr/)의 서버 부분을 실행하지만, SM 클라이언트(132)는 이러한 시스템의 개별 클라이언트 부분을 실행한다.

또한, SM 서버(130)는 통신 라인을 통해 방화벽 제어 애플리케이션(134)에 접속되는 반면, SM 클라이언트(132)는 광채널 어댑터(112)로의 통신 링크를 갖는다. 방화벽 제어 애플리케이션(134)에는, (상술한) 광채널 어댑터를 통해 각 방화벽(114, 115 및 116)으로의 통신 링크가 제공되거나, 이들 방화벽으로의 직접적인 통신 링크가 제공된다.

SM 서버(130)는, 각각의 보안 쉘 인터페이스(144, 145 및 146)를 통해, 즉, 원격 컴퓨터에 로그인하고 그 컴퓨터 상에서 명령을 실행하기 위한 프로그램을 통해, 컴퓨터 시스템들(104, 105 및 106) 각각에 액세스한다. 각 컴퓨터 시스템(104, 105 및 106)에 제공된 매핑 인터페이스(MAP_IF; 154, 155, 156)는, 광채널-대-SCSI(소형 컴퓨터 시스템 인터페이스) 매핑을 관리한다. 이것은 예를 들어 명령 라인 인터페이스 또는 구성 파일로서 구현될 수 있다.

각 컴퓨터 시스템(104, 105 및 106)은 "비신뢰"일 수 있는 운영 체제(164, 165, 및 166)를 실행한다. 여기서, "비신뢰(untrusted)"라는 것은, 운영 체제가 잠재적으로 위험한 엔티티에 의해 제어되거나 조작될 수 있다는 것을 의미한다. 이러한 잠재적으로 위험한 엔티티로서, SAN 내의 정보를 액세스하거나 변경하기 위해 운영 체제를 조작하는 컴퓨터 바이러스와 같은 악성 코드나 사람이 있을 수 있다. 운영 체제 자체는 인터내셔널 비즈니스 머신 코퍼레이션에 의한 AIX 또는 z/OS, 유닉스 및 리눅스와 같은 광범위한 운영 체제 중 임의의 것에 의해 형성될 수 있다.

본 발명에 따른 보안 메커니즘은 비인증 SAN 액세스 요청이 컴퓨터 시스템(104, 105 및 106) 중 임의의 하나에 의해 발생될 수 있음을 고려하고 있기 때문에, 각각의 방화벽(114, 115 및 116)은 모든 SAN 액세스 요청을 필터링하여 방화벽 제어 애플리케이션에 의해 인증된 것만을 승인한다. 컴퓨터 시스템(104, 105 및 106) 중 임의의 것에 의해 액세스가능하지 않은 방화벽 제어 애플리케이션(134)만이 비인증 SAN 액세스 요청을 규정하는 방화벽 설정을 변경할 수 있게 된다. 본 발명의 제1 실시예에 따르면, SM 서버(130)는 방화벽 제어 애플리케이션(134)을 제어한다.

SM 클라이언트(132)는 "신뢰"이어야 하는 운영체계(170) 상의 HBA_API(168; 호스트 버스 어댑터 애플리케이션 프로그램 인터페이스) 상에서 실행하며, 즉, 운영 체제는 SAN 액세스 권한 등을 변형하려 하지 않는 엔티티에 의해 관리된다. 광채널 어댑터(112)는 비신뢰 및 신뢰 운영 체제를 구별할 수 있다. 이러한 인증은, 본 발명의 일부가 아닌, 고정 소스 어드레스(예를 들면, 하드웨어 ID), 키 및 암호 알고리즘, 또는 패스워드와 같은 공지된 인증 방식 중 일부 수단에 의해 운영 체제(170) 또는 SM 클라이언트(132)를 식별하여 달성될 수 있다. SM 클라이언트(132)를 실행하는 신뢰 운영 체제(170)로부터의 SAN 구성, SAN 컴포넌트, SAN 액세스 권한, 에러 메시지, 통계 또는 빌링 정보 등의 중요 정보 관련 요청만이 허용될 수 있다. 이에 따라, 이러한 요청의 결과는 신뢰 운영 체제(170)에만 리턴된다. 즉, SM 서버에 의해 제어되고 비신뢰 운영 체제를 대신하여 동작하는 SM 클라이언트(132)는 SAN 관리를 수행한다.

SM 서버(130)는 SAN과 관리 운영 체제에서 정보를 조회 및 설정하기 위해서 요청을 발생하는 SM 코어 엔진(미도시)과 코어 엔진에서 클라이언트로의 요청 및 응답을 라우팅하는 통신 모듈(미도시)을 포함한다. 이 통신 모듈은 "신뢰" 환경에서 상주하는 SM 서버의 일부로서 구현되거나 SM 서버 및 SM 클라이언트 컴포넌트 상에 배치될 수 있다.

보안 쉘 대몬(sshd)과 OS 특정 FC 구성 인터페이스는 RA(원격 액세스) 서버로서 요약된다. 원격 액세스 서버는 SM 서버 또는 SM 클라이언트에 의해 전송된 인증 요청에 응답할 수 있다. 바람직한 구현예에서, SM 클라이언트와 RA 서버는 패스워드, 사용자 ID, 및 암호 키와 같은 인증 데이터를 사용하여 요청의 발신자를 식별한다. 따라서, 통신 모듈이 상기 인증 정보에 대한 저장소(미도시)에 구비된다. 통신 모듈은 패브릭 관련 요청, 즉 SAN을 관리하는 요청, 및 어댑터 관련 요청, 즉, 광채널 어댑터를 관리하는 요청을 후술하는 명령 리스트에 따른 OS 특정 요청과 분리한다. 상술한 바와 같이, SM 서버, SM 클라이언트 및 RA 서버 간의 접속은 예를 들면 이더넷 상에서 동작하는 IP 기반 네트워크이다. 다른 실시예에서, SM 클라이언트, SM 서버 및 RA 서버 간의 몇몇 통신은 개별 네트워크 대신에 다른 프로토콜을 전송하는 FC 어댑터 성능을 사용한다. 일 예로서 광채널 상의 TCP/IP이다.

도 2를 참조하면, 본 발명의 제2 실시예에 따른 시스템을 나타내는 블록도가 도시되어 있다.

제1 실시예의 시스템(도 1)에 대응하여, 본 시스템은 여러 컴퓨터 시스템(204, 205', 206)을 포함하며, 이들은 모두 WWPN 1(월드 와이드 포트 네임)을 갖는 하나의 공통 광채널 어댑터(212)를 통해 SAN(210)에 액세스할 수 있다. 명확성을 위해서, 단지 3개의 컴퓨터 시스템이 도시되어 있다. 컴퓨터 시스템의 개수는 수백 또는 수천일 수 있다. 이러한 많은 개수의 컴퓨터 시스템을 가짐으로써, 복수의 컴퓨터 시스템이 하나의 동일한 광채널 어댑터를 공유한 경우에도, 단지 하나의 광채널 어댑터가 이 시스템에서 존재할 수 있다.

컴퓨터 시스템(204, 205 및 206) 각각과 광채널 어댑터 간의 게이트웨이로서, 각 컴퓨터 시스템이 SAN(210)의 허용 부분을 액세스할 수 있도록 보장하는 방화벽(214, 215 및 216)이 각각 제공된다. 방화벽(214, 215, 및 216)은 광채널 어댑터에 통합되거나 이에 부착될 수 있다.

제1 실시예와 달리, LPAR(논리 분할된 모드) 플러스 VM(가상 머신)과 같은 가상 서버 환경에서 동작하는 가상 서버는 컴퓨터 시스템(204, 205 및 206)을 형성한다. 또한, 다른 가상 서버(223)가 SAN 관리 클라이언트(232; SM 클라이언트)를 호스팅하도록 제공된다. 컴퓨터 시스템(204, 205, 및 206)은 하이퍼소켓 상부의 보안 쉘 접속을 통해 SM 클라이언트(232)와 통신한다.

개별 컴퓨터 시스템(222)은 SAN 관리 서버(230; SM 서버)를 호스팅하도록 제공된다. SM 서버(230)는 예를 들면, 티볼리 스토리지 네트워크 관리자인 SAN 관리 시스템의 서버 부분을 실행하지만, SM 클라이언트(232)는 이러한 시스템의 개별 클라이언트 부분을 실행한다. SM 서버(230)는 이더넷 네트워크(220)를 통해 SM 클라이언트를 액세스한다.

각 컴퓨터 시스템(204, 205 및 206)은 "비신뢰"일 수 있는 운영 체제(264, 265 및 266)를 실행한다(상기 참조). 운영 체제 차체는 인터네셔널 비즈니스 머신스 코퍼레이션에 의한 AIX 또는 z/OS, 유닉스 또는 리눅스와 같은 광범위한 운영 체제 중 임의의 것에 의해 형성될 수 있다.

SM 클라이언트(232)는 "신뢰"이어야 하는(상기 참조) 운영체계(270) 상의 HBA_API(268; 호스트 버스 어댑터 애플리케이션 프로그램 인터페이스) 상에서 실행한다. 또한, 광채널 어댑터(212)는 비신뢰 또는 신뢰 운영 체제를 구별할 수 있다. SM 클라이언트(232)를 실행하는 신뢰 운영 체제(270)로부터의 중요 정보에 관련된 요청이 구별될 수 있다. 또한, 방화벽 제어 애플리케이션(234)은 "신뢰" 운영 체제(270)의 상부에서 동작한다. 방화벽 제어 애플리케이션은 "비신뢰" 운영 체제(264, 265 및 266)로부터의 특정 액세스 요청을 SAN(210)에 전송할 지를 방화벽(214, 215 및 216)에 지시하는데 사용된다. 즉, 컴퓨터 시스템(2045, 205, 및 206) 중 임의의 것에 의해 액세스가능하지 않은 방화벽 제어 애플리케이션(234) 만이 인증된 SAN 액세스 요청을 규정하는 방화벽 설정을 변경할 수 있게 된다.

도 3을 이하 참조하면, 여러 서버가 본 발명에 따른 광채널 어댑터를 공유하는(블록 300), 서버 환경에서 스토리지 영역 네트워크를 운영하는 방법을 나타낸 흐름도가 도시되어 있다. 우선, SM 코어 엔진은 요청을 생성한 후(블록 302), 통신 모듈이 요청의 타겟을 결정한다(블록 306). SAN 또는 광채널 어댑터에 대한 요청의 경우, 통신 모듈은 패스워드, 사용자 id 및 암호 키일 수 있는 개별 인증 데이터를 사용하여 SM 클라이언트로의 통신 경로를 설정한다(블록 308).

다음으로, SM 클라이언트는 인증이 유효한지를 점검한다(블록 312). 그러한 경우, SM 클라이언트는 스위치(예를 들면, SAN에서 FC 장치 리스트를 검색) 또는 디스크 제어기(예를 들면, 제어기에서 논리 디스크 리스트를 검색)와 같이 SAN 내의 엔티티를 조회하고 특정 유형의 에러 통지 메시지를 등록하는데 사용되는 RNID-ELS와 같은 SAN 속성과 광채널 어댑터를 설정함으로써, 응답을 생성한다(블록 314).

그렇지 않은 경우, SM 클라이언트는 SM 코어에 상기 요청이 거절되었음을 알리는 거부 응답을 생성한다(블록 316). 두 응답은 모두 통신 시스템에 응답을 전송하도록 진행한다(블록 318).

블록 306으로 돌아가서, HBAGetFcpTargetMappingFunc와 같은 펑션에 의해 한정된 광채널 대 SCSI 매핑 등의 운영 체제(OS) 구성 데이터에 대한 요청을 SM 코어 엔진이 생성했다고 통신 모듈이 판정한 경우, 통신 모듈은 각각의 인증 데이터를 사용하여 RA 서버에 통신 경로를 설정한다(블록 320).

그 후, RA 서버, 예를 들면, sshd에서 인증 컴포넌트는 제시된 사용자 id, 패스워드 및 키를 인증된 RA 서버에 의해 알려진 사용자 id, 패스워드 및 키와 비교함으로써, 인증이 유효한지를 판정한다(블록 321).

그러한 경우, RA 서버는 운영 체제의 광채널 장치 구동기에 의해 저장된 광채널 구성 정보를 액세스하는 등의 OS 구성 동작에 의해 응답을 생성한다(블록 324).

그렇지 않은 경우, RA 서버는 거부 응답을 생성한다(블록 326). 또한, 두 응답은 모두 모두 통신 시스템에 응답을 전송하도록 진행한다(블록 318). 다음으로, 통신 시스템은 SM 코어 엔진에 응답을 전송하고(블록 320) SM 서버 내의 SM 코어 엔진은 상술한 두 특허에서 한정된 응답을 처리한다(블록 322).

다음으로, 바람직한 구현예에서 요청의 흐름과 유형을 나타낸다. HBA_API 명령의 구문은 ""Fibre Channel HBA_API Working draft"(ftp ://ftp. tl. prg/tll/pub/fc/hba/02-2 68v2. pdf)에서 찾을 수 있다.

1. 우선, 패브릭 및 어댑터 관련 요청 CT, ELS, SCSI 명령이 열거된다. 어댑터와 SAN 자원을 사용하는 SM 클라이언트는 이들 명령을 처리한다:

```
typedef HBA_STATUS(* HBASendCTPassThruFunc) (HBA_HANDLE, void
*, HBA_UINT32, void *, HBA_UINT32);
typedef HBA_STATUS (* HBASendRNIDFunc)(HBA_HANDLE, HBA_WWN,
HBA_WWNTYPE, void *,HBA_UINT32 *);
typedef HBA_STATUS (* HBASendScsiInquiryFunc)(HBA_HANDLE,
HBA_WWN,HBA_UINT64, HBA_UINT8,HBA_UINT32, void *, HBA_UINT32,
void *, HBA_UINT32);
typedef HBA_STATUS (* HBASendReportLUNsFunc)(HBA_HANDLE,
```

```
HBA_WWN, void *, HBA_UINT32, void *,HBA_UINT32);
typedef HBA_STATUS (* HBASendReadCapacityFunc)(HBA_HANDLE,
HBA_WWN, HBA_UINT64, void *, HBA_UINT32,void *, HBA_UINT32);
typedef HBA_STATUS (* HBASendCTPassThruV2Func)(HBA_HANDLE,
HBA_WWN, void *, HBA_UINT32, void *,HBA_UINT32 *);
typedef HBA_STATUS (* HBASendRNIDV2Func) (HBA_HANDLE, HBA_WWN,
HBA_WWN, HBA_UINT32, HBA_UINT32, void *,HBA_UINT32*);
typedef HBA_STATUS (* HBAScsiInquiryV2Func)
(HBA_HANDLE,HBA_WWN,HBA_WWN, HBA_UINT64, HBA_UINT8, HBA_UINT8,
void *, HBA_UINT32 *, HBA_UINT8 *,void *, HBA_UINT32 *);
typedef HBA_STATUS (* HBAScsiReportLUNsV2Func)(HBA_HANDLE,
HBA_WWN, HBA_WWN, void *, HBA_UINT32 *,HBA_UINT8 *, void *,
HBA_UINT32 *);
typedef HBA_STATUS (* HBAScsiReadCapacityV2Func)(HBA_HANDLE,
HBA_WWN, HBA_WWN, HBA_UINT64, void *,HBA_UINT32 *, HBA_UINT8
*, void *, HBA_UINT32 *);
typedef HBA_STATUS (* HBASendRPLFunc)(HBA_HANDLE, HBA_WWN,
HBA_WWN, HBA_UINT32,HBA_UINT32, void *, HBA_UINT32 *);
typedef HBA_STATUS (* HBASendRPSFunc)(HBA_HANDLE, HBA_WWN,
HBA_WWN, HBA_UINT32, HBA_WWN,HBA_UINT32, void *, HBA_UINT32
*);
typedef HBA_STATUS (* HBASendSRLFunc)(HBA_HANDLE, HBA_WWN,
HBA_WWN, HBA_UINT32, void *,HBA_UINT32 *);
typedef HBA_STATUS (* HBASendLIRRFunc)(HBA_HANDLE, HBA_WWN,
HBA_WWN, HBA_UINT8, HBA_UINT8,void *, HBA_UINT32 *);

typedef HBA_HANDLE (* HBAOpenAdapterFunc)(char *);
typedef void (* HBACloseAdapterFunc)(HBA_HANDLE);
typedef HBA_STATUS (* HBAGetAdapterAttributesFunc)(HBA_HANDLE,
HBA_ADAPTERATTRIBUTES *);
typedef HBA_STATUS (* HBAGetAdapterPortAttributesFunc)
(HBA_HANDLE, HBA_UINT32, HBA_PORTATTRIBUTES *);
typedef HBA_STATUS (* HBAGetPortStatisticsFunc) (HBA_HANDLE,
```

```
HBA_UINT32, HBA_PORTSTATISTICS *);
typedef HBA_STATUS (* HBAGetDiscoveredPortAttributesFunc)
(HBA_HANDLE, HBA_UINT32, HBA_UINT32,HBA_PORTATTRIBUTES *);
typedef HBA_STATUS (* HBAGetPortAttributesByWWNFunc)
(HBA_HANDLE, HBA_WWN, HBA_PORTATTRIBUTES *);
typedef void (* HBARefreshInformationFunc)(HBA_HANDLE);
typedef void (* HBAResetStatisticsFunc)(HBA_HANDLE,
HBA_UINT32);
typedef HBA_STATUS (* HBAGetEventBufferFunc) (HBA_HANDLE,
HBA_EVENTINFO *, HBA_UINT32 *);
typedef HBA_STATUS (* HBASetRNIDMgmtInfoFunc) (HBA_HANDLE,
HBA_MGMTINFO *);
typedef HBA_STATUS (* HBAGetRNIDMgmtInfoFunc) (HBA_HANDLE,
HBA_MGMTINFO *);
typedef HBA_STATUS (* HBAOpenAdapterByWWNFunc) (HBA_HANDLE *,
HBA_WWN);
typedef void (* HBARefreshAdapterConfigurationFunc) ();
typedef HBA_UINT32 (*
HBAGetVendorLibraryAttributesFunc)(HBA_LIBRARYATTRIBUTES *);
typedef HBA_STATUS (* HBAGetFC4StatisticsFunc)(HBA_HANDLE,
HBA_WWN, HBA_UINT8,HBA_FC4STATISTICS *);
typedef HBA_STATUS (* HBAGetFCPStatisticsFunc)(HBA_HANDLE,
const HBA_SCSIID *,HBA_FC4STATIS
typedef HBA_UINT32 (* HBAGetNumberOfAdaptersFunc)();
typedef HBA_STATUS (* HBAGetAdapterNameFunc)(HBA_UINT32, char
*);
```

도 4는 상술한 유형의 HBA_API 명령에 대한 간이 신호 및 흐름을 나타내는 흐름도이다. 도 4에서 명확한 바와 같이, SM 서버는 SM 클라이언트와 통신하며, SM 클라이언트는 FC 스위치와 통신한다. SM 서버, SM 클라이언트 및 FC 스위치가 개시되어 블록 402, 404 및 406에 의해 개시한 바와 같이 서로 독립하여 실행된다. 다음 단계는 동작의 일부만을 구성할 수 있으며, 보다 많은 요청이 후술하는 방법 전후에 전송 또는 수신된다.

초기에는, SM 서버는 SM 클라이언트에 요청을 전송한다(블록 408 및 410). 이 예에서, 이는 HBASendLIRRFunc이다. 구체적으로, SM은 광채널 표준 FC-FS와 FC-GS3에 의해 한정된 바와 같은 CTIU, ELS 또는 FCPCMD 시퀀스를 패브릭에 전송하기 위해서 HBA_API를 사용한다(412, 414).

FC 표준에 의해 한정된 바와 같이 패브릭 내에 생성된 응답은 HBA_API 콜의 완성 부분에 의해 SM 클라이언트에 전송된다(418, 420). SM 클라이언트는 상기 응답을 SM 서버에 전송한다(422, 424).

2. FCP < - > SCSI 매핑 명령

이들 명령은 OS와 OS 장치 구동기 자원 명령을 사용하는 RA 서버를 통해 처리된다. 이 구문은 "Fibre Channel HBA_API Working draft" (ftp://ftp. tll.org/tll/pub/fc/hba/02-268v2. pdf)"에서 찾을 수 있다.

```
typedef HBA_STATUS (* HBAGetFcpTargetMappingFunc)(HBA_HANDLE,
HBA_FCPTARGETMAPPING *);
typedef HBA_STATUS (*
HBAGetFcpPersistentBindingFunc)(HBA_HANDLE, HBA_FCPBINDING *);
typedef HBA_STATUS (* HBAGetBindingCapabilityFunc)(HBA_HANDLE,
HBA_WWN, HBA_BIND_CAPABILITY *);
typedef HBA_STATUS (* HBAGetBindingSupportFunc) (HBA_HANDLE,
HBA_WWN, HBA_BIND_CAPABILITY *);
typedef HBA_STATUS (* HBASetBindingSupportFunc) (HBA_HANDLE,
HBA_WWN, HBA_BIND_CAPABILITY);
```

```
typedef HBA_STATUS (* HBASetPersistentBindingV2Func)
(HBA_HANDLE, HBA_WWN, const HBA_FCPBINDING2 *);
typedef HBA_STATUS (* HBAGetPersistentBindingV2Func)
(HBA_HANDLE, HBA_WWN, HBA_FCPBINDING2 *);
typedef HBA_STATUS (* HBARemovePersistentBindingFunc)
(HBA_HANDLE, HBA_WWN, const HBA_FCPBINDING2 *);
typedef HBA_STATUS (*
HBARemoveAllPersistentBindingsFunc)(HBA_HANDLE, HBA_WWN);
typedef HBA_STATUS (*
HBAGetFcpTargetMappingV2Func)(HBA_HANDLE, HBA_WWN,
HBA_FCPTARGETMAPPING *);
```

도 5를 이하 참조하면, 상술한 유형의 HBA_API 명령에 대한 간이 신호 및 흐름을 나타내는 흐름도가 도시되어 있다. 도 5에서 명확한 바와 같이, SM 서버는 SM 클라이언트와 통신한다. SM 클라이언트는 리턴으로 방화벽과 OS 이미지를 각각 통신한다. SM 서버, SM 클라이언트, 방화벽 및 OS 이미지는 개시되어 블록 502, 504, 506 및 508에 의해 나타낸 바와 같이 서로 독립적으로 동작한다. 다음 단계는 동작의 일부만을 구성하며, 보다 많은 요청이 후술하는 방법 전후에 송수신된다.

우선, SM 서버는 예를 들면 HBASetPersistentBindingV2 HBA_API 요청에 대응하는 요청을 SM 클라이언트에 전송한다(블록 510, 512). SM 클라이언트가 방화벽 상에서 직접 제어를 갖는 경우, 방화벽 보안 정책에 의해 요구되면 이는 선택적으로 방화벽을 변형할 수 있다(514, 516, 518, 520). 그 후, SM 클라이언트는 전용 갱신 방화벽 메시지의 전송 동작을 트리거하여 방화벽을 변형한다(514, 516).

방화벽은 SM 클라이언트에 동작의 완료를 알린다(518, 520). 그 후, SM 클라이언트는 RA 서버(522, 524)에 의해 비신뢰 OS 이미지에서 설정 또는 조회 요청을 트리거한다(526). SM 클라이언트는 상기 요청의 완성 메시지를 대기한다(528, 530). SM 클라이언트는 SM 서버에 상기 요청의 응답을 리턴한다(532, 534).

3. 인커밍 ELSses(RNID)

이들은 패브릭 내에서 개시하여 SM 서버에 전송될 필요가 있는 메시지이다. 이들 메시지는 SAN에서 발생하는 문제점의 소스를 식별하는데 사용된다.

인커밍 ELSses를 처리하는 HBA_API에 한정된 명령:

```
typedef HBA_STATUS (* HBARegisterForAdapterAddEventsFunc)(void
(*)(void *, HBA_WWN, HBA_UINT32),void *,HBA_CALLBACKHANDLE *);
typedef HBA_STATUS (* HBARegisterForAdapterEventsFunc)(void
(*)(void *, HBA_WWN, HBA_UINT32),void *,HBA_HANDLE,
HBA_CALLBACKHANDLE *);
typedef HBA_STATUS (*
HBARegisterForAdapterPortEventsFunc)(void (*)(void *, HBA_WWN,
HBA_UINT32, HBA_UINT32),void *, HBA_HANDLE, HBA_WWN,
HBA_CALLBACKHANDLE *);
typedef HBA_STATUS (* HBARegisterForLinkEventsFunc)(void
(*)(void *, HBA_WWN, HBA_UINT32, void *,HBA_UINT32),void *,
void *, HBA_UINT32, HBA_HANDLE,HBA_CALLBACKHANDLE *);
typedef HBA_STATUS (*
HBARegisterForAdapterPortStatEventsFunc)(void (*)(void *,
HBA_WWN, HBA_UINT32), void *,HBA_HANDLE, HBA_WWN,
HBA_PORTSTATISTICS,HBA_UINT32, HBA_CALLBACKHANDLE *);
typedef HBA_STATUS (* HBARegisterForTargetEventsFunc)(void
```

```
(*)(void *, HBA_WWN, HBA_WWN, HBA_UINT32),void *, HBA_HANDLE,
HBA_WWN, HBA_WWN,HBA_CALLBACKHANDLE *, HBA_UINT32 );
typedef HBA_STATUS (*
HBARemoveCallbackFunc)(HBA_CALLBACKHANDLE);
```

도 6을 이하 참조하면, 상술한 유형의 HBA_API 명령에 대한 간이 신호 및 흐름을 나타내는 흐름도가 도시되어 있다. 도 6에 명확한 바와 같이, SM 서버는 SM 클라이언트와 통신하며, SM 클라이언트는 FC 스위치와 통신한다. SM 서버, SM 클라이언트 및 FM 스위치는 개시되어 블록 602, 604 및 606에 나타낸 바와 같이 서로 독립적으로 동작한다. 다음 단계는 동작의 일부만을 구성하며 보다 많은 요청이 후술하는 방법 전후에 송수신된다.

SM 서버(602)는 SM 클라이언트(604)가 HBA_API에 의한 이벤트에 한번 등록하도록 지시하고(608, 610), 비신뢰 OS 이미지가 등록할 수 없음의 완성 확인을 대기한다(612, 614). 이 것이 행해진 후, SAN에 의해 생성된 각 메시지(616)는 다음 절차를 트리거한다:

1. SM 클라이언트는 FC 어댑터로부터 이벤트를 수신한다(616, 618)

2. SM 클라이언트는 이벤트를 SM 서버에 전송한다(620, 622).

다른 구현예에서, SM 클라이언트는 SM 서버에 전송된 메시지의 개수를 감소시키는 메시지를 필터링하여 요약할 수 있다(616, 618).

본 발명에 관련되지 않은 HPA API 펑션:

```
HBA_API functions which are not relevant to this invention:
typedef HBA_UINT32 (* HBAGetVersionFunc)();
typedef HBA_STATUS (* HBALoadLibraryFunc)();
typedef HBA_STATUS (* HBAFreeLibraryFunc)();
```

본 발명은 하드웨어, 소프트웨어 또는 하드웨어와 소프트웨어의 조합으로 구현될 수 있다. 본 명세서에서 기술된 방법을 실행할 수 있도록 적응된 임의 종류의 컴퓨터 시스템 또는 다른 장치도 적합하다. 전형적인 하드웨어 및 소프트웨어의 조합으로서, 로딩되어 실행될 때 본 명세서에서 기술된 방법을 실행하도록 컴퓨터 시스템을 제어하는 컴퓨터 프로그램을 갖는 범용 컴퓨터 시스템이 될 수 있다. 또한, 본 발명은 본 명세서에서 설명하는 방법의 구현예를 가능하게 하는 모든 특성을 포함하는 컴퓨터 프로그램 제품으로 임베디드되어, 컴퓨터 시스템에 로딩될 때 이들 방법을 실행할 수 있다.

본 발명에서 컴퓨터 프로그램 수단 또는 컴퓨터 프로그램은, 정보 처리 능력을 갖는 시스템이 특정 펑션을 직접 수행하거나, a) 다른 언어, 코드 또는 표지로의 변환; b) 상이한 자료 형태로의 재생 중 어느 하나 또는 양자 모두를 수행한 이후에 수행하도록 의도된 임의의 언어, 코드, 또는 표기로 된 한 세트의 명령어로 나타날 수 있다.

도면의 간단한 설명

본 발명의 상기 및 다른 목적, 특징 및 이점은 후술하는 상세한 설명으로부터 명백해질 것이다.

본 발명의 신규한 특징이 첨부한 청구항에서 설명된다. 그러나, 본 발명 자체뿐만 아니라 이의 바람직한 실시 양태, 다른 목적 및 이점은 첨부 도면과 함께 후술하는 실시예의 상세한 설명으로부터 이해될 수 있다.

도 1은 본 발명의 제1 실시예에 따른 실시예를 나타내는 블록도.

도 2는 본 발명의 제2 실시예에 따른 실시예를 나타내는 블록도.

도 3은 본 발명에 따라 여러 서버가 하나의 광채널 어댑터를 공유하는 서버 환경에서 스토리지 영역 네트워크를 운영하는 방법을 나타내는 흐름도.

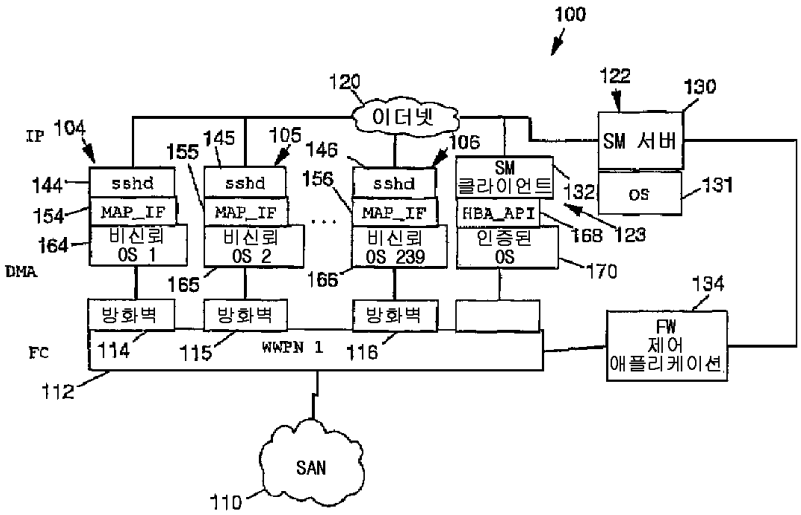
도 4는 상술한 유형의 패브릭 관련 HBA_API 명령에 대한 간이 신호 및 흐름을 나타내는 흐름도.

도 5는 상술한 유형의 FCP 대 SCSI 매핑 HBA_API 명령에 대한 간이 신호 및 흐름을 나타내는 흐름도.

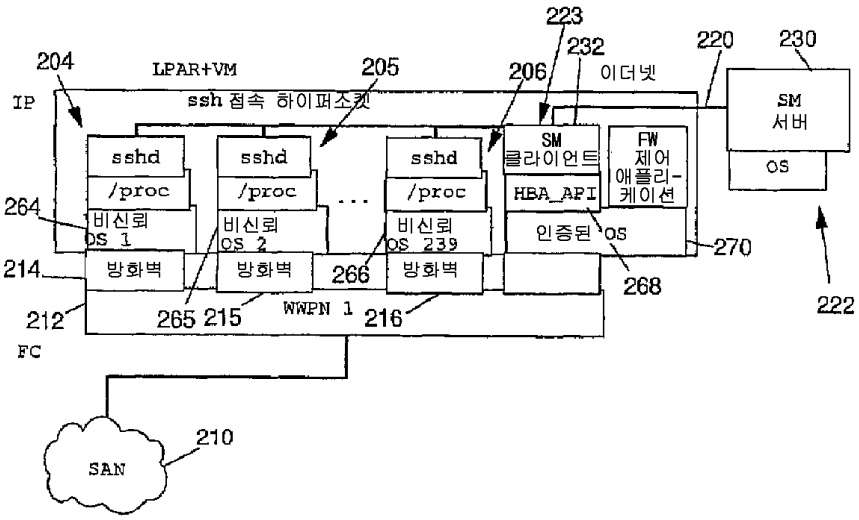
도 6은 SAN에 의해 개시된 ELS 요청을 처리하는 상술한 유형의 HBA_API 명령에 대한 간이 신호 및 흐름을 나타내는 흐름도.

도면

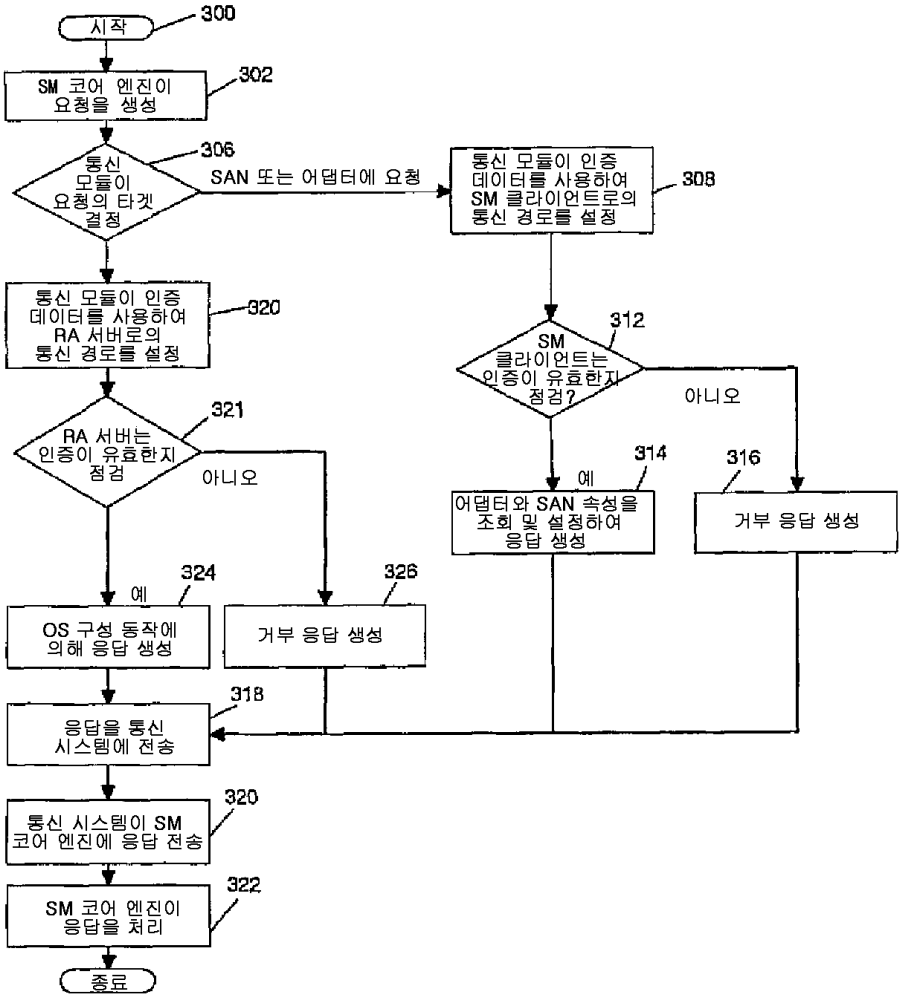
도면1



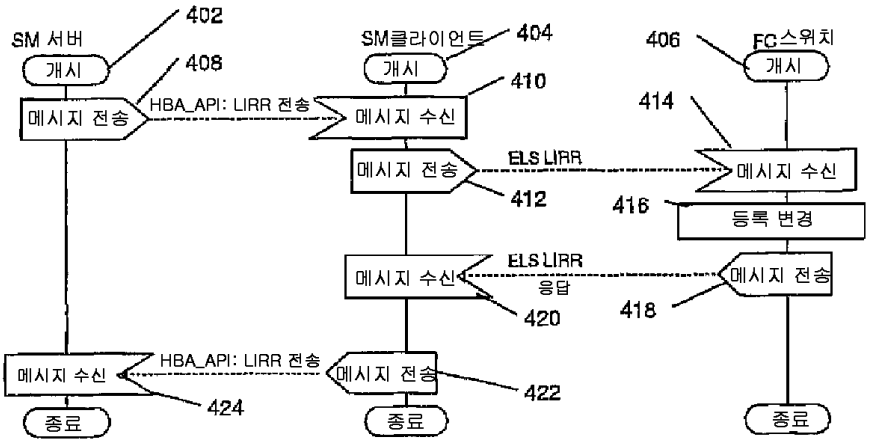
도면2



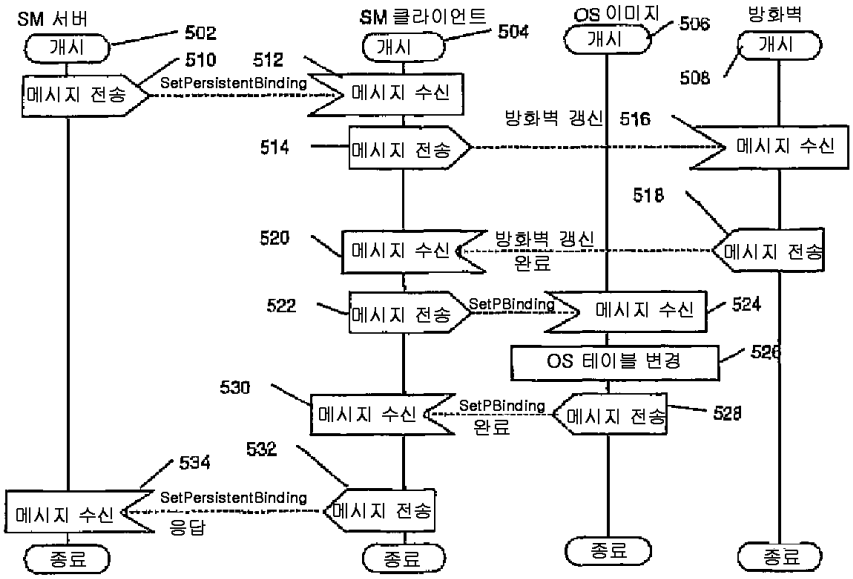
도면3


시작
300
SM 코어 엔진이 요청을 생성
302
통신 모듈이 요청의 타겟 결정
306
SAN 또는 어댑터에 요청
통신 모듈이 인증 데이터를 사용하여 SM 클라이언트로의 통신 경로를 설정
308
통신 모듈이 인증 데이터를 사용하여 RA 서버로의 통신 경로를 설정
320
SM 클라이언트는 인증이 유효한지 점검?
312
아니오
RA 서버는 인증이 유효한지 점검
321
아니오
예
314
316
어댑터와 SAN 속성을 조회 및 설정하여 응답 생성
거부 응답 생성
예
324
326
OS 구성 동작에 의해 응답 생성
거부 응답 생성
응답을 통신 시스템에 전송
318
통신 시스템이 SM 코어 엔진에 응답 전송
320
SM 코어 엔진이 응답을 처리
322
종료

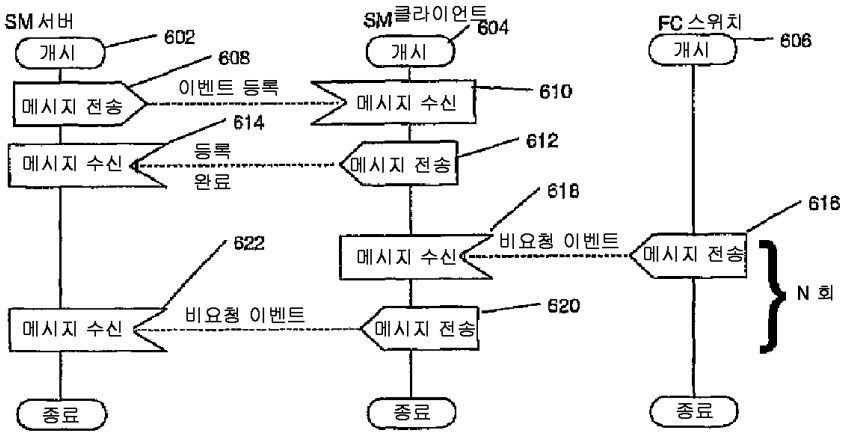
도면4


SM 서버
402
개시
408
메시지 전송
HBA_API: LIRR 전송
메시지 수신
424
종료
SM클라이언트
404
개시
410
메시지 수신
메시지 전송
412
ELS LIRR
메시지 전송
422
종료
406
FC스위치
개시
414
메시지 수신
416
등록 변경
메시지 수신
ELS LIRR
응답
420
메시지 전송
418
종료

도면5


SM 서버
SM 클라이언트
OS 이미지
방화벽
개시
502
504
506
508
510
512
메시지 전송
SetPersistentBinding
메시지 수신
방화벽 갱신
514
516
518
520
방화벽 갱신 완료
522
SetPBinding
524
OS 테이블 변경
526
530
528
532
534
SetPersistentBinding 응답
종료

도면6


SM 서버
SM클라이언트
FC 스위치
개시
602
604
606
608
이벤트 등록
610
614
등록 완료
612
618
비요청 이벤트
616
622
N 회
620
종료