



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2012년01월30일
(11) 등록번호 10-1109257
(24) 등록일자 2012년01월17일

(51) Int. Cl.

G06F 12/16 (2006.01) G06F 17/30 (2006.01)

(21) 출원번호 10-2004-0053359

(22) 출원일자 2004년07월09일

심사청구일자 2009년07월06일

(65) 공개번호 10-2005-0007179

(43) 공개일자 2005년01월17일

(30) 우선권주장

60/486,627 2003년07월10일 미국(US)

10/733,459 2003년12월10일 미국(US)

(56) 선행기술조사문헌

"Oracle8i - Replication Management API Reference - Release 2 (8.1.6)" ORACLE, December 1999, XP002307254

"Replication" Transaction Processing Concepts and Techniques Western Institute for Computer Science at Stanford Univ

US5806074 A

US5787262 A

전체 청구항 수 : 총 28 항

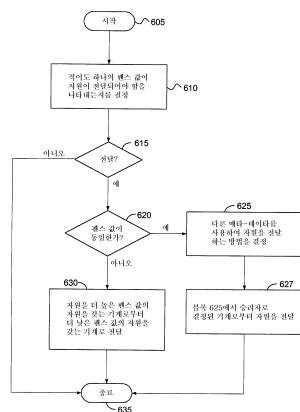
심사관 : 권오성

(54) 펜싱 및 언펜싱을 통한 복제된 정보의 권한에 대한 입상제어

(57) 요약

어느 콘텐츠가 우선 순위를 갖고 복제되는 지를 제어하기 위한 방법 및 시스템이 개시된다. 복제 집합(replica set)은 자원 집합으로 구성된다. 각각의 자원(resource)은 자원 데이터 및 자원 메타-데이터(meta-data)와 관련된 다. 파일 기반 시스템에 있어서, 자원 데이터는 파일 콘텐츠 및 속성을 포함하는 한편, 자원 메타-데이터는 복제 동안 동기화를 협의하는 것과 관련된 추가적인 속성을 포함한다. "펜스 값(fence value)"이라고 불리는 여분의 필드가 각각의 자원과 관련된 메타-데이터에 추가된다. 동기화 동안, 제1 펜스 값들이 비교된다. 가장 높은 펜스 값을 갖는 자원은 제어하고 복제되는 콘텐츠를 포함한다. 펜스 값이 동일한 경우(및 특정 값보다 더 클 경우), 제어하는 자원은 다른 메타-데이터에 기초하여 결정된다.

대표도 - 도6



특허청구의 범위

청구항 1

실행될 때 방법을 수행하는 컴퓨터 실행가능 명령어들을 저장하는 컴퓨터 판독가능 기록매체에 있어서, 상기 방법은,

복제(replica) 집합의 제1 멤버의 제1 펜스(fence) 값과 제1 메타-데이터(meta-data)를 수신하는 단계 - 상기 제1 펜스 값은 상기 복제 집합의 제1 멤버의 콘텐츠에 대해 이루어진 임의의 변경에 독립적이고, 상기 복제 집합의 제1 멤버의 정보와 상기 복제 집합의 제2 멤버의 정보 간의 충돌(conflict)을 해결하는데 이용됨 - ;

상기 제1 펜스 값을 상기 복제 집합의 제2 멤버의 제2 펜스 값과 비교하는 단계;

상기 제1 펜스 값이 상기 제2 펜스 값보다 우선권(precedence)을 갖는지를 판정하는 단계;

상기 판정에 기초하여, 상기 제1 멤버의 콘텐츠를 반영하도록 상기 제2 멤버의 콘텐츠를 변경하는 단계; 및

상기 제2 멤버의 콘텐츠에서의 변경을 표시하도록 상기 제2 멤버의 제2 메타-데이터를 변경하고, 상기 제2 펜스 값을 변경시키지 않은 채로 유지하는 단계 - 상기 제2 펜스 값은 상기 제2 멤버의 콘텐츠에 이루어진 임의의 변경에 독립적임 -;

를 포함하는, 컴퓨터 판독가능 기록매체.

청구항 2

제1항에 있어서,

상기 제2 펜스 값 및 상기 제2 메타-데이터는 상기 제2 멤버의 콘텐츠로부터 분리된 저장소에 저장되는, 컴퓨터 판독가능 기록매체.

청구항 3

제1항에 있어서,

상기 제2 멤버의 콘텐츠를 변경하는 단계는 상기 제1 멤버의 콘텐츠와 상기 제2 멤버의 콘텐츠 간의 하나 이상의 차이를 판정하는 단계를 포함하는, 컴퓨터 판독가능 기록매체.

청구항 4

제1항에 있어서,

상기 제2 멤버의 콘텐츠는 파일 데이터 및 파일 속성을 포함하는, 컴퓨터 판독가능 기록매체.

청구항 5

제1항에 있어서,

상기 제1 메타-데이터는 제1 자원을 요약하는 제1 다이제스트(digest)를 더 포함하고, 상기 제2 메타-데이터는 제2 자원을 요약하는 제2 다이제스트를 더 포함하는, 컴퓨터 판독가능 기록매체.

청구항 6

제5항에 있어서,

상기 제1 다이제스트를 상기 제2 다이제스트와 비교하는 단계와, 상기 제1 다이제스트가 상기 제2 다이제스트와 동일한 경우 상기 제2 멤버의 콘텐츠를 변경하는 단계를 바이패스(bypass)하는 단계를 더 포함하는, 컴퓨터 판독가능 기록매체.

청구항 7

제1항에 있어서,

상기 제1 펜스 값이 상기 제2 펜스 값과 동일한 경우, 상기 제1 메타-데이터를 상기 제2 메타-데이터와 비교하

여 상기 제2 멤버의 콘텐츠가 변경되어야 하는지를 판정하는 단계를 더 포함하는, 컴퓨터 판독가능 기록매체.

청구항 8

제1항에 있어서,

상기 제1 펜스 값 및 상기 제2 펜스 값 각각이 상기 복제 집합의 각 멤버의 콘텐츠의 일부로 할당되는, 컴퓨터 판독가능 기록매체.

청구항 9

실행될 때 방법을 수행하는 컴퓨터 실행가능 명령어들을 저장하는 컴퓨터 판독가능 기록매체에 있어서, 상기 방법은,

제1 기계(machine) 상에 상주하는 제1 자원이 제2 기계 상에 상주하는 제2 자원을 갱신하는데 사용되어야 하는지를 판정하는 단계 - 각각의 상기 자원은 펜스 값, 메타-데이터 및 콘텐츠와 연관되고, 각각의 상기 메타-데이터는 상기 연관된 자원의 콘텐츠가 변경될 때마다 갱신되는 하나 이상의 필드를 포함하며, 각각의 상기 펜스 값은 자신과 연관된 자원이 다른 기계 상의 자원을 갱신하는데 사용되어야 하는지를 나타내고, 상기 펜스 값은 상기 메타-데이터보다 우선권을 가지며, 각각의 상기 펜스 값은 자신과 연관된 자원의 콘텐츠에 대해 이루어진 임의의 변경에 독립적이며 상기 제1 기계의 정보와 상기 제2 기계의 정보 간의 충돌을 해결하는데 이용됨 - ;

상기 제2 자원이 전달되지 않아야 한다는 것을 나타내는 상기 제2 자원의 펜스 값에 기초하여 상기 제2 기계로부터의 전달을 막는 단계; 및

상기 제2 자원의 펜스 값보다 우선권을 갖는 상기 제1 자원의 펜스 값에 기초하여 상기 제1 자원으로부터 상기 제2 자원을 갱신하고, 상기 제1 자원의 펜스 값과 상기 제2 자원의 펜스 값을 변경하지 않은 채로 유지하는 단계

를 포함하는, 컴퓨터 판독가능 기록매체.

청구항 10

제9항에 있어서.

상기 제1 및 제2 자원의 펜스 값이 동일한 경우, 상기 메타-데이터에 기초하여 어떤 기계가 갱신될지를 판정하는 단계를 더 포함하는, 컴퓨터 판독가능 기록매체.

청구항 11

제10항에 있어서,

상기 메타-데이터는 대응 콘텐츠가 갱신되었던 최종 시각을 나타내는 논리 클록을 포함하는, 컴퓨터 판독가능 기록매체.

청구항 12

제11항에 있어서,

상기 펜스 값은 더 높은 펜스 값을 갖는 다른 자원이 다른 기계 상에 위치할 때까지 자신의 대응 자원이 또 다른 기계로 전달될 수 있다는 것을 나타내는, 컴퓨터 판독가능 기록매체.

청구항 13

제9항에 있어서,

상기 제1 자원과 연관된 메타-데이터는 자신의 대응 자원으로부터 분리된 데이터 구조에 저장되는, 컴퓨터 판독가능 기록매체.

청구항 14

제13항에 있어서,

상기 데이터 구조는 손상되거나 삭제되고,

상기 데이터 구조를 재구축하고 상기 제1 자원과 연관된 펜스 값을 감소시키는 단계를 더 포함하는, 컴퓨터 판독가능 기록매체.

청구항 15

제14항에 있어서,

상기 데이터 구조를 복수 번 재구축하는 단계와, 상기 데이터 구조가 재구축될 때마다 상기 제1 자원과 연관된 펜스 값을 감소시키는 단계를 더 포함하는, 컴퓨터 판독가능 기록매체.

청구항 16

제15항에 있어서,

상기 제2 자원의 펜스 값이 상기 제1 자원의 펜스 값보다 우선권을 갖는 경우, 상기 제2 자원으로부터 상기 제1 자원을 갱신하는 단계를 더 포함하는, 컴퓨터 판독가능 기록매체.

청구항 17

데이터를 복제하기 위한 시스템에 있어서,

제1 자원 집합을 갖는 제1 기계; 및

제2 자원 집합을 갖는 제2 기계 - 각각의 상기 기계 상의 각각의 상기 자원은 펜스 값, 메타-데이터 및 콘텐츠와 연관되고, 각각의 상기 메타-데이터는 상기 연관된 자원의 콘텐츠가 변경될 때마다 갱신되는 하나 이상의 필드를 포함하며, 각각의 상기 펜스 값은 자신과 연관된 자원이 다른 메타-데이터로부터 독립적으로 또 다른 기계 상의 자원을 갱신하는데 사용되어야 하는지를 나타내고, 각각의 상기 펜스 값은 자신과 연관된 자원의 콘텐츠에 대해 이루어진 임의의 변경에 독립적이며 상기 제1 기계의 정보와 상기 제2 기계의 정보 간의 충돌을 해결하는데 이용됨 - ;

를 포함하고,

상기 제1 및 제2 기계는,

양 기계가 포함하는 자원에 관한 정보를 통신하고,

상기 기계들 중 하나의 기계 상의 자원의 펜스 값이 다른 기계 상의 대응 자원의 펜스 값보다 우선권을 갖는지를 판정하고, 상기 기계들 중 상기 하나의 기계 상의 자원의 펜스 값이 상기 다른 기계 상의 대응 자원의 펜스 값보다 우선권을 갖는 경우 상기 다른 기계를 상기 하나의 기계 상의 자원으로 갱신하고, 그렇지 않으면, 변경되지 않은 채로 유지되는 상기 펜스 값 이외의 데이터에 기초하여 상기 기계들 상의 자원을 갱신하는 우선권 방식에 따라 구식인 자원을 각각 갱신하도록 구성되는, 데이터 복제 시스템.

청구항 18

제17항에 있어서,

상기 제1 자원 집합은 백업으로부터 로딩되고, 상기 제1 자원 집합의 펜스 값은 상기 제1 자원 집합이 임의의 다른 자원 집합보다 우선권을 갖도록 설정되어, 상기 자원 집합들에 대응하는 임의의 다른 기계 상의 임의의 다른 자원 집합이 상기 제1 자원 집합으로부터 갱신되는, 데이터 복제 시스템.

청구항 19

제17항에 있어서,

상기 제1 자원 집합의 펜스 값이 다른 기계상의 대응 자원보다 우선권을 갖는 것으로 표시되어, 상기 대응 자원이 상기 제1 자원 집합으로부터 갱신되는, 데이터 복제 시스템.

청구항 20

데이터를 복제하는 컴퓨터 구현 방법에 있어서,

복제 집합의 제1 멤버의 제1 펜스 값과 제1 메타-데이터를 수신하는 단계 - 상기 제1 펜스 값은 상기 복제 집합의 제1 멤버의 콘텐츠에 대해 이루어진 임의의 변경에 독립적이고, 상기 복제 집합의 제1 멤버의 정보와 상기 복제 집합의 제2 멤버의 정보 간의 충돌을 해결하는데 이용됨 - ;

상기 제1 펜스 값을 상기 복제 집합의 제2 멤버의 제2 펜스 값과 비교하는 단계;

상기 제1 펜스 값이 상기 제2 펜스 값보다 우선권을 갖는지를 판정하는 단계;

상기 판정에 기초하여, 상기 제1 멤버의 콘텐츠를 반영하도록 상기 제2 멤버의 콘텐츠를 변경하는 단계; 및

상기 제2 멤버의 콘텐츠에서의 변경을 표시하도록 상기 복제 집합의 제2 멤버의 제2 메타-데이터를 변경하고, 상기 제2 펜스 값을 변경시키지 않은 채로 유지하는 단계 - 상기 제2 펜스 값은 상기 제2 멤버의 콘텐츠에 이루어진 임의의 변경에 독립적임 -;

를 포함하는, 컴퓨터 구현 방법.

청구항 21

제20항에 있어서,

상기 제2 메타-데이터는 상기 제2 멤버의 콘텐츠로부터 분리된 저장소에 저장되는, 컴퓨터 구현 방법.

청구항 22

제20항에 있어서,

상기 제2 멤버의 콘텐츠를 변경하는 단계는 상기 제1 멤버의 콘텐츠와 상기 제2 멤버의 콘텐츠 간의 하나 이상의 차이를 판정하는 단계를 포함하는, 컴퓨터 구현 방법.

청구항 23

제22항에 있어서,

상기 제1 멤버의 콘텐츠와 상기 제2 멤버의 콘텐츠가 동일한, 컴퓨터 구현 방법

청구항 24

제20항에 있어서,

상기 제2 멤버의 콘텐츠는 파일 데이터 및 파일 속성을 포함하는, 컴퓨터 구현 방법.

청구항 25

제20항에 있어서,

상기 제1 메타-데이터는 제1 자원을 요약하는 제1 다이제스트를 포함하고, 상기 제2 메타-데이터는 제2 자원을 요약하는 제2 다이제스트를 포함하는, 컴퓨터 구현 방법.

청구항 26

제25항에 있어서,

상기 제1 다이제스트와 상기 제2 다이제스트를 비교하는 단계와, 상기 다이제스트들이 동일한 경우 상기 제2 멤버의 콘텐츠를 변경하는 단계를 바이패스하는 단계를 더 포함하는, 컴퓨터 구현 방법.

청구항 27

제20항에 있어서,

상기 제1 펜스 값이 상기 제2 펜스 값과 동일한 경우, 상기 제1 메타-데이터를 상기 제2 메타-데이터와 비교하여 상기 제2 멤버의 콘텐츠가 변경되어야 하는지를 판정하는 단계를 더 포함하는, 컴퓨터 구현 방법.

청구항 28

제20항에 있어서,

상기 제1 펜스 값 및 상기 제2 펜스 값 각각이 상기 복제 집합의 각 멤버의 일부로 할당되는, 컴퓨터 구현 방법.

청구항 29

삭제

청구항 30

삭제

청구항 31

삭제

청구항 32

삭제

청구항 33

삭제

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

[0019] 본 출원은 여기에 전체로 통합되고, 펜싱 및 언펜싱을 통한 복제된 정보의 권한에 대한 입상 제어(GRANULAR CONTROL OVER THE AUTHORITY OF REPLICATED INFORMATION VIA FENCING AND UNFENCING)라는 명칭의 2003년 7월 10일에 제출된 미국 특허 출원 60/486,627호의 이익을 청구한다.

[0020] 본 발명은 일반적으로 컴퓨팅 장치에 관한 것이며, 더 상세하게는 자원 복제 시스템에 관한 것이다.

[0021] 기회주의적인 멀티-마스터 복제 시스템은 주어진 복제 집합에 관여하는 임의의 기계 상에서 복제된 콘텐츠에 대한 무제한 변경을 허용한다. 이 잠재적으로 충돌하는 변경은 모든 충돌 상황에 대하여 어느 충돌하는 변경이 다른 변경보다 우선하는지를 정의한 충돌 해결 기준의 집합을 사용하는 복제 시스템의 제어 하에서 조정된다. 종전에, 사용된 주요 충돌 해결 기준은 물리적 또는 논리적인 변경 시각이었고, 가장 최근의 변경이 다른 모든 변경보다 우선하였다.

발명이 이루고자 하는 기술적 과제

[0022] 그러나, 사용자 또는 어플리케이션이, 어느 동시적인 업데이트가 다른 것보다 우선 순위를 갖고 반대로 어느 업데이트가 우선 순위를 양보해야 하는지를 제어하는 추가적인 방법을 갖기 원할 수 있는 다수의 경우가 존재한다. 어느 콘텐츠가 우선 순위를 갖고 복제되는지를 제어하기 위한 유연한 방법 및 시스템이 요구된다.

발명의 구성 및 작용

[0023] 간략하게, 본 발명은 어느 콘텐츠가 우선 순위를 갖고 복제되는지를 제어하기 위한 방법 및 시스템을 제공한다. 복제 집합은 자원의 집합으로 구성된다. 각각의 자원은 자원 데이터 및 자원 메타-데이터와 관련된다. 파일의 경우, 자원 데이터는 파일 콘텐츠 및 속성을 포함하는 한편, 자원 메타-데이터는 복제 동안 동기화를 협의하는 것과 관련된 추가적인 속성을 포함한다. "펜스 값(fence value)"라고 불리는 여분의 필드가 각각의 자원과 관련된 메타-데이터에 추가된다. 동기화 동안, 펜스 값들이 비교된다. 가장 높은 펜스 값을 갖는 자원이 제어하고 복제되는 콘텐츠를 포함한다. 펜스 값이 동일한 경우(및 특정 값보다 클 경우), 다른 메타-데이터에 기초하여, 제어하는 자원이 결정된다.

- [0024] 펜스 값은 콘텐츠에 대한 로컬 변경에 독립적이다. 즉, 콘텐츠 내의 로컬 변경이 다른 메타-데이터(예를 들어, 시각 소인, 클릭 값 등)에 영향을 줄 수 있는 반면, 콘텐츠 내의 로컬 변경은 달리 지시되지 않는 한 펜스 값에 영향을 주지 않는다.
- [0025] 본 발명의 일 양태에서, 자원은 자신이 언펜싱되는 것을 나타내는 펜스 값을 가질 수 있다. 자원이 언펜싱되는 경우, 이것은 자원이 자신이 저장된 기계로부터 전송되지 않아야 한다는 것을 나타낸다. 상충하는 자원이 동기화를 위해 수신되면, 언펜싱된 자원은 펜싱된 자원에게 패배한다(그리고 동기화 동안에 대체됨).
- [0026] 본 발명의 또다른 양태에서, 승리하는 자원을 갖는 기계, 및 패배하는 콘텐츠를 갖는 기계 상의 자원들 간의 차이만 전송된다. 예를 들어, 자원 메타-데이터는 자원 콘텐츠를 전송하지 않고 전송될 수 있다. 또다른 예로서, 자원 콘텐츠에서의 차이는 동기화 동안에 전송될 수 있다.
- [0027] 도면과 관련된 다음의 상세한 설명으로부터 다른 장점이 명백해질 것이다.
- [0028] 예시적인 오퍼레이팅 환경
- [0029] 도 1은 본 발명이 구현될 수 있는 적합한 컴퓨팅 시스템 환경(100)의 예를 도시한다. 컴퓨팅 시스템 환경(100)은 단지 적절한 컴퓨팅 환경의 일 예이며 본 발명의 사용 또는 기능의 범위에 제한을 가하도록 의도된 것은 아니다. 컴퓨팅 환경(100)은 예시적인 오퍼레이팅 환경(100)에 도시된 컴포넌트들 중의 임의의 하나 또는 조합에 관하여 임의의 종속성(dependency) 또는 요구사항(requirement)을 갖는 것으로 해석되어서는 안된다.
- [0030] 본 발명은 많은 다른 범용 또는 특수목적 컴퓨팅 시스템 환경들 또는 구성들과 함께 동작될 수 있다. 본 발명과 함께 사용하기에 적합할 수 있는 잘 알려진 컴퓨팅 시스템, 환경, 및/또는 구성의 예로는, 퍼스널 컴퓨터, 서버 컴퓨터, 핸드헬드(hand-held) 또는 랩탑 장치, 멀티프로세서 시스템, 마이크로프로세서-기반 시스템, 셋탑 박스(set top box), 프로그램가능한 가전제품(programmable consumer electronics), 네트워크 PC, 미니컴퓨터, 메인프레임 컴퓨터, 상기의 시스템 또는 장치 중의 임의의 것을 포함하는 분산형 컴퓨팅 환경 등이 포함될 수 있지만, 이에 한정되지 않는다.
- [0031] 본 발명은 컴퓨터에 의해 실행되는, 프로그램 모듈과 같은 컴퓨터 실행가능 명령과 일반적으로 관련하여 기술될 수 있다. 일반적으로, 프로그램 모듈은 특정 태스크를 수행하거나 특정 추상 데이터 유형을 구현하는 루틴, 프로그램, 오브젝트, 컴포넌트, 데이터 구조 등을 포함한다. 본 발명은 또한 통신 네트워크를 통해 링크된 원격 프로세싱 장치에 의해 태스크를 수행하는 분산형 컴퓨팅 환경에서 실행될 수 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈은 메모리 저장 장치를 포함하는 로컬 및 원격 컴퓨터 저장 매체 내에 위치할 수 있다.
- [0032] 도 1을 참조하면, 본 발명을 구현하기 위한 예시적인 시스템은 컴퓨터(110)의 형태의 범용 컴퓨팅 장치를 포함한다. 컴퓨터(110)의 컴포넌트들로는, 프로세싱 유닛(120), 시스템 메모리(130), 및 시스템 메모리를 포함하는 다양한 시스템 컴포넌트를 프로세싱 유닛(120)에 연결시키는 시스템 버스(121)가 포함될 수 있지만, 이에 한정되는 것은 아니다. 시스템 버스(121)는 다양한 버스 아키텍처 중의 임의의 것을 사용하는 로컬 버스, 주변 버스, 및 메모리 버스 또는 메모리 컨트롤러를 포함하는 몇가지 유형의 버스 구조 중의 임의의 것일 수 있다. 예로서, 이러한 아키텍처는 산업 표준 아키텍처(ISA) 버스, 마이크로 채널 아키텍처(MCA) 버스, 인헨스드 ISA(Enhanced ISA; EISA) 버스, 비디오 일렉트로닉스 표준 어소시에이션(VESA) 로컬 버스, 및 메자닌(Mezzanine) 버스로도 알려진 주변 컴포넌트 상호접속(PCI) 버스를 포함하지만, 이에 한정되는 것은 아니다.
- [0033] 컴퓨터(110)는 통상적으로 다양한 컴퓨터 판독가능 매체를 포함한다. 컴퓨터 판독가능 매체는 컴퓨터(110)에 의해 액세스될 수 있는 임의의 이용가능한 매체일 수 있으며, 휘발성 및 비휘발성 매체, 분리형(removable) 및 비분리형(non-removable) 매체를 둘다 포함한다. 예로서, 컴퓨터 판독가능 매체는 컴퓨터 저장 매체 및 통신 매체를 포함할 수 있지만, 이에 한정되는 것은 아니다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령, 데이터 구조, 프로그램 모듈 또는 다른 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현되는 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 둘다 포함한다. 컴퓨터 저장 매체는 RAM, ROM, EEPROM, 플래시 메모리 또는 기타 메모리 기술, CD-ROM, DVD(digital versatile disk) 또는 기타 광학 디스크 저장장치, 자기 카세트, 자기 테이프, 자기 디스크 저장장치 또는 기타 자기 저장장치, 또는 컴퓨터(110)에 의해 액세스될 수 있고 원하는 정보를 저장하는 데 사용될 수 있는 임의의 기타 매체를 포함할 수 있지만, 이에 한정되지 않는다. 통신 매체는 통상적으로 반송파 또는 기타 전송 메커니즘 등의 변조된 데이터 신호에 컴퓨터 판독가능 명령, 데이터 구조, 프로그램 모듈, 또는 다른 데이터를 구현하며, 임의의 정보 전달 매체를 포함한다. "변조된 데이터 신호"라는 용어는 신호 내에 정보를 인코딩하도록 설정되거나 변환된 특성을 하나 또는 그 이상을 갖는 신호를 의미한다. 예로서, 통신 매체는 유선 네트워크 또는 직접 유선 접속 등의 유선 매체와, 음향, RF, 적외선 및 기타

무선 매체 등의 무선 매체를 포함하지만, 이에 한정되지 않는다. 상술한 것들 중의 임의의 조합이 컴퓨터 판독가능 매체의 범위 내에 포함되어야 한다.

[0034] 시스템 메모리(130)는 ROM(131) 및 RAM(132) 등의 휘발성 및/또는 비휘발성 메모리의 형태의 컴퓨터 저장 매체를 포함한다. 시동중과 같은 때에 컴퓨터(110) 내의 구성요소들간에 정보를 전송하는 것을 돕는 기본 루틴을 포함하는 기본 입출력 시스템(133; BIOS)은 일반적으로 ROM(131)에 저장된다. RAM(132)은 일반적으로 프로세싱 유닛(120)에 즉시 액세스될 수 있고 및/또는 프로세싱 유닛(120)에 의해 현재 작동되는 프로그램 모듈 및/또는 데이터를 포함한다. 예로서, (한정하고자 하는 것은 아님) 도 1은 오퍼레이팅 시스템(134), 어플리케이션 프로그램(135), 기타 프로그램 모듈(136), 및 프로그램 데이터(137)를 도시한다.

[0035] 컴퓨터(110)는 또한 다른 분리형/비분리형, 휘발성/비휘발성 컴퓨터 저장 매체를 포함할 수 있다. 단지 예로서, 도 1에는 비분리형 비휘발성 자기 매체로부터 판독하거나 그 자기 매체에 기록하는 하드 디스크 드라이브(140), 분리형 비휘발성 자기 디스크(152)로부터 판독하거나 그 자기 디스크에 기록하는 자기 디스크 드라이브(151), 및 CD-ROM 또는 기타 광학 매체 등의 분리형 비휘발성 광학 디스크(156)로부터 판독하거나 그 광학 디스크에 기록하는 광학 디스크 드라이브(155)가 도시되어 있다. 예시적인 오퍼레이팅 환경에서 사용될 수 있는 다른 분리형/비분리형, 휘발성/비휘발성 컴퓨터 저장 매체는 자기 테이프 카세트, 플래쉬 메모리 카드, DVD(Digital versatile disk), 디지털 비디오 테이프, 고체 RAM, 고체 ROM 등을 포함하지만 이에 한정되지 않는다. 하드 디스크 드라이브(141)는 일반적으로 인터페이스(140)와 같은 비분리형 메모리 인터페이스를 통해 시스템 버스(121)에 접속되고, 자기 디스크 드라이브(151) 및 광학 디스크 드라이브(155)는 일반적으로 인터페이스(150)와 같은 분리형 메모리 인터페이스에 의해 시스템 버스(121)에 접속된다.

[0036] 앞서 기술되고 도 1에 도시된 드라이브 및 그 관련 컴퓨터 저장 매체는 컴퓨터(110)를 위한 컴퓨터 판독가능 명령, 데이터 구조, 프로그램 모듈 및 기타 데이터의 저장을 제공한다. 도 1에서, 예를 들어, 하드 디스크 드라이브(141)는 오퍼레이팅 시스템(144), 어플리케이션 프로그램(145), 기타 프로그램 모듈(146), 및 프로그램 데이터(147)를 저장하는 것으로 도시된다. 이들 컴포넌트는 오퍼레이팅 시스템(134), 어플리케이션 프로그램(135), 기타 프로그램 모듈(136), 및 프로그램 데이터(137)와 동일할 수도 있고 다를 수도 있다. 오퍼레이팅 시스템(144), 어플리케이션 프로그램(145), 다른 프로그램 모듈(146), 및 프로그램 데이터(147)는 최소한 다른 복사본(different copies)임을 나타내기 위하여 다른 번호를 부여하였다. 사용자는 일반적으로 마우스, 트랙볼, 또는 터치 패드라 불리는 포인팅 장치(161) 및 키보드(162)와 같은 입력 장치를 통해 컴퓨터(110)에 명령 및 정보를 입력할 수 있다. (도시되지 않은) 기타 입력 장치는 마이크로폰, 조이스틱, 게임 패드, 위성 안테나, 스캐너, 핸드헬드 PC의 접촉식 스크린 또는 기타 기록 테이블 등을 포함할 수 있다. 이들 입력 장치 및 그외의 입력 장치는 시스템 버스에 연결된 사용자 입력 인터페이스(160)를 통해 종종 프로세싱 유닛(120)에 접속되지만, 병렬 포트, 게임 포트 또는 유니버설 시리얼 포트(USB)와 같은 기타 인터페이스 및 버스 구조에 의해 접속될 수 있다. 모니터(191) 또는 다른 유형의 디스플레이 장치는 또한 비디오 인터페이스(190) 등의 인터페이스를 통해 시스템 버스(121)에 접속된다. 모니터외에도, 컴퓨터는 또한 출력 주변 인터페이스(190)를 통해 접속될 수 있는 스피커(197) 및 프린터(196) 등의 기타 주변 출력 장치를 포함할 수 있다.

[0037] 컴퓨터(110)는 원격 컴퓨터(180)와 같은 하나 이상의 원격 컴퓨터로의 논리적 접속을 이용한 네트워크 환경에서 동작할 수 있다. 원격 컴퓨터(180)는 퍼스널 컴퓨터, 서버, 라우터, 네트워크 PC, 피어(peer) 장치, 또는 기타 공통 네트워크 노드일 수 있으며, 비록 도 1에는 메모리 저장 장치(181)만이 도시되어 있지만, 컴퓨터(110)에 관하여 상술한 구성요소 중 다수 또는 모든 구성요소를 일반적으로 포함할 수 있다. 도 1에 도시된 논리적 접속은 근거리 통신망(LAN; 171) 및 원거리 통신망(WAN; 173)을 포함하지만, 그 외의 네트워크를 포함할 수도 있다. 이러한 네트워크 환경은 사무실, 기업 광역 컴퓨터 네트워크(enterprise-wide computer network), 인트라넷, 및 인터넷에서 일반적인 것이다.

[0038] LAN 네트워크 환경에서 사용되는 경우, 컴퓨터(110)는 네트워크 인터페이스 또는 어댑터(170)를 통해 LAN(171)에 접속된다. WAN 네트워크 환경에서 사용되는 경우, 컴퓨터(110)는 일반적으로 인터넷 등의 WAN(173)을 통해 통신을 구축하기 위한 모뎀(172) 또는 기타 수단을 포함한다. 내장형 또는 외장형일 수 있는 모뎀(172)은 사용자 입력 인터페이스(160) 또는 기타 적절한 메커니즘을 통해 시스템 버스(121)에 접속될 수 있다. 네트워크 환경에서, 컴퓨터(110)에 관하여 도시된 프로그램 모듈 또는 그 일부분은 원격 메모리 저장 장치에 저장될 수 있다. 예로서 (한정하고자 하는 것은 아님), 도 1은 메모리 장치(181)에 상주하는 원격 어플리케이션 프로그램(185)을 도시한다. 도시된 네트워크 접속은 예시적인 것이며, 컴퓨터들간의 통신 링크를 구축하는 그 외의 수단이 사용될 수 있다.

- [0039] 자원 복제로 업데이트 제어
- [0040] 도 2는 본 발명의 다양한 양태에 따라 자원을 복제하는 2개의 기계를 포함하는 자원 복제 시스템을 나타내는 블록도이다. 기계(201 및 202) 양자는 자원 A를 복제하고 콘텐츠 x를 갖는 것에서 콘텐츠 y, u 또는 z를 갖는 것으로 동시에 업데이트한다. 콘텐츠 x, y, u 및 z는 자원 A에 대하여 예를 들어, 상이한 버전 일련 번호 및 클럭에 대응할 수 있다.
- [0041] "기계"라는 용어는 단순히 물리적인 기계로 제한되지 않는다. 오히려, 단일 물리적인 기계는 다수의 가상 기계를 포함할 수 있다. 여기서 사용되는 것으로서, 하나의 기계로부터 다른 기계로의 복제는 동일한 복제 집합의 하나 이상의 구성원을 하나의 가상 또는 물리적인 기계로부터 또다른 가상 또는 물리적인 기계로 복제하는 것을 의미한다. 단일의 물리적인 기계는 동일한 복제 집합의 다수의 구성원을 포함할 수 있다. 따라서, 복제 집합의 구성원을 복제하는 것은 동일한 복제 집합의 2개 이상의 구성원을 포함하는 단일의 물리적인 기계의 구성원들을 동기화하는 것을 포함할 수 있다.
- [0042] 복제 시스템은 일반적으로 각각의 자원과 관련된 2개의 데이터 집합, 자원 데이터 및 자원 메타-데이터를 유지한다. 파일 시스템에서 명명된 파일에 기초한 데이터 저장을 포함하는 복제 시스템에서, 자원 데이터는 파일 콘텐츠 뿐만 아니라 파일 콘텐츠와 관련되어 파일 시스템 상에 저장된 임의의 파일 속성도 포함할 수 있다. 파일 속성은 액세스 제어 리스트(ACL), 생성/수정 시각, 및 파일과 관련된 기타 데이터를 포함할 수 있다. 파일 시스템에서 명명된 파일에 기초하지 않은 데이터 저장(예를 들어, 자원이 데이터베이스에 저장되는 데이터 저장, 또는 객체기반 데이터 저장)을 포함하는 복제 시스템에서, 데이터 저장에 적합한 자원 데이터가 저장된다. 본 명세서에서는, 파일 시스템 내의 파일에 기초한 복제 시스템이 종종 설명을 위해 사용되지만, 콘텐츠를 저장할 수 있는 임의의 데이터 저장이 본 발명의 취지 또는 범주를 벗어나지 않고 사용될 수 있음을 인식할 것이다.
- [0043] 자원 메타-데이터는 복제 동안 동기화를 협의하는 것과 관련된 추가적인 속성의 집합을 포함한다. 각각의 자원에 있어서, 자원 메타-데이터는 GUID(globally unique identifier), 자원이 삭제되었는 지의 여부, 변경의 저작권과 함께 버전 일련 번호, 변경이 발생한 시각을 반영하는 클럭 값, 및 자원 데이터의 값을 요약하고 자원 콘텐츠에 대한 서명을 포함할 수 있는 다이제스트(digest)와 같은 다른 필드를 포함할 수 있다. 다이제스트는 예를 들어, 복제 동기화 동안 데이터 전송을 우회하기 위해 빠른 비교에 사용될 수 있다. 수신 기계 상의 자원이 소스 기계 상의 콘텐츠와 동기화되면(예를 들어, 다이제스트에 의해 나타난 바와 같이), 네트워크 오버헤드는 자원 데이터 자체를 전송하지 않고 자원 메타-데이터만 전송함으로써 최소화될 수 있다. 수신 기계가 후속하는 복제 실행에서 소스 기계 상에 포함되는 메타-데이터를 반영할 수 있도록 자원 메타-데이터가 전송된다. 이것은 수신 기계가 예를 들어, 후속하는 복제 실행에서 소스 기계가 되는 것을 허용한다. 소스 메타-데이터는 본 발명의 취지 또는 범주에서 벗어나지 않고 자원 데이터와 함께, 또는 자원 데이터로부터 분리되어 저장될 수 있다.
- [0044] 일반적으로, 광역 분산 시스템에서, 미세한 입상(very granular) 레벨의 클럭 동기화를 가정하는 것은 불가능하다. 이것은 동시적인 업데이트 및 생성에 대한 승리자를 결정하기 위하여 복제 시스템이 글로벌 클럭을 사용하는 것을 막는다. 복제기는 일반적으로 분산 콘텐츠에 대한 메타-데이터에 소인(stamp)되는 논리적인 분산 클럭을 사용한다. 변경 시각의 물리적인 로컬 클럭에 의해 덮어쓰여지는 것에 대립하여, 논리적인 클럭은 콘텐츠가 업데이트될 때 증가된다. 따라서, 논리적인 클럭은 인과관계(causality)를 중요시 한다. 동일한 콘텐츠에 대한 업데이트는 클럭 값을 증가시키는 것으로 소인(stamp)된다. 도 2의 A:x의 클럭 값은 예를 들어, 4일 수 있다. 복제 시스템은 A:y 및 A:z와 관련된 클럭 값이 4보다 큰 지를 확인한다. 이 클럭 값들 간의 관계는 임의적이다. 그 값들은 독립적으로 할당된 것이므로{예를 들어, 기계(201)가 A:y에 대하여 클럭 값을 할당하는 한편, 기계(202)가 A:z에 대하여 클럭 값을 할당함} 동일할 수도 있고 아닐 수도 있다.
- [0045] 일반적인 복제 동기화에서, 클럭 값은 최종-기록자-승리 충돌 해결 전략에 기초하여 충돌 승리자를 결정하는 데 사용될 수 있다. 가장 높은 클럭 값을 갖는 데이터는 더 낮은 클럭 값을 갖는 복제된 데이터보다 더 최근의 데이터를 나타낼 수 있다. 최종-기록자-승리 전략은 인과관계(causality)를 유지하기 때문에 논리적인 클럭과 밀접하게 관련된다.
- [0046] 본 발명의 양태에 따라, 자원 메타-데이터는 "펜스 값(fence value)"이라고 불리는 숫자 필드로 증가된다. 펜스 값은 각각의 자원 또는 그것의 일부에 할당될 수 있다. 펜스 값은 이하에 정의된 규칙 집합에 따라, 다른 메타-데이터와 결합하여 충돌 해결동안 사용된다.

- [0047] 본 발명의 일 실시예에서, 규칙에 따라, 펜스 값은 0 또는 1로 초기화된다. 값 0은 자원이 송신되지 않거나 (복제 메커니즘을 통해) 다른 기계에 보여져야 한다는 것을 나타낸다. 값 1은 자원이 복제될 수 있고 다른 기계에 보여져야 한다는 것을 나타낸다. 펜스 값 0을 갖는 자원은 슬레이브 자원(slave resource)으로 간주될 수 있는 한편, 펜스 값 1을 갖는 자원은 마스터 자원(master resource)으로 간주될 수 있다.
- [0048] 2개의 기계 간의 복제 실행 동안, 그 기계들 중 하나의 기계{예를 들어, 기계(201)} 상의 자원(예를 들어, 자원 A)이 다른 기계{예를 들어, 기계(202)} 상의 동일한 자원보다 더 높은 펜스 값을 갖는다면, 더 높은 펜스 값을 갖는 자원은 더 낮은 펜스 값을 갖는 자원을 가진 기계에 복제된다. 다시 말하면, 더 높은 펜스 값을 갖는 자원은 다른 자원 메타-데이터가 제공되는 것과 상관없이 승리한다(그리고 복제됨).
- [0049] 2개의 기계 상에 이미 존재하지만 서로 상이한 자원을 복제하는 경우에는, 자원을 동기화하기 위하여 가능한 한의 소량 데이터를 전달하는 것을 시도하는 메커니즘이 사용될 수 있다. 예를 들어, 자원과 관련된 모든 데이터를 송신하는 것 대신에, 복제 메커니즘은 송신하는 기계의 자원 내의 어떤 데이터가 수신하는 기계의 자원 내의 데이터와 상이한지를 결정하여, 수신하는 기계의 자원을 업데이트하기 위하여, 하나 이상의 차이 또는 델타(delta)를 송신할 수 있다.
- [0050] 복제 실행 동안, 펜스 값들이 동일하고 0보다 더 클 경우, 어느 자원이 승리하는(그리고 복제됨) 가는 각각의 자원과 관련된 다른 자원 메타-데이터에 의존한다. 다시 말하면, 펜스 값이 동일할 때, 복제는 복제와 관련된 일반적인 규칙에 따라 진행된다.
- [0051] 따라서, 펜스 값들은 충돌 해결에서 우선 순위를 갖기 때문에, 충돌 해결 프로세스에 대하여 미세하게 낱알모양으로 된(fine-grained) 제어를 공급한다. 즉, 2개의 자원의 메타-데이터가 비교될 때, 가장 높은 펜스 값을 갖는 자원이 우선 균위를 갖는다. 논리적인 클럭과 같은 다른 속성은 펜스가 동일할 때만 비교된다.
- [0052] 펜스 값이 단지 증가될 수 있거나 0으로 재설정될 수 있는 일 구현에서와 마찬가지로, 펜스 값은 논리적인 클럭과 유사한 속성을 가질 수 있다. 예를 들어, 펜스 값은 사용자 또는 프로세스가 복제 시스템에게 명령할 때 증가될 수 있다. 이것은 종종 "자원을 펜싱한다" 또는 단순히 "펜싱"이라고 불린다. 펜스 값의 증가는 자원 데이터의 업데이트에 독립적이다. 펜스의 증가는 복제에 의해 보일 수 있게 된다(예를 들어, 메타-데이터에서 전 송됨). 펜싱은 빈번한 동작이 아니므로, 벽 시계 시각의 정수 표현이 펜스를 (현재 펜스 값 플러스 1)과 (현재 벽 시계 시각)의 최대값으로 증가시키는 데 사용될 수 있다.
- [0053] 펜스 값은 사용자 또는 프로세스가 복제 시스템에게 명령할 때 0으로 재설정될 수 있다. 이것은 종종 "자원을 언펜싱한다" 또는 단순히 "언펜싱"이라고 불린다. 언펜싱된(즉, 그 펜스 값이 0으로 재설정됨) 슬레이브 모드 자원은 자신을 소유하는 기계 외부의 임의의 다른 기계에 복제되지 않을 수 있다. 이것은 슬레이브 모드 자원을 외부적으로 보여지는 것으로부터 막는다. 따라서, 펜스 값 재설정은 복제에 의해 보이지 않는다.
- [0054] 펜싱 및 언펜싱을 제외하고, 펜스 값은 복제 동안 상수로 남는다. 특히, 양수 펜스 값은 자원 데이터가 업데이트되거나 논리적인 클럭이 변경될 때 변하지 않는다. 0인 펜스 값도 슬레이브 기계에 대한 콘텐츠 업데이트에 대해서 변하지 않아야 한다.
- [0055] 하나의 기계 상의 자원을 펜싱하는 것은 복제 집합의 다른 구성원 상의 자원에 대한 동시적인 업데이트에 상관없이 상술한 자원의 복제를 강요하는 것을 허용함을 인식할 것이다. 자원을 언펜싱하는 것은 임의의 다른 복제된 자원이 (언펜싱된) 로컬 자원보다 우선 순위를 갖는 것을 강요하는 것을 허용하고, 언펜싱된 자원을 복제로부터 막는다는 것도 인식할 것이다.
- [0056] 펜스는 선택된 자원을 펜싱하는 복제 시스템 어플리케이션 프로그래밍 인터페이스(API)를 통해 기존 자원 상에서 설정될 수 있다. 또한, 펜스는 자원이 복제 시스템에 보일 때 (자원 이름 및 속성과 같은) 특정 매개변수와 일치하는 자원을 펜싱하는 API를 노출함으로써 미래의 자원 상에서 설정될 수 있다. 미래의 펜싱 및 언펜싱에 대한 제어는 지정하는 스티키니스(stickiness)에 의해 제공될 수도 있다. 펜싱/언펜싱 된 디렉토리 하의 자원과 같이 간접적으로 관련된 자원의 생성은 부모 상에 설정된 정책에 기초하여 펜싱 값을 승계(inherit)할 수 있다.
- [0057] 언펜싱은 복제 파트너의 마스터 또는 슬레이브 행위를 제어하는 입상 방법도 제공한다. 선택된 자원 상의 펜스를 설정하는 것은 효과적으로 기계를 콘텐츠의 버전의 마스터로 만든다(콘텐츠가 처음으로 업데이트 될때까지). 선택된 자원을 언펜싱하는 것은 기계를 선택된 자원과 관련된 콘텐츠에 대하여 슬레이브로서 행동하게 한다.
- [0058] 펜싱 및 언펜싱이 백업 복구, 대역 외(out-of-band) 복제, 더 새로운 버전의 복제기로의 업그레이드, 및 관리자

제어 제공을 포함한 다수 영역에서 사용될 수 있음을 인식할 것이다.

- [0059] 언펜싱된 자원(예를 들어, 펜스 값 0을 갖는 것)을 포함하는 기계는 논리적으로 슬레이브 상에서 업데이트된 자원에 대하여 언펜싱된 자원 펜스 값을 펜싱된 자원 값(예를 들어, 1)으로 변경하는 것을 선택할 수 있다. 이것은 예를 들어, 업데이트를 보이게하기 위하여 행해질 수 있다.
- [0060] 도 3은 본 발명의 다양한 양태에 따라 2개의 기계가 양자에 포함된 자원을 중재하는 시스템을 나타내는 블록도이다. 도 3에서, 각각의 기계는 자신의 자원 버전을 갖고, 그 자원은 동일한 이름 또는 식별자를 갖는다. 중재 마지막에, 목표는 콘텐츠 x 또는 y를 갖는 복제된 자원을 갖는 것이다. 도 3에 도시되는 시스템에서, 기계(301)의 콘텐츠 x는 기계(302)의 콘텐츠 y에 대하여 승리한다. 이것은 예를 들어, 펜스 값이 기계(301) 상의 자원 A에 대해, 또는 다른 자원 메타-데이터의 비교를 통해 더 높은 경우 발생할 수 있다.
- [0061] 도 4는 본 발명의 다양한 양태에 따라 도 2 및 도 3의 A:x에 대하여 사용될 수 있는 일부 예시적인 자원 데이터 및 메타-데이터를 도시한다. 자원 메타-데이터는 펜스 값, 클럭 값, GUID, 변경을 저술한 복제 구성원, 및 다이제스트를 포함한다. 자원 데이터는 자원의 이름, 데이터 자체, 생성 시각, 수정 시각 및 자원 데이터의 다른 속성을 포함한다.
- [0062] 도 5는 도 3에서 도시되는 동기화와 관련하여, 본 발명의 다양한 양태에 따라 사용될 수 있는 일부 예시적인 자원 데이터 및 메타-데이터를 도시한다. x와 관련된 자원에 대한 펜스 값(즉, 1056603359)이 y와 관련된 자원에 대한 펜스 값(즉, 1)보다 더 크기 때문에, 기계(501) 상의 콘텐츠(예를 들어, x)는 기계(502) 상의 콘텐츠(예를 들어, y)에 대하여 승리하여 기계(502)에 복제된다. 통상적인 복제 방법에서는 더 큰 업데이트 클럭 시각 때문에 y가 x에 대하여 승리한다는 것을 유념한다.
- [0063] 도 6은 본 발명의 다양한 양태에 따라 2개의 기계 사이에서 자원을 동기화하도록 야기할 수 있는 예시적인 단계를 일반적으로 나타낸 데이터 흐름도이다. 프로세스는 블록(605)에서 시작한다.
- [0064] 블록(610)에서, 적어도 하나의 펜스 값이 자원이 전달되어야 함을 나타내는 지에 대하여 결정한다. 펜스 값 양자가 언펜싱된 상태(예를 들어, 0)로 설정될 수 있다. 이러한 경우에, 자원은 전달되지 않아야 한다. 펜스 값 중 하나가 펜싱된 값(예를 들어, 1이상)으로 설정된 경우, 자원은 2개의 자원을 동기화하는 데 필요하다면 전달되어야 한다.
- [0065] 블록(615)에서, 자원이 전달되어야 하는 경우, 프로세스는 블록(620)으로 분기하고 그렇지 않으면 블록(635)으로 분기한다. 블록(620)에서, 펜스 값이 동일한 지에 대해 결정한다. 동일하다면, 프로세스는 다른 메타-데이터가 사용되어 자원을 전달하는 방법을 결정하는 블록(625)으로 분기한다. 블록(627)에서, 자원 또는 그 일부(예를 들어, 메타-데이터, 펜스 값, 콘텐츠 내의 차이 등)는 블록(625)에서 결정된 기계로부터 전달된다.
- [0066] 블록(620)에서 펜스 값이 동일하지 않다면, 프로세스는 블록(630)으로 분기한다. 블록(630)에서, 자원 또는 그 일부(예를 들어, 메타-데이터, 펜스 값, 콘텐츠 내의 차이 등)는 자원에 대해 더 높은 펜스 값을 갖는 기계로부터 자원에 대해 더 낮은 펜스 값을 갖는 기계로 전달된다. 블록(635)에서, 프로세스는 종료한다.
- [0067] 다음은 본 발명의 양태가 사용될 수 있는 일부 예시적인 시나리오이다.
- [0068] 무권한 백업 복구 : 데이터가 손상되거나 손실될 때, 다수의 복제된 자원 시스템을 관리하는 사용자는 구성원을 소거하고, 자원 복제 시스템을 통하여 그 구성원과 관련된 모든 데이터를 획득하는 것을 요구할 수 있다. 구성원이 자신을 자원 복제 시스템으로 접속시키는 링크의 대역폭과 관련하여 상대적으로 클 때, 이 실행의 과정은 시간이 많이 걸리거나 비용이 많이 들 수 있다. 그러나, 본 발명의 양태를 사용하여, 사용자는 백업으로부터 구성원의 자원을 복구할 수 있다. 그리고 사용자는 백업 내의 자원을 언펜싱하고(예를 들어, 펜스 값을 0으로 설정), 자원 복제 시스템이 구식(out of date) 자원을 업데이트하는 것을 허용할 수 있다. 이 경우에, 적합한 논리가 주어진 "콘텐츠 캐쉬(dontent cache)"로서 실행하는 복구된 콘텐츠는 복제기에 의해 사용되어, 느린 링크로(예를 들어, 유선으로) 콘텐츠를 전송하여 초기 동기화 트래픽이 업데이트에 필요한 메타-데이터 및 자원 데이터로 제한된 백업 복구를 계속해서 따르는 것을 피할 수 있다. 동기화 후에, 펜스 값 0을 갖는 임의의 남아 있는 자원은 삭제되거나, 그 펜스 값을 1로 설정시켜 복제하는 것을 허용할 수 있다.
- [0069] 대역 외 복사(copy) : 복제 집합의 새로운 구성원은 잠재적으로 많은 양의 데이터를 포함할 수 있다. 더 빠르고/빠르거나 비용이 덜 드는 전송을 용이하게 하기 위하여, 새로운 구성원은 사용자에게 있어서 더 낮은 비용 및/또는 더 빠른 서비스를 갖는 채널을 통해 송신될 수 있다. 예를 들어, 새로운 구성원은 하드 디스크에 복사되거나 CD-ROM 또는 DVD-ROM 상에 구워져(burn), 밤새 수송될 수 있다. 새로운 구성원을 먼 장소에 있는 시스

템에 복사할 때, 그 구성원은 무권한 백업 복구에 대하여 상술된 바와 같이 언펜싱될 수 있다. 새로운 구성원을 언펜싱하는 것은 먼 장소에 복사된 콘텐츠를 다른 장소로 전송하는 것을 피하기 위하여 행해진다.

- [0070] 권한 백업 복구 : 사용자는 백업으로부터 복구하여 복구된 콘텐츠를 복제 자원 시스템 내의 모든 시스템에 전달하기를 원할 수 있다. 그렇게 하기 위하여, 사용자는 자원을 시스템으로 복구하여 사용자가 전달하기 원하는 자원을 펜싱할 수 있다.
- [0071] 특별(Ad hoc) 시나리오 : 사용자는 특정 버전의 자원 집합이 복제하여 복제 집합 내의 임의의 다른 충돌하는 콘텐츠보다 우선 순위를 갖는 것을 강요하기를 원할 수 있다. 이것은 자원 집합을 펜싱함으로써 행해질 수 있다.
- [0072] 초기 동기화 : 초기 동기화는 대역 외 복사의 특수한 경우이다. 복제된 시스템을 설정 또는 업데이트할 때, 사용자는 하나의 기계를 마스터(즉, 다른 기계에 복제되는 콘텐츠를 포함함)로 지정할 수 있고, 다른 기계를 슬레이브(즉, 마스터로부터 콘텐츠를 수신함)로 지정할 수 있다. 그렇게 하기 위하여, 사용자는 초기 펜스 값 1을 갖는 마스터 상의 자원을 펜싱하고, 펜스 값 0을 갖는 슬레이브 상의 기존 자원을 언펜싱할 수 있다. 이 초기화 후에 슬레이브 상에 생기는 자원은 복제되도록 1로 펜싱될 수 있다. 언펜싱된 슬레이브 자원은 복제되지 않기 때문에, 그러한 자원은 마스터로부터의 이름-충돌하는 자원에 대하여 패배할 때 추가적인 메타-데이터를 동기화할 필요는 없다.
- [0073] 도 7은 본 발명의 다양한 양태에 따라 무권한 백업 복구를 수행하도록 야기할 수 있는 예시적인 단계를 일반적으로 나타낸 데이터 흐름도이다. 구성원 내의 데이터가 삭제되거나 손상된다(블록 (710)). 관리자는 백업으로부터 복구한다(블록 (715)). 관리자는 복구된 데이터를 언펜싱된 것으로 표시한다(블록 (720)). 블록(725)에서 나타난 바와 같이, 동기화가 발생한다. 관리자는 남아있는 자원을 삭제하거나 펜스 값 1로 표시하고(블록 (730)), 프로세스는 종료한다(블록 (735)).
- [0074] 도 7은 상술된 무권한 백업에서 발생할 수 있는 단계에 대한 일례로서 제공된다. 전송된 다른 적용도 충분히 상세하게 설명되어서 데이터 흐름도로 용이하게 정리하였다는 것을 인식할 것이다. 도 7의 단계 또는 상술된 적용에 대하여 본 발명의 취지 또는 범주를 벗어나지 않고 다수의 다른 변화가 생길 수 있다는 것도 이해할 것이다.
- [0075] 도 8은 본 발명의 다양한 양태에 따라 자원 복제 시스템에서 작동하도록 구성된 기계를 나타내는 블록도이다. 기계(805)는 업데이트 메커니즘(810), 자원(822) 및 통신 메커니즘(840)을 포함한다.
- [0076] 업데이트 메커니즘(810)은 펜스 값을 비교하여 자원이 기계(805)로부터 전달되어야 하는 지, 또는 자원이 다른 기계에도 보여져야 하는 지를 결정하는 데 사용되는 펜스 비교기 논리(815)를 포함한다. 펜스 비교기 논리(815)는 펜싱 수행, 또는 자원 메타-데이터(803)의 손상 또는 삭제, 및 후속하는 재구축(이하에 상세히 설명됨) 시, 펜스 값이 어떻게 업데이트되어야 하는 지를 결정하는 데도 사용될 수 있다.
- [0077] 기타 메타-데이터 갈등 해결 논리(820)는 자원에 대한 펜스 값이 동일한 경우, 어느 자원이 승리하는 지(및 전달되는 지)를 결정하는 데 사용되는 논리를 포함한다. 기타 메타-데이터 충돌 해결 논리(820)를 적용할 때, 업데이트 메커니즘(810)은 자원 메타-데이터(830) 및/또는 자원 데이터(825) 내의 데이터를 액세스하여, 자원(또는 그 일부)이 또다른 기계에 전달되어야 하는 지 또는 그로부터 수신되어야 하는 지를 결정할 수 있다.
- [0078] 자원(822)은 자원 데이터(825) 및 자원 메타-데이터(830)를 포함한다. 자원 데이터와 자원 메타-데이터 사이의 구분은 도 2를 참조하여 상술되었다. 동일한 박스(box) 내에서 도시되었지만, 자원 데이터(825)는 자원 메타-데이터(830)와 관련된 개별 저장소에 저장될 수 있다.
- [0079] 통신 메커니즘(840)은 업데이트 메커니즘(810)이 다른 기계 상의 다른 업데이트 메커니즘(도시되지 않음)과 통신하는 것을 허용한다. 또한, 업데이트 메커니즘은 어느 자원이 동기화되어야 하고, 동기화가 어떻게 발생할 수 있는 지를 결정한다. 통신 메커니즘(840)은 네트워크 인터페이스 또는 어댑터(170), 모뎀(172), 또는 도 1과 관련하여 설명된 통신을 설립하기 위한 임의의 다른 수단일 수 있다.
- [0080] 도 8에 도시된 메커니즘의 다른 변화가 본 발명의 취지 및 범주를 벗어나지 않고 구현될 수 있다는 것을 인식할 것이다.
- [0081] 발명의 일 실시예에서, 3개의 초기 펜스 값, -1, 0 및 1이 있다. -1은 상기의 0의 역할을 하고(즉, 언펜싱된 값), 0은 1 이상의 펜스 값을 갖는 다른 자원이 없는 한 자원이 복제할 수 있음을 나타낸다. 이 확장된 유용성은 복제 메타-데이터의 손실을 처리하는 데에 있다. 예를 들어, 자원 복제기는 메타-데이터를 자원 데이터를 저장하는 데 사용된 저장소와 구별된 저장소에 유지할 수 있다. 따라서, 메타-데이터 및 자원 데이터(즉, 콘텐츠

츠)는 독립적으로 실패할 수 있다. 재구축동안, 자신의 메타-데이터를 손실한 기계는 자원이 이미 복제되었는지 또는 로컬로 생성되었는지를 모를 때(따라서, 복제 집합 내의 다른 기계에 알려지지 않음), 초기에 -1 값으로 자신의 자원을 펜싱할 수 있다. 펜스 값을 -1에서 1로 너무 이르게 변경하는 것은 오래된 콘텐츠를 훨씬 최신의 콘텐츠로서 네트워크에 다시 재도입하는 데 영향을 줄 수 있지만, 접속해제된 기계는 덮어쓰여질 수 있다. 한편, 펜스를 -1에서 0으로 변경하는 것은 일부 다른 참가자가 그것이 오래된 것이라고 결정할 수 있을 때까지 그러한 콘텐츠가 재도입되는 것을 허용할 수 있다. 상승된 각각의 값을 시프팅(shift)하는 양의 정수만 요구된다면, 0, 1 및 2(또는 일부 다른 번호 지정 스킴)가 대신에 사용될 수 있음을 유념한다.

[0082] 그러나, 본 발명의 또다른 실시예는 가능한 초기 펜스 값으로서 $-\infty$ 및 0을 사용한다. 기능에 있어서, $-\infty$ 의 펜스 값은 0(즉, 언펜싱된 값)을 대신하는 한편, 0은 1(즉, 자원이 복제되어야 함을 나타냄)을 대신한다. 또한, 다른 음의 펜스 값(예를 들어, -1, -2, -3, ..., -n)은 복제될 수 있는 자원을 나타내는 데 사용될 수 있다. 그러한 펜스 값은 더 높은 값으로 펜싱된 자원에 대하여 패배한다. 이 일반성의 유용성은 복구동안 최신 자원의 가용성이 대역폭 사용보다 더 중요하다는 시나리오를 포함한다. 기계는 각각 초기 자원에서 사용되는 상이한 펜스 값(예를 들어, -1, -2, ..., -n)을 할당받을 수 있다. 이 순서화(ordring)는 펜스 -1을 사용하는 리더 기계에게 우선 순위를 부여하지만, 모든 기계로부터 충돌하지 않는 콘텐츠는 즉시 사용가능하다.

[0083] 이 표현의 또다른 적용은 기계가 메타-데이터의 손상 또는 손실에 응답하여 자신의 메타-데이터를 회복하거나 재구축할 때마다 펜스 값을 감소시키는 것을 포함한다. 이 적용의 발견적(heuristic) 값은 기계가 자신의 메타-데이터를 재구축할 때마다 기계 상에 저장된 자원 데이터가 점점 덜 믿을 만하거나/만하고 덜 최신의 것이 되는 값이다. 이 메커니즘을 통하여, 자신의 메타-데이터가 덜 빈번하게 재구축되는 기계 상의 콘텐츠는 더 믿을 만한 것으로 간주되어, 자신의 메타-데이터를 더 빈번하게 재구축하는 기계 상의 콘텐츠에 대한 펜스 값을 통해 승리한다.

[0084] 상승된 펜스 값은 임의의 다양한 물리적인 펜스 값에서 구현될 수 있는 논리적인 펜스 값을 인식할 것이다. 예를 들어, 펜스 값은 물리적으로 본 발명의 취지 또는 범주로부터 벗어나지 않고 부호 없는 정수, 부동 소수점 점수, 비트 값, 또는 임의의 다른 유형의 번호 지정 스킴으로 나타날 수 있다.

[0085] 진술된 상세한 설명으로부터 보여질 수 있는 바와 같이, 정보를 복제하는 것에 대한 입상 제어를 위한 개선된 방법 및 시스템이 제공된다. 본 발명이 다양한 수정 및 대안적인 구성의 여지가 있는 한편, 그 특정 설명된 실시예가 도면에서 도시되며, 상세하게 진술되었다. 그러나, 개시된 특정 형태로 본 발명을 제한하지 않고 반대로, 모든 수정, 대안적인 구성, 및 본 발명의 취지 및 범주 내에 포함되는 등가물을 포함함을 이해해야한다.

발명의 효과

[0086] 본 발명에 따르면, 어느 콘텐츠가 우선 순위를 갖고 복제되는지를 제어하기 위한 유연한 방법 및 시스템이 제공된다.

도면의 간단한 설명

- [0001] 도 1은 본 발명이 구현될 수 있는 컴퓨터 시스템을 나타내는 블록도.
- [0002] 도 2는 본 발명의 다양한 양태에 따라 자원을 복제하는 2개의 기계를 포함하는 자원 복제 시스템을 나타내는 블록도.
- [0003] 도 3은 본 발명의 다양한 양태에 따라 2개의 기계가 양자에 포함된 자원을 조정하는 시스템을 나타내는 블록도.
- [0004] 도 4는 본 발명의 다양한 양태에 따라 도 2 및 도 3의 기계에 대하여 사용될 수 있는 일부 예시적인 자원 데이터 및 메타-데이터를 도시.
- [0005] 도 5는 본 발명의 다양한 양태에 따라 사용될 수 있는 일부 예시적인 자원 데이터 및 메타-데이터를 도시.
- [0006] 도 6은 본 발명의 다양한 양태에 따라 2개의 기계 사이에서 자원을 동기화하도록 발생할 수 있는 예시적인 단계를 일반적으로 나타낸 데이터 흐름도.
- [0007] 도 7은 본 발명의 다양한 양태에 따라 무권한 백업을 수행하도록 야기할 수 있는 예시적인 단계를 일반적으로 나타낸 데이터 흐름도.
- [0008] 도 8은 본 발명의 다양한 양태에 따라 자원 복제 시스템에서 작동하도록 구성된 예시적인 기계를 나타내는 블록

도면5

501
 기계

A:x

메타-데이터:
 fence value = 1056603359
 clock = 4
 GUID = { 0xfeef11f0-0x01fee0-0x10da-0xdd11-0x0123456789ab }
 version = 24
 author = { 0xfeef11f0-0x01fee0-0x10da-0xdd11-0x0123456789ab }
 digest = 0x12324554

자원-데이터:
 name = "X"
 data = "X"
 CreateTime = 21-Jun-03-08:43:21
 UpdateTime = 25-Jun-03-21:44:32
 ...

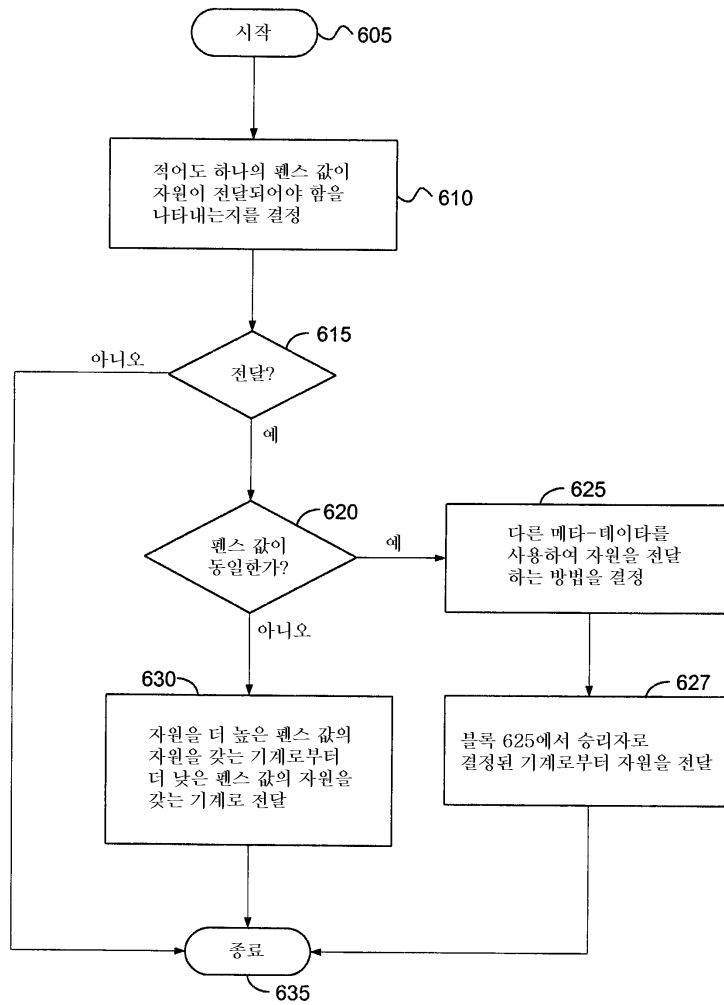
502
 기계

A:y

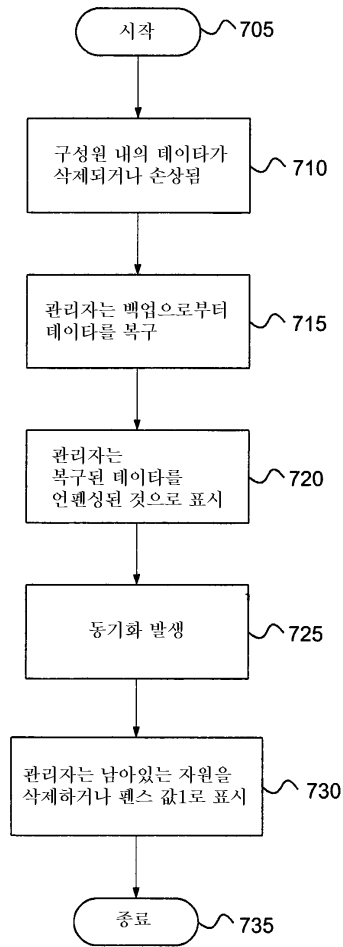
메타-데이터:
 fence value = 1
 clock = 7
 GUID = { 0xfeef11f0-0x01fee0-0x10da-0xdd11-0x0123456789ab }
 version = 21
 author = { 0xfa11afe1-0xeeaa-0x0000-0x0fee0-0xfa11afe1fa11afe1 }
 digest = 0x98127645

자원-데이터:
 name = "Y"
 data = "Y"
 CreateTime = 21-Jun-03-08:43:21
 UpdateTime = 25-Jun-03-21:46:32
 ...

도면6



도면7



도면8

