



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2011-0067111
(43) 공개일자 2011년06월21일

- | | |
|--|--|
| <p>(51) Int. Cl.
<i>G06F 13/12</i> (2006.01)</p> <p>(21) 출원번호 10-2011-7007450</p> <p>(22) 출원일자(국제출원일자) 2008년11월20일
심사청구일자 없음</p> <p>(85) 번역문제출일자 2011년03월31일</p> <p>(86) 국제출원번호 PCT/US2008/012964</p> <p>(87) 국제공개번호 WO 2010/036231
국제공개일자 2010년04월01일</p> <p>(30) 우선권주장
61/100,034 2008년09월25일 미국(US)</p> | <p>(71) 출원인
엘에스아이 코퍼레이션
미국 캘리포니아 밀피타스 바어버 레인 1621 (우:95035)</p> <p>(72) 발명자
지베 마흐무드
미국, 67230 캔자스, 위치타, 엔. 브리지필드 319
다르 크리스하누
인도, 560075 방갈로르, 지.엠. 팔야, 스리니바사 엔클레이브, 샵103</p> <p>(74) 대리인
특허법인세신</p> |
|--|--|

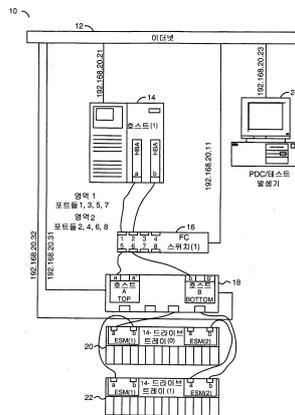
전체 청구항 수 : 총 20 항

(54) 외부 스토리지 어레이에서의 대역의 관리 애플리케이션을 인증하기 위한 방법 및/또는 장치

(57) 요약

스토리지 어레이를 구성하기 위한 방법은, (A) 초기 테스트를 위해 상기 스토리지 어레이를 최소개의 컴포넌트들로 구성하는 단계; (B) 상기 스토리지 어레이의 구조의 테스트를 개시하기 위해 상기 스토리지 어레이에 제1 심볼 호출 명령을 전달하는 단계; (C) 상기 스토리지 어레이로부터 응답을 수신하는 단계; 및 (D) 상기 응답에 응하여 상기 테스트를 통과하였는지를 판정하는 단계를 포함한다.

대표도 - 도1



특허청구의 범위

청구항 1

스토리지 어레이를 구성하기 위한 방법으로서,

- (A) 초기 테스트를 위해 상기 스토리지 어레이를 최소개의 컴포넌트들로 구성하는 단계;
- (B) 상기 스토리지 어레이의 구조의 테스트를 개시하기 위해 상기 스토리지 어레이에 제1 심볼 호출 명령을 전달하는 단계;
- (C) 상기 스토리지 어레이로부터 응답을 수신하는 단계; 및
- (D) 상기 응답에 응하여 상기 테스트를 통과하였는지를 판정하는 단계를 포함하는 것을 특징으로 하는 스토리지 어레이 구성 방법.

청구항 2

제1항에 있어서,

상기 방법은 상기 테스트를 통과하면 상기 구조를 캡처하는 것을 특징으로 하는 스토리지 어레이 구성 방법.

청구항 3

제2항에 있어서,

상기 방법은 상기 테스트를 통과하지 못하면 상기 구조를 분석하는 것을 특징으로 하는 스토리지 어레이 구성 방법.

청구항 4

제3항에 있어서,

상기 방법은 상기 스토리지 어레이의 다음 테스트를 위해 단계들 (B) 내지 (D)를 반복하는 것을 특징으로 하는 스토리지 어레이 구성 방법.

청구항 5

제1항에 있어서,

상기 방법은 스누퍼 디바이스에서 단계들 (C) 및 (D)를 구현하는 것을 특징으로 하는 스토리지 어레이 구성 방법.

청구항 6

제1항에 있어서,

상기 방법은 상기 테스트를 통과하면 상기 구조를 파일에 쓰는 것을 특징으로 하는 스토리지 어레이 구성 방법.

청구항 7

제6항에 있어서,

상기 파일은 XML 파일을 포함하는 것을 특징으로 하는 스토리지 어레이 구성 방법.

청구항 8

제6항에 있어서,

상기 파일은 HTML 파일을 포함하는 것을 특징으로 하는 스토리지 어레이 구성 방법.

청구항 9

제6항에 있어서,

상기 파일은 TXT 파일을 포함하는 것을 특징으로 하는 스토리지 어레이 구성 방법.

청구항 10

제1항에 있어서,

상기 테스트는 다양한 테스트들 중 하나 이상을 포함하는 것을 특징으로 하는 스토리지 어레이 구성 방법.

청구항 11

제1항에 있어서,

상기 스토리지 어레이는 상기 제1 심볼 호출 명령에 응하여 객체 그래프를 채우는 것을 특징으로 하는 스토리지 어레이 구성 방법.

청구항 12

제1 운영체제와 관련된 제1 세트의 심볼 호출들을 발생시키도록 구성된 제1 회로;

(i) 상기 제1 세트의 심볼 호출들을 수신하고 (ii) 상기 제1 세트의 심볼 호출들에 대하여 한 세트의 응답들을 발생시키도록 구성된 제2 회로; 및

(i) 상기 한 세트의 응답들을 캡처하고 (ii) 상기 한 세트의 응답들을 저장하도록 구성된 제3 회로를 포함하는 것을 특징으로 하는 장치.

청구항 13

제12항에 있어서,

(i) 상기 제1 회로로부터 제2 운영체제와 관련된 제2 세트의 심볼 호출들을 수신하고 (ii) 상기 제2 세트의 심볼 호출들에 응하여 상기 제1 회로에 상기 한 세트의 응답들을 전달하도록 구성된 제4 회로를 더 포함하는 것을 특징으로 하는 장치.

청구항 14

제13항에 있어서,

상기 제4 회로는 (i) 네트워크 인터페이스, (ii) 스누퍼 기능부, 및 (iii) 상기 한 세트의 응답들을 저장하기 위한 메모리를 포함하는 것을 특징으로 하는 장치.

청구항 15

제14항에 있어서,

상기 제4 회로는 상기 한 세트의 응답들을 XML 파일에 저장하는 것을 특징으로 하는 장치.

청구항 16

제15항에 있어서,

상기 XML 파일은 객체 그래프의 구조를 포함하는 것을 특징으로 하는 장치.

청구항 17

제12항에 있어서,

상기 제1 회로는 샌트리시티/심플리시티 호스트로 구현되는 것을 특징으로 하는 장치.

청구항 18

제12항에 있어서,

상기 제1 회로는 복수의 운영체제들 중 어느 하나 이상을 실행시킬 수 있는 것을 특징으로 하는 장치.

청구항 19

제12항에 있어서, 상기 제2 회로는 스토리지 어레이로 구현되는 것을 특징으로 하는 장치.

청구항 20

스토리지 어레이를 인증하기 위한 방법으로서,

- (A) 상기 스토리지 어레이에 제1 운영체제와 관련된 제1 세트의 심볼 호출들을 전달하는 단계;
- (B) 상기 스토리지 어레이로부터의 한 세트의 응답들을 스누퍼 디바이스로 캡처하는 단계;
- (C) 상기 응답들을 디바이스에 저장하는 단계;
- (D) 상기 스토리지 어레이에 다른 운영체제와 관련된 다른 세트의 심볼 호출들을 전달하는 단계;
- (E) 상기 다른 세트의 심볼 호출들을 상기 디바이스로 인터셉트하는 단계; 및
- (F) 상기 다른 세트의 심볼 호출들에 응하여 상기 디바이스로부터의 상기 저장된 응답들을 전달하는 단계를 포함하는 것을 특징으로 하는 스토리지 어레이 인증 방법.

명세서

기술분야

[0001] 본 발명은 스토리지 어레이들에 관한 것으로, 특히 외부 스토리지 어레이에서의 대역외(out of band) 관리 애플리케이션을 인증하기 위한 방법 및/또는 장치에 관한 것이다.

배경기술

[0002] 종래의 테스트 환경들은 다양한 문제점들에 직면하고 있다. 클라이언트 인터페이스는 정확한 명령을 전달할 수 있지만 어레이는 비응답 상태에 있을지 모른다. 따라서, 어레이는 전혀 응답하지 않거나 또는 부정확한 정보로 응답한다. 이러한 경우에, 엔지니어는 종종 구성을 검사하기 위한 시간을 보낸다. 문제는 쉬운 해결책을 가질지도 모르지만, 이러한 문제를 제거하기 위해 아마 상당한 공수(man-hours)가 소모될 것이다.

[0003] 종래의 해결방식들은 이 문제에 대한 직접적인 해결책을 제공하지 못한다. 종래의 어레이가 구성 문제에 직면하면, 종래의 해결책은 문제를 제거하여 바로잡는 것이다. 종래의 해결방식들은 많은 단점들을 가지고 있다. 이러한 종래의 해결방식들에 의해, 엔지니어는 검사를 위한 시간을 보낼 준비를 하여야 한다. 스토리지 어레이를 최적의 상태에 이르게 하기 위해 많은 작업 시간이 소모될 수 있다.

발명의 내용

해결하려는 과제

[0004] 외부 스토리지 어레이에서의 OOB 관리 애플리케이션을 인증하기 위한 방법 및/또는 장치를 구현하는 것이 바람직하다.

과제의 해결 수단

[0005] 본 발명은 (A) 초기 테스트를 위해 스토리지 어레이를 최소개의 컴포넌트들로 구성하는 단계; (B) 상기 스토리지 어레이의 구조의 테스트를 개시하기 위해 상기 스토리지 어레이에 제1 심볼 호출 명령을 전달하는 단계; (C) 상기 스토리지 어레이로부터 응답을 수신하는 단계; 및 (D) 상기 응답에 응하여 상기 테스트를 통과하였는지를 판정하는 단계를 포함하는 스토리지 어레이 구성 방법에 관한 것이다.

[0006] 본 발명의 목적들, 특징들 및 장점들은 (i) 어레이로부터의 심볼 호출 (및 해당 응답들)을 (예를 들면, 객체 그래프 구조 형태로) 캡처하고, (ii) 심볼 호출 및 링크를 적절한 심볼 응답으로 해석하기 위한 메커니즘을 제공하고, (iii) 다른 환경들(예를 들면, 운영체제들)에서 OOB 샌드리시터를 인증하기 위한 추가 하드웨어의 필요성을 없애고 및/또는 (iv) 블록 스토리지 어레이 네트워크 제품들(예를 들면, SAN) 또는 네트워크 어레이 스토리지(NAS)에 사용될 수 있는 디바이스(예를 들면, 스누퍼 디바이스)의 제공을 포함한다.

발명의 효과

[0007] 본 발명이 바람직한 실시예들을 참조하여 특별히 도시되고 설명되었지만, 본 발명의 범위를 일탈하지 않고 형태와 상세에 있어 다양한 변형물들이 가능하다는 것을 이 기술분야의 당업자는 이해할 것이다.

도면의 간단한 설명

[0008] 본 발명의 목적들, 특징들 및 장점들은 다음의 상세한 설명, 첨부한 청구항들, 및 도면들로부터 명백할 것이다.

도 1은 어레이 구성의 블록도이다.

도 2는 데이터 캡처 단계에서의 본 발명의 일실시예의 블록도이다.

도 3은 데이터 캡처 단계에서의 본 발명의 일실시예의 프로세스의 흐름도이다.

도 4는 데이터 검색 단계에서의 본 발명의 일실시예의 블록도이다.

도 5는 데이터 검색 단계에서의 본 발명의 일실시예의 프로세스의 흐름도이다.

도 6은 본 발명의 일실시예에 따른 상세 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0009] 도 1은 전형적인 어레이 구성을 구현하는 시스템(10)의 블록도이다. 시스템(10)은 블록(12), 블록(또는 회로)(14), 블록(또는 회로)(16), 블록(또는 회로)(18), 블록(또는 회로)(20), 블록(또는 회로)(22), 및 블록(또는 회로)(24)를 포함한다. 블록(12)은 이더넷 네트워크를 나타낸다. 블록(14)은 호스트 디바이스를 나타낸다. 블록(16)은 섬유 채널(FC) 스위치를 나타낸다. 블록(18)은 스토리지 어레이(또는 컨트롤러)를 나타낸다. 블록(20)은 드라이브 트레이를 나타낸다. 블록(22)은 드라이브 트레이를 나타낸다. 블록(24)은 테스트 발생기를 나타낸다. 시스템(10)은 이더넷 네트워크(12)를 통해 OOB 관리를 하는 어레이 구성을 나타낸다.

[0010] 본 발명의 일실시예는 구성의 반복 테스트동안 스토리지 어레이(18) 및 드라이브 트레이들(20 및 22)을 제거할 수 있다. 스토리지 어레이(18) 및 드라이브 트레이들(20 및 22)은 (i)이더넷 네트워크(12)를 통해 프레임들을 송수신하고, (ii)스누퍼(snooper) 기능을 제공하고, 및/또는 (iii)서로다른 심볼 호출들을 위한 객체 그래프(Object Graph)를 XML 파일에 저장하기 위한 능력을 제공할 수 있는 능력들을 갖는 것으로 알려진 디바이스로 교체될 수 있다.

[0011] 심볼 호출이 OOB 관리를 통해 테스트될 필요가 있으면, 호스트(14)는 함수 호출을 스토리지 어레이(18)에의 TCP/IP 패킷으로 캡슐화한다. 스토리지 어레이(18)는 호출을 수신하여 실행시킬 수 있다. 스토리지 어레이(18)는 적절한 리턴 코드를 클라이언트에 전송하고 객체 그래프내의 구조를 차지할 수 있다. 호스트(14)가 객체 그래프내의 데이터를 해독하면, 호스트(14)는 적절한 정보를 GUI(graphical user interface)로 디스플레이한다.

[0012] 도 2는 본 발명의 바람직한 실시예에 따른 시스템(100)의 블록도이다. 시스템(100)은 일반적으로 블록(또는 회로)(102), 블록(또는 회로)(104), 및 블록(106)을 포함한다. 회로(102)는 호스트 회로로 구현될 수 있다. 일예에 있어서, 회로(102)는 샌트리티티/심플리시티(SANtricity/Simplicity) 회로(또는 모듈)로 구현될 수 있다. 블록(104)은 스토리지 어레이일 수 있다. 예를 들면, 블록(104)은 디스크 드라이브들 또는 다른 스토리지 디바이스들(예를 들면, 고체 상태 스토리지 등)의 어레이를 나타낼 수 있다. 블록(106)은 네트워크(예를 들면, 이더넷 네트워크)를 나타낼 수 있다. 네트워크(106)는 일반적으로 복수의 블록들(108a-108n), 블록(또는 회로)(110), 블록(또는 회로)(112), 및 블록(또는 회로)(114)를 포함한다. 블록들(108a-108n)은 각각 스토리지 어레이(104)에의 심볼 호출을 나타낸다. 블록(110)은 스토리지 어레이(104)로부터의 응답을 나타낼 수 있다. 블록(112)은 스누퍼 회로로 구현될 수 있다. 블록(114)은 어레이(104)로부터 수신된 또 하나의 응답을 나타낼 수 있다. 추가 응답들(110 및/또는 114)이 구현될 수 있다.

[0013] 시스템(100)의 다양한 실시예들은 많은 이점들을 가지고 있다. 예를 들면, 테스트동안 스토리지 어레이(104)의 필요성이 없어질 수 있다. 스토리지 어레이(104)는 디바이스를 위한 데이터베이스를 구축하는 동안 한번 사용될 수 있다. 이러한 구현은 테스트가 완료된 후 스토리지 어레이(104)를 동작시킬 때 구성 문제에 부딪칠 위험성을 낮출 수 있다. 또 하나의 예에 있어서, 호스트(102)로부터의 심볼 호출들(108a-108n) 중 하나에 응하여 스토리지 어레이(104)에 의해 전송된 응답들(110 또는 114)은 일반적으로 운영체제(OS)에 종속되지 않는다. 심볼 호출들(108a-108n)은 어떠한 타입의 운영체제도 실행시키는 호스트(102)(또는 네트워크(106)에 연결된 또 하나의 호

스트)에 의해 초기화될 수 있다. 또 하나의 예에 있어서, 호스트(102)는 스토리지 어레이(104)에서의 동작이 완료되기를 기다릴 필요가 없기 때문에 심볼 호출들(108a-108n)을 테스트하는데 걸리는 시간은 시스템(100)이 없는 환경에서의 테스트 시간보다 일반적으로 짧을 것이다. 시스템(100)은 관리 애플리케이션들을 인증하는데 필요한 리소스들을 저감시킬 수 있다.

[0014] 스토리지 어레이(104)를 테스트하기 위한 흐름은 두 단계((i)데이터 캡처 단계 및 (ii)데이터 검색 단계)로 분류될 수 있다. 데이터 캡처 단계는 호스트(102), 스누퍼 디바이스(112), 및 어레이(104)의 최소한의 구성을 이용할 수 있다. 호스트(102)는 일반적으로 하드웨어 디바이스 및/또는 소프트웨어 애플리케이션으로서 설치된 센트리시티/심플리시티를 가질 수 있다.

[0015] 도 3은 프로세스(200)를 나타낸 도면이다. 프로세스(200)는 데이터 캡처 단계에 있는 시스템(100)을 나타낼 수 있다. 프로세스(200)는 일반적으로 단계(또는 상태)(202), 단계(또는 상태)(204), 단계(또는 상태)(206), 판정 단계(또는 상태)(208), 단계(또는 상태)(210), 및 단계(또는 상태)(212)를 포함한다. 단계(202)는 프로세스(200)를 시작하게 할 수 있다. 단계(204)는 호스트(102)(예를 들면, 센트리시티/심플리시티)에게 초기 테스트(예를 들면, 테스트 i)의 실행을 명령할 수 있다. 단계(206)는 스누퍼(112)에게 스토리지 어레이(104)로부터의 응답(110) 기록을 명령한다. 판정 단계(208)는 테스트를 통과하였는지를 판정할 수 있다. 테스트를 통과하면, 프로세스(200)는 단계(210)로 계속될 수 있다. 단계(210)는 XML 파일내의 구조를 캡처할 수 있다. XML 파일이 언급되었지만, 다른 타입의 파일들(예를 들면, HTML, TXT 등)이 구현될 수 있다. 테스트를 통과하지 못하면, 프로세스(200)는 단계(212)로 이동할 수 있다. 단계(212)는 실패를 분석하고 테스트를 재실행할 수 있다. 프로세스(200)가 단계(212)로 이동한 후, 프로세스(200)는 단계(204)로 돌아갈 수 있다.

[0016] 설명한 테스트는 다양한 테스트들 중 하나 이상을 포함할 수 있다. 일예에 있어서, 연속적인 두개의 테스트로서 한 세트(suite)의 테스트가 구현될 수 있다. 예를 들면, 한 세트내의 첫번째 테스트는 create volume 동작의 함수를 테스트하는 것일 수 있다(여기서, i=0(예를 들면, 제로로 초기화된 변수 'i'를 얻음)). 호스트(102)는 관련 파라미터들을 갖는 특정 심볼 호출(예를 들면, CREATEVOLUME)을 이더넷 네트워크(106)를 통해 스토리지 어레이(104)로 전송할 수 있다. 스토리지 어레이(104)는 심볼 호출 CREATEVOLUME을 수신하고, 요청된 함수를 실행하고, 적절한 리턴 코드(예를 들면, 응답(110))를 다시 전송하고, 객체 그래프를 채울 수 있다. 다음 표 1은 객체 그래프의 일예를 나타낸다.

표 1

```
VOLUME - 0xc0e2b84
volumeHandle : 0x2
raidLevel : 0x0
dssPrealloc : 0x1
absMaxSegSize : 0x200000
offline : 0x0
sectorOffset : 0xa00000
blk/segSize : 0x200/0x20000
capacity : 0x140000000
reconPriority : 0x1
preReadRedun : 0x0
media scan : 0x0/0x0
status/action : 0x1 OPTIMAL / 0x1
cache : CMA CME RCA RCE WCA WCE
cache modifier: 0x8
readAheadMult : 0x1
WWN : 60 0a 0b 80 00 2f c0 4b 00 00 a3 9e 47 cb 98 54
volumeGroupRef: 04 00 00 00 60 0a 0b 80 00 2f c0 4b 00 00 a3 9c 47 cb 98 1d
volumeRef : 02 00 00 00 60 0a 0b 80 00 2f c0 4b 00 00 a3 9e 47 cb 98 54
currentMgr : 07000000000000000000000000000001
preferredMgr : 07000000000000000000000000000001
label : 00 76 00 6f 00 6c 00 75 00 6d 00 65 00 33
label : volume3
permissions : MAP=Y, SNAP=Y, FORMAT=Y
RECONFIG=Y, READ=Y, WRITE=Y
MIRROR PRIMARY=Y, MIRROR SECONDARY=Y
COPY SOURCE=Y, COPY TARGET=Y
```

[0017]

[0018] 그리고, 스토리지 어레이(104)로부터 전송된 패킷을 받아오면 스누퍼 디바이스(112)는 리턴 코드(110)를 캡처할

수 있다. 리턴 코드(110)는 호스트(102)로 전달될 수 있다. 호스트(102)가 리턴 코드(110)(예를 들면, RETCODE_OK)를 수신하면, 호스트(102)는 파일(예를 들면, jRPC)을 통해 스토리지 어레이(104)로부터 객체 그래프를 수신할 수 있다. 다음 표 2는 특정 심볼 호출을 위해 어레이(104)가 호스트(102)로 전송할 수 있는 복수의 리턴 코드들(예를 들면, 응답들(110 및 114))을 나타낸다.

표 2

RETCODE_CANNOT_FORMAT_VOLUME
RETCODE_CONTROLLER_IN_SERVICE_MODE
RETCODE_DRIVE_NOT_EXIST
RETCODE_DRIVE_NOT_UNASSIGNED
RETCODE_ERROR
RETCODE_ILLEGAL_PARAM
RETCODE_INVALID_LABEL
RETCODE_MAX_VOLUMES_EXCEEDED
RETCODE_NO_HEAP
RETCODE_OK
RETCODE_RAID6_FEATURE_DISABLED
RETCODE_RAID6_FEATURE_UNSUPPORTED
RETCODE_TRY_ALTERNATE
RETCODE_VOLUME_GROUP_RECONSTRUCTING
RETCODE_VOLUME_GROUP_UNDERGOING_COPYBACK

[0019]

[0020]

객체 그래프가 호스트(102)로 전송되는 동안, 스누퍼 디바이스(112)는 패킷내의 데이터를 캡처하고, 데이터를 국부적으로(예를 들면, XML 파일에) 저장하고, 패킷을 호스트(102)로 전달할 수 있다. 그리고, 호스트(102)는 한 세트내의 다음 테스트를 실행할 준비를 할 수 있다(예를 들면, $i = i + 1$, 여기서 1 만큼 증분된 'i'를 얻음). 테스트에 실패하면, 실패의 원인을 확정하기 위해 일반적으로 분석이 행해진다. 그리고, 이러한 실패한 테스트는 재실행될 수 있다(예를 들면, $i=0$, 여기서 값 'i'는 변함이 없음).

[0021]

데이터 캡처 단계는 일반적으로 특별한 반복 테스트 사이클에서 한번 일어난다. 데이터 캡처 단계에서, 하나의 운영체제(예를 들면, 윈도우 등)와 관련된 심볼 호출들(108a-108n)이 적용될 수 있다. 객체 구조들은 캡처되어 국부적으로 저장될 수 있다. 데이터 검색 단계에서, 데이터 캡처 단계에서 테스트된 심볼 호출들(108a-108n)은 도 4와 관련하여 설명된 기술을 이용하여 다른 운영체제들(예를 들면, AIX, Linux, Solaris, HP-UX 등)에 대하여 테스트될 수 있다.

[0022]

도 4는 데이터 검색 단계에서의 시스템(300)의 블록도이다. 시스템(300)은 일반적으로 블록(또는 회로)(302), 블록(또는 회로)(304), 블록(306), 복수의 심볼 호출들(308a-308n), 및 복수의 응답들(310a-310n)을 포함한다. 회로(302)는 호스트 회로로 구현될 수 있다. 일예에 있어서, 호스트 회로(302)는 샌트리시티/심플리시티 회로로 구현될 수 있다. 호스트 회로(302)는 복수의 표준 운영체제들 중 어느 하나 이상을 실행시킬 수 있다. 블록(304)은 디바이스 회로로 구현될 수 있다. 일예에 있어서, 디바이스(304)는 스토리지 어레이(104)의 IP 어드레스, 스누퍼 능력들 및/또는 데이터 저장 능력(예를 들면, 메모리)을 갖는 하드웨어로 구현될 수 있다. 일예에 있어서, 데이터는 XML 포맷으로 저장될 것이다. 블록(306)은 네트워크(예를 들면, 이더넷 네트워크)로 구현될 수 있다.

[0023]

일예에 있어서, 데이터 검색 단계를 위한 최소한의 구성은 (샌트리시티/심플리시티가 설치된) 호스트(302) 및 (스누퍼 유사 능력과 네트워크 인터페이스를 가지며 구조를 XML 파일에 저장가능한) 디바이스(304)일 수 있다. 테스트 세트가 데이터 캡처 단계에서 완료된 후, 다른 운영체제들로부터의 심볼 호출들(308a-308n)을 테스트하는데 하기 기술이 사용될 수 있다. 이 세트는 데이터 캡처 단계에서와 동일한 일련의 심볼 호출들(308a-308n)로 세트에서의 호출들의 테스트를 시작할 수 있다.

[0024]

호스트(302)는 유사한 파라미터들을 갖는 특정 심볼 호출(예를 들면, CREATEVOLUME)을 디바이스(304)에 전송할 수 있다(예를 들면, $i=0$, 여기서 'i'는 제로로 초기화됨). 디바이스(304)는 스누퍼 기능을 이용하여 요청을 획득하고, 심볼 호출들(308a-308n)에 대한 리턴 코드들(예를 들면, 응답들(310a-310n))을 스토리지에서 검색하고, 리턴 코드들(310a-310n)을 호스트(302)로 전송할 수 있다. 응답들(310a-310n)을 수신하면, 호스트(302)는 객체 그래프에 대한 요청을 디바이스(304)로 전송할 수 있다. 그리고, 객체 그래프의 구조를 저장하는 XML 파일은 호

스트(302)로 전송될 수 있다. 그리고, 호스트(302)는 객체 그래프를 해독하여 테스트에 통과하였는지를 판정할 수 있다. 테스트를 통과하면, 호스트(302)는 세트내의 다음 테스트로 진행할 수 있다(예를 들면, $i=i+1$, 여기서 1 만큼 증분된 'i'를 얻음). 실패한 경우에, 문제점이 없는지 설정(set up)이 점검되어야 하고 테스트는 재실행될 것이다(예를 들면, $i=0$, 여기서 'i'의 값은 변하지 않음).

[0025] 도 5는 프로세스(400)를 나타낸 도면이다. 프로세스(400)는 데이터 검색 단계에서의 시스템(300)의 일예를 나타낸다. 프로세스(400)는 초기화 단계(또는 상태)(402), 단계(또는 상태)(404), 단계(또는 상태)(406), 판정 단계(또는 상태)(408), 및 단계(또는 상태)(410)를 포함할 수 있다. 단계(402)는 프로세스(400)를 시작할 수 있다. 단계(404)는 호스트(302)(샌트리시티/심플리시티)에 테스트(예를 들면, i)의 실행을 명령할 수 있다. 단계(406)는 디바이스(304)에 응답들(310a-310n)의 검색을 명령할 수 있다. 판정 단계(408)는 테스트를 통과하였는지를 판정할 수 있다. 테스트를 통과하면, 프로세스(400)는 단계(404)로 돌아간다. 테스트를 통과하지 못하면, 프로세스(400)는 단계(410)로 진행한다. 단계(410)는 실패에 대하여 설정을 분석한다. 프로세스(400)가 단계(410)로 진행한 후, 프로세스(400)는 단계(404)로 돌아간다.

[0026] 도 6은 프로세스(500)의 흐름도이다. 프로세스(500)는 데이터 검색 단계에서의 디바이스(304)를 나타낸다. 프로세스(500)는 초기화 단계(또는 상태)(502), 단계(또는 상태)(504), 단계(또는 상태)(506), 단계(또는 상태)(508), 판정 단계(또는 상태)(510), 단계(또는 상태)(512), 단계(또는 상태)(514), 단계(또는 상태)(516), 및 단계(또는 상태)(518)를 포함한다. 단계(502)는 프로세스(500)를 시작할 수 있다. 단계(504)에서, 디바이스(304)는 호스트(302)로부터 심볼 호출을 수신할 수 있다. 단계(506)에서, 디바이스(304)는 이더넷 패킷으로부터 심볼 요청을 캡처할 수 있다. 단계(508)에서, 디바이스(304)는 적절한 구조의 이용가능성에 대하여 록업 테이블(예를 들면, 표 1 또는 표 2)을 참조할 수 있다. 판정 단계(510)에서, 디바이스(304)는 구조가 이용가능하지를 판정할 수 있다. 이용가능하지 않으면, 프로세스(500)는 단계(512)로 이동한다. 단계(512)에서, 디바이스(304)는 적절한 에러 메시지(예를 들면, RESPONSE_NOT_AVAILABLE)와 함께 테스트를 실패할 수 있다. 이용가능하면, 프로세스(500)는 단계(514)로 이동한다. 단계(514)에서, 디바이스(304)는 리포지토리(repository)에서 구조를 검색할 수 있다. 단계(516)에서, 디바이스(304)는 이더넷 네트워크(306)를 통해 호스트(302)로 응답(310a-310n)을 전송할 수 있다. 단계(518)는 프로세스(500)를 종료할 수 있다.

[0027] 도 3, 5, 및 6의 흐름도에 의해 수행된 기능은 본 발명에 따라 프로그램된 종래의 범용 디지털 컴퓨터를 이용하여 구현될 수 있다는 것은 이 기술분야의 당업자에게 자명할 것이다. 본 발명을 기반으로 하여 숙련된 프로그래머들에 의해 적절한 소프트웨어 코딩이 용이하게 준비될 수 있다는 것도 이 기술분야의 당업자에게 자명할 것이다.

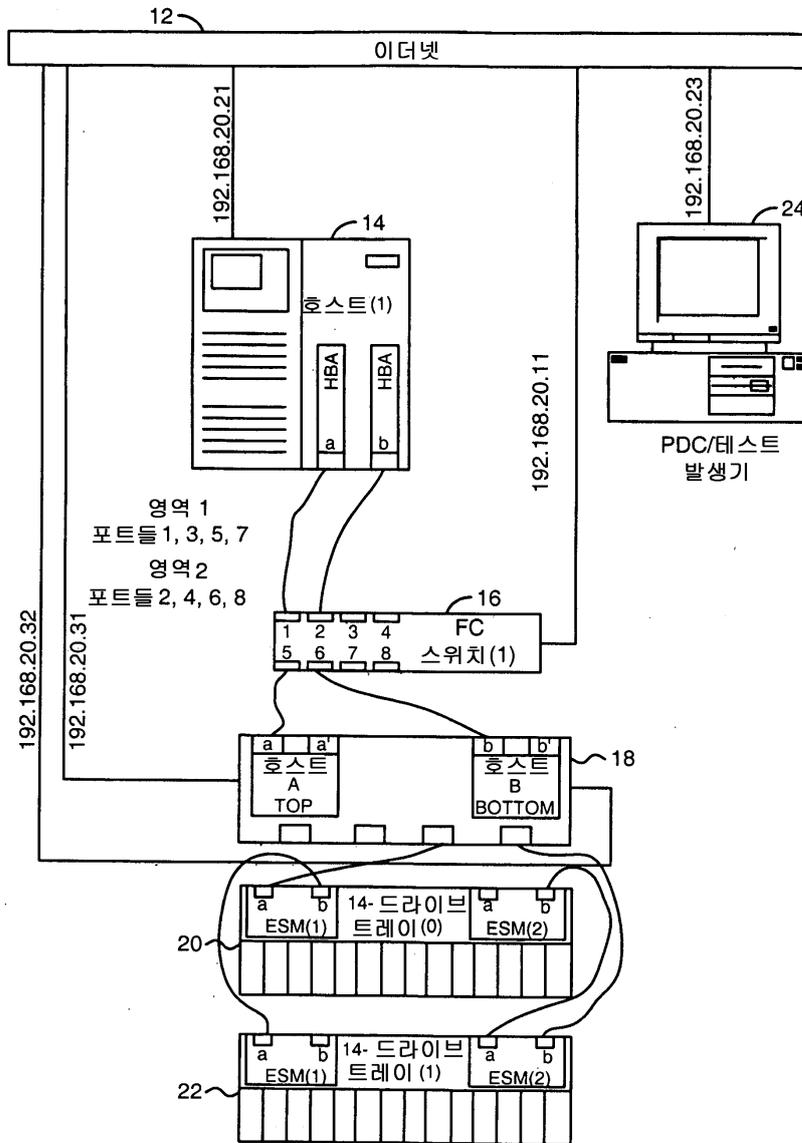
[0028] 또한, 여기에 설명된 것처럼, 본 발명은 ASIC들, FPGA들의 준비에 의해 또는 적절한 종래의 컴포넌트 회로 네트워크의 상호연결에 의해 구현될 수 있으며, 그 변형물들은 이 기술분야의 당업자에게 자명할 것이다.

[0029] 따라서, 본 발명은 컴퓨터가 본 발명에 따라 프로세스를 수행하도록 프로그램하는데 사용될 수 있는 명령들을 포함하는 스토리지 매체일 수 있는 컴퓨터 제품을 포함할 수 있다. 스토리지 매체는, 그에 한정되지는 않지만, 플로피 디스크를 포함하는 모든 타입의 디스크, 광학 디스크, CD-ROM, 자기-광 디스크들, ROM들, RAM들, EPROM들, EEPROM들, 플래시 메모리, 자기 또는 광학 카드들, 또는 전자 명령들을 저장하기에 적합한 모든 타입의 매체들을 포함할 수 있다.

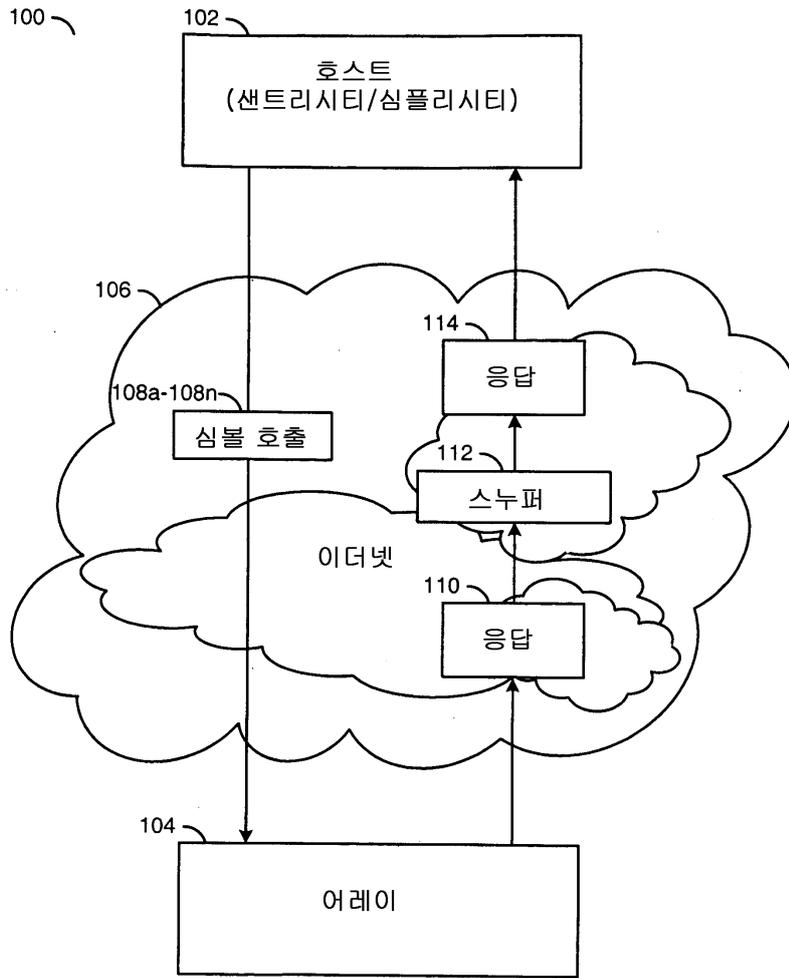
도면

도면1

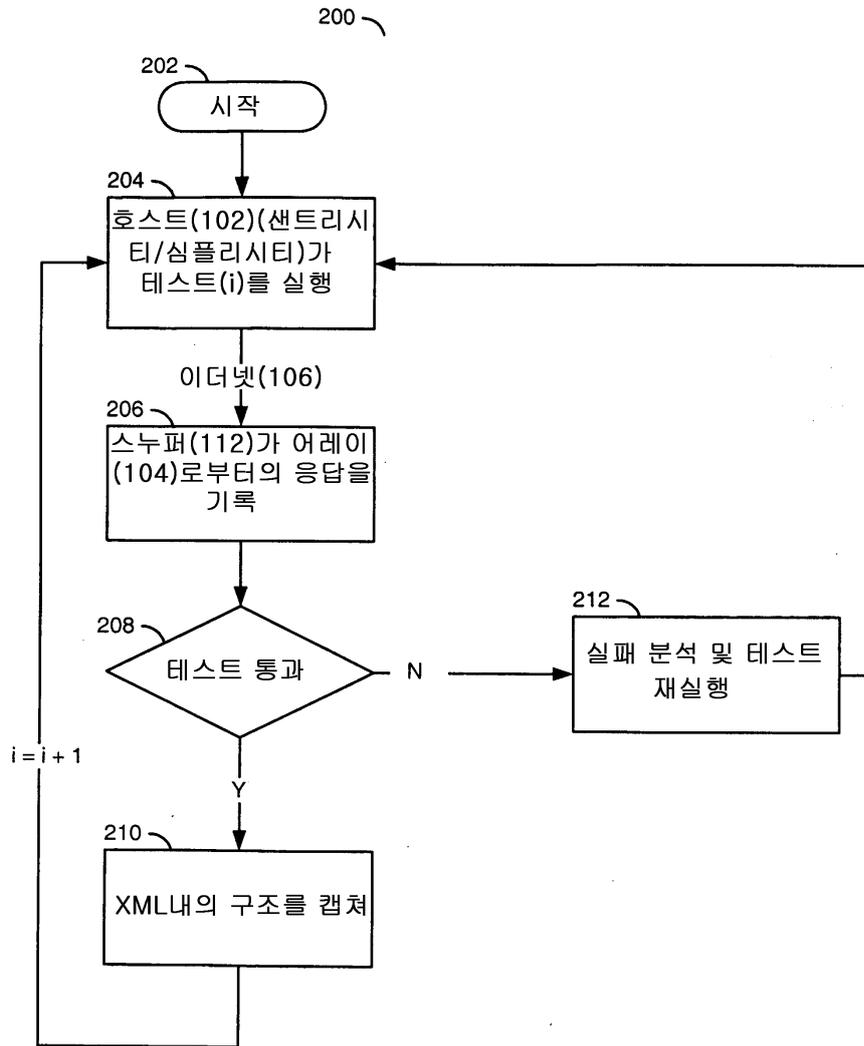
10



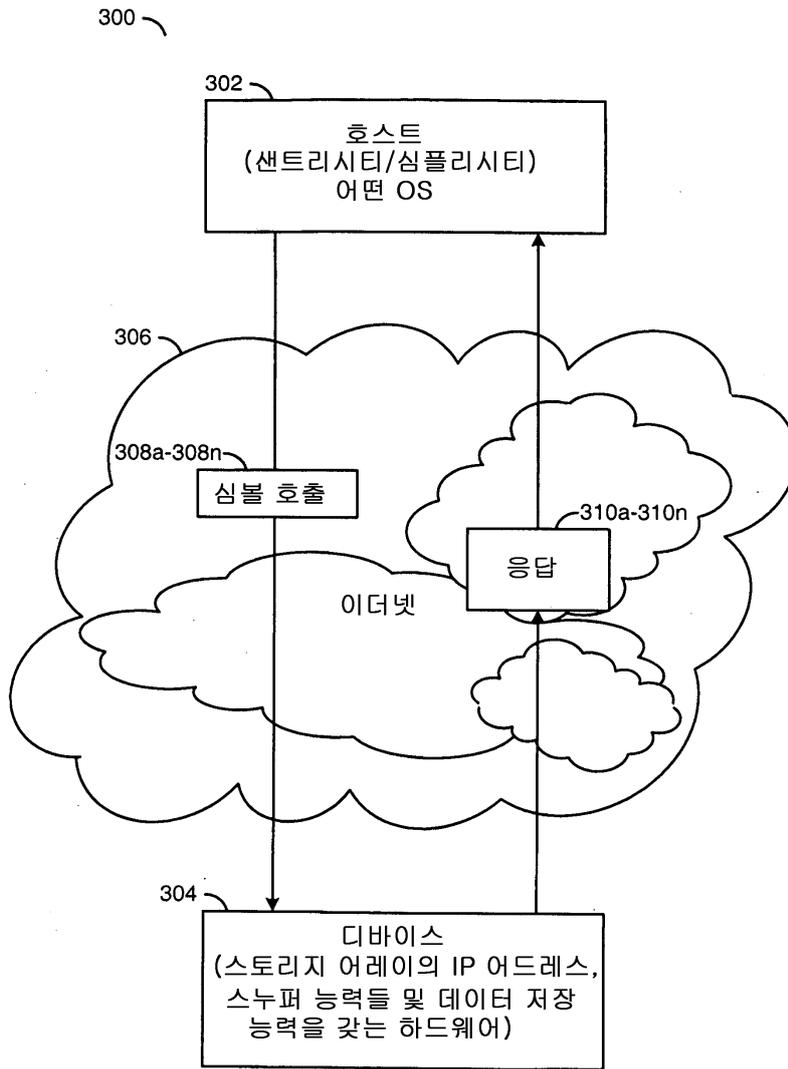
도면2



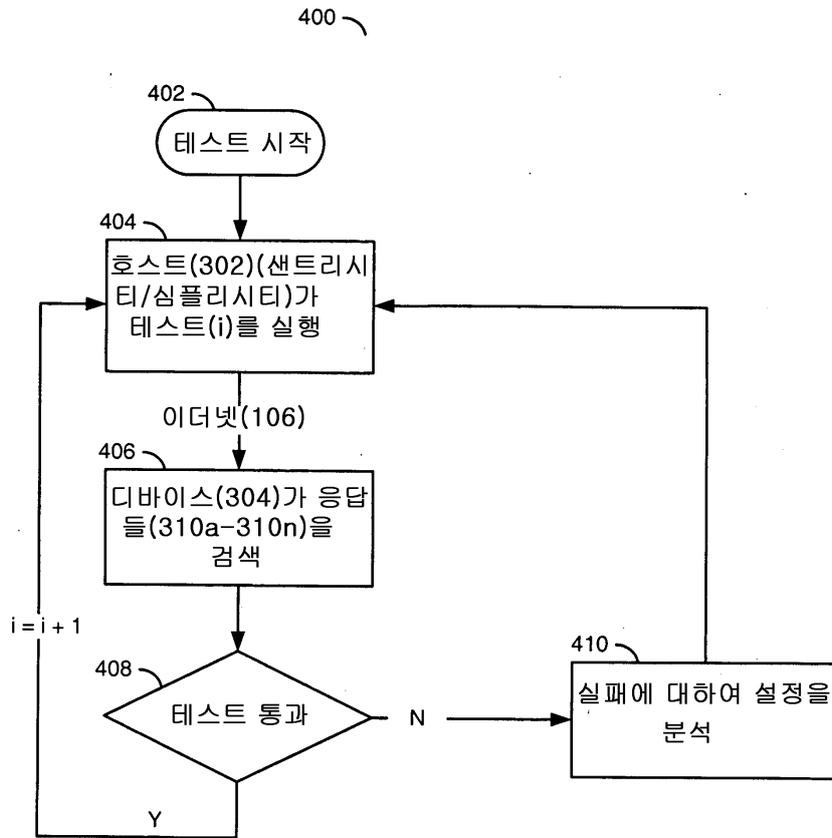
도면3



도면4



도면5



도면6

500

