



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2008-0087791
(43) 공개일자 2008년10월01일

(51) Int. Cl.

G06F 9/44 (2006.01) G06F 17/00 (2006.01)

(21) 출원번호 10-2008-7013644

(22) 출원일자 2008년06월05일

심사청구일자 없음

번역문제출일자 2008년06월05일

(86) 국제출원번호 PCT/US2006/046222

국제출원일자 2006년12월04일

(87) 국제공개번호 WO 2007/067480

국제공개일자 2007년06월14일

(30) 우선권주장

11/364,234 2006년02월28일 미국(US)

60/742,508 2005년12월05일 미국(US)

(71) 출원인

マイクロソフト コーポレーション

미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
マイクロソフト ウェイ

(72) 발명자

수리아나라야난, 구한

미국 98052-6399 워싱턴주 레드몬드 원 마이크로
소프트 웨이

리우, 휘생

미국 98052-6399 워싱턴주 레드몬드 원 마이크로
소프트 웨이

브저너, 니콜라즈, 에스.

미국 98052-6399 워싱턴주 레드몬드 원 마이크로
소프트 웨이

(74) 대리인

양영준, 백만기

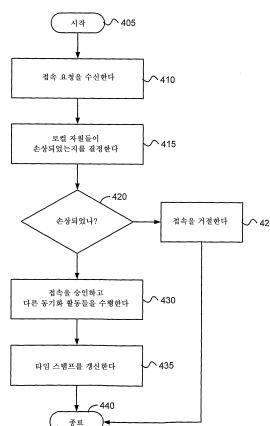
전체 청구항 수 : 총 19 항

(54) 자원 신선도 및 복제를 위한 방법, 장치 및 컴퓨터 실행가능 명령들을 포함하는 컴퓨터 판독 가능 매체

(57) 요약

여기서 설명되는 본 발명의 양태들은 자원 복제 시스템에 관한 것이다. 양태들에서, 로컬 자원들이 손상된 때를 검출하는 메카니즘이 설명된다. 로컬 자원들은 최종 성공 동기화 활동과 현재 시간 사이의 시간이 손상 값을 초과한 때 손상될 수 있다. 로컬 자원들이 손상된 것으로 결정되면, 로컬 멤버는 소정의 동기화 활동들을 재개하는 것이 허가될 때까지 이러한 소정의 동기화 활동들을 중지할 수 있다. 로컬 자원들이 손상되지 않은 경우, 멤버와 업스트림 또는 다운스트림 파트너 간의 자원들의 동기화를 포함하는 추가적인 동기화 활동들이 수행될 수 있다.

대 표 도 - 도4



특허청구의 범위

청구항 1

컴퓨터 실행 가능 명령들을 포함하는 컴퓨터 판독 가능 매체(130, 141, 152, 156, 181)로서,

타임 스탬프를 통해 로컬 자원들이 손상되었는지(stale)를 검출(415)하는 명령-상기 타임 스탬프는 상기 로컬 자원들을 호스트하는 로컬 머신과 상기 제1 머신의 업스트림 파트너 간의 성공적인 동기화 활동이 언제 발생하였는지를 지시하고, 상기 로컬 머신 및 상기 업스트림 파트너는 복제 그룹에 관여함-;

상기 로컬 자원들이 손상된 경우, 상기 로컬 머신이 동기화 활동들을 재개하는 것이 허가될 때까지 소정의 동기화 활동들을 금지(425)하는 명령; 및

상기 로컬 자원들이 손상되지 않은 경우, 상기 소정의 동기화 활동들을 수행(430, 435)하는 명령을 포함하는 컴퓨터 판독 가능 매체.

청구항 2

제1항에 있어서, 상기 타임 스탬프를 통해 상기 로컬 자원들이 손상되었는지를 검출하는 명령은 상기 타임 스탬프와 상기 로컬 머신의 로컬 시스템 클럭의 시간 간의 차이를 결정하는 명령을 포함하는 컴퓨터 판독 가능 매체.

청구항 3

제1항에 있어서, 상기 로컬 자원들이 손상되었는지를 검출하는 명령은 상기 차이가 손상 값(staleness value)보다 큰지를 결정하는 명령을 포함하는 컴퓨터 판독 가능 매체.

청구항 4

제3항에 있어서, 상기 손상 값을 시스템 관리자에 의해 선택되는 컴퓨터 판독 가능 매체.

청구항 5

제3항에 있어서, 상기 손상 값을 자원과 연관된 자원 메타데이터가 삭제되기 전에 상기 자원이 얼마나 오래 삭제된 것으로 표기되었는지를 지시하는 투스톤 기간 이하인 컴퓨터 판독 가능 매체.

청구항 6

제1항에 있어서, 상기 소정의 동기화 활동들은 상기 로컬 자원들을 상기 업스트림 머신 상의 자원들과 동기화하는 활동을 포함하는 컴퓨터 판독 가능 매체.

청구항 7

제1항에 있어서,

상기 로컬 머신 상의 대응 자원과 동일하지 않은 상기 업스트림 머신 상의 임의의 자원을 취득하는 명령;

상기 업스트림 머신 상에서는 포함되지만 상기 로컬 머신 상에는 포함되지 않은 임의의 자원을 추가하는 명령; 및

상기 타임 스탬프를 개신함으로써 상기 로컬 자원들을 신선한 것으로 표기하는 명령을 더 포함하는 컴퓨터 판독 가능 매체.

청구항 8

제7항에 있어서, 상기 업스트림 파트너 상에 없는 상기 로컬 머신 상의 임의의 자원을 삭제하는 명령을 더 포함하는 컴퓨터 판독 가능 매체.

청구항 9

제1항에 있어서, 성공적인 동기화 활동이 상기 업스트림 파트너와 상기 로컬 머신 간의 자원 동기화를 포함하는

컴퓨터 관독 가능 매체.

청구항 10

제1항에 있어서, 성공적인 동기화 활동이 상기 업스트림 멤버과의 접속 유지를 포함하는 컴퓨터 관독 가능 매체.

청구항 11

컴퓨터에 의해 적어도 부분적으로 구현되는 방법으로서,

서로 자원들을 복제하는 멤버들 간의 제1 동기화 활동의 시간에 대응하는 값을 저장하는 단계(435);

제2 동기화 활동에 관여하기 위한 요청을 수신하는 단계(410)-상기 요청은 상기 멤버들 중 제1 멤버에서 수신됨-; 및

상기 제1 멤버의 자원들이 상기 제2 동기화 활동에 관여할 수 있을 만큼 충분히 신선한지를 결정하는 단계(415)를 포함하는 방법.

청구항 12

제11항에 있어서, 상기 값은 타임 스탬프를 포함하는 방법.

청구항 13

제11항에 있어서, 상기 제1 동기화 활동은 상기 제1 멤버와 상기 제1 멤버의 파트너 간의 성공적인 자원 동기화를 포함하는 방법.

청구항 14

제11항에 있어서, 상기 제2 동기화 활동은 상기 제1 멤버와 상기 제1 멤버의 다운스트림 파트너 간의 자원 동기화를 위한 요청을 포함하는 방법.

청구항 15

제11항에 있어서, 상기 제1 멤버의 자원들이 상기 제2 동기화 활동에 관여할 수 있을 만큼 충분히 신선한지를 결정하는 단계는 상기 제1 동기화 활동의 시간과 상기 제2 동기화 활동에 관여하기 위한 요청이 수신된 시간 간의 차이를 계산하고, 상기 차이가 손상 값보다 큰 경우에 상기 제1 멤버의 자원들이 상기 제2 동기화 활동에 관여할 수 있을 만큼 충분히 신선하지 않은 것으로 결정하는 단계를 포함하는 방법.

청구항 16

제11항에 있어서, 상기 요청과 연관된 동기화 세션을 거절하고, 상기 제1 멤버의 자원들이 동기화에 관여할 수 있을 만큼 충분히 신선하지 않음을 지시하는 값을 반환하는 단계를 더 포함하는 방법.

청구항 17

컴퓨팅 환경에서,

복제 그룹에 관여하도록 배열된 멤버들에 의해 복제되는 자원들에 대한 자원 메타데이터(330)를 저장하고, 또한 성공적인 동기화 활동이 발생한 시간에 대응하는 값을 저장하는 자원 저장소(332); 및

상기 자원 저장소의 자원들이 제2 동기화 활동에 관여할 수 있을 만큼 충분히 신선한지를 결정하는 갱신 메카니즘(225)

을 포함하는 장치.

청구항 18

제17항에 있어서, 상기 갱신 메카니즘은 상기 자원 저장소의 자원들이 상기 다른 동기화 활동에 관여할 수 있을 만큼 충분히 신선한 조건들을 지시하는 에포크 로직(epoch logic)을 포함하는 장치.

청구항 19

제18항에 있어서, 상기 자원들은 성공적인 동기화 활동이 발생한 시간이 상기 다른 동기화 활동이 요청된 시간의 손상 값 내에 있을 경우에 충분히 신선한 장치.

명세서

기술 분야

<1> 본 발명은 자원 신선도 및 복제에 관한 것이다.

배경 기술

<2> 자원 복제 시스템들은 대형 네트워크들에서 가용성 및 장애 허용 한계를 확보하기 위해 점점 더 중요해지고 있다. 도메인 자격 증명서들 및 정책들을 포함하는 파일들을 복제하는 범위 네트워크들은 가용성, 확장성 (scalability), 일관성, 및 신뢰성이 요구되는 하나의 예이다. 그러나 다양한 복제 거동을 허가하는 유연한 자원 복제 시스템의 생성은 어렵다.

<3> <발명의 요약>

<4> 요컨대, 여기에 설명되는 본 발명의 양태들은 자원 복제 시스템에 관한 것이다. 양태들에서, 로컬 자원들이 손상된 때를 검출하는 메카니즘이 설명된다. 로컬 자원들은 최종 성공 동기화 활동과 현재 시간 사이의 시간이 손상 값을 초과한 때 손상될 수 있다. 로컬 자원들이 손상된 것으로 결정되면, 로컬 멤버는 소정의 동기화 활동들을 재개하는 것이 허가될 때까지 이러한 소정의 동기화 활동들을 중지할 수 있다. 로컬 자원들이 손상되지 않은 경우, 멤버와 업스트림 또는 다운스트림 파트너 간의 자원들의 동기화를 포함하는 추가적인 동기화 활동들이 수행될 수 있다.

<5> 본 요약은 아래의 상세한 설명에서 더 설명되는 본 발명의 소정 양태들을 간단히 식별하기 위해 제공된다. 본 요약은 청구 발명의 중요한 또는 필수적인 특징들을 식별하고자 하는 의도도 없고, 청구 발명의 범위를 한정하는 데 사용하고자 하는 의도도 없다.

<6> "여기에 설명되는 본 발명"이라는 문구는 문맥이 명확히 다르게 지시하지 않는 한은 상세한 설명에서 설명되는 발명을 지칭한다. "양태들"이라는 용어는 "하나 이상의 양태"로서 해석되어야 한다. 상세한 설명에 설명되는 본 발명의 양태들을 식별하는 것은 청구 발명의 중요한 또는 필수적인 특징들을 식별하는 것을 의도하지 않는다.

<7> 전술한 양태들 및 다른 양태들은 도면들과 관련하여 파악될 때 아래의 상세한 설명으로부터 명확해질 것이다.

발명의 상세한 설명

<14> 예시적인 운영 환경

<15> 도 1은 여기에 설명되는 본 발명의 양태들이 구현되기에 적합한 컴퓨팅 시스템 환경(100)의 일례를 도시하고 있다. 컴퓨팅 시스템 환경(100)은 적합한 컴퓨팅 환경의 일례에 불과하며, 여기에 설명되는 본 발명의 양태들의 용도 또는 기능성의 범위에 관해 어떤 제한을 암시하고자 하는 것이 아니다. 컴퓨팅 환경(100)이 예시적인 운영 환경(100)에 도시된 컴포넌트들 중 임의의 하나 또는 그 컴포넌트들의 임의의 조합과 관련하여 어떤 의존성 또는 요구사항을 갖는 것으로 해석되어서는 안 된다.

<16> 여기에 설명되는 본 발명의 양태들은 많은 기타 범용 또는 특수 목적의 컴퓨팅 시스템 환경 또는 구성에서 동작 할 수 있다. 여기에 설명되는 본 발명의 양태들에서 사용하는 데 적합할 수 있는 잘 알려진 컴퓨팅 시스템, 환경 및/또는 구성의 예로는 퍼스널 컴퓨터, 서버 컴퓨터, 핸드-헬드 또는 랙톱 장치, 멀티프로세서 시스템, 마이크로컨트롤러 기반 시스템, 셋톱 박스, 프로그램 가능한 가전제품, 네트워크 PC, 미니컴퓨터, 메인프레임 컴퓨터, 상기 시스템들이나 장치들 중 임의의 것을 포함하는 분산 컴퓨팅 환경, 기타 등등이 있지만 이에 제한되는 것은 아니다.

<17> 여기에 설명되는 본 발명의 양태들은 일반적으로 컴퓨터에 의해 실행되는 프로그램 모듈과 같은 컴퓨터 실행 가능 명령어와 관련하여 기술될 수 있다. 일반적으로, 프로그램 모듈은 특정 태스크를 수행하거나 특정 추상 데 이터 유형을 구현하는 루틴, 프로그램, 개체, 컴포넌트, 데이터 구조 등을 포함한다. 여기에 설명되는 본 발명

의 양태들은 또한 통신 네트워크를 통해 연결되어 있는 원격 처리 장치들에 의해 태스크가 수행되는 분산 컴퓨팅 환경에서 실시될 수 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈은 메모리 저장 장치를 비롯한 로컬 및 원격 컴퓨터 저장 매체 둘 다에 위치할 수 있다.

<18> 도 1과 관련하여, 여기에 설명되는 본 발명의 양태들을 구현하는 예시적인 시스템은 컴퓨터(110) 형태의 범용 컴퓨팅 장치를 포함한다. 컴퓨터(110)의 컴퓨트론트들은 처리 장치(120), 시스템 메모리(130), 및 시스템 메모리 를 비롯한 각종 시스템 컴퓨트론트들을 처리 장치(120)에 연결시키는 시스템 버스(121)를 포함하지만 이에 제한되는 것은 아니다. 시스템 버스(121)는 메모리 버스 또는 메모리 컨트롤러, 주변 장치 버스 및 각종 버스 아키텍처 중 임의의 것을 이용하는 로컬 버스를 비롯한 몇몇 유형의 버스 구조 중 어느 것이라도 될 수 있다. 예로서, 이러한 아키텍처는 ISA(industry standard architecture) 버스, MCA(micro channel architecture) 버스, EISA(Enhanced ISA) 버스, VESA(video electronics standard association) 로컬 버스, 그리고 메자닌 버스(mezzanine bus)로도 알려진 PCI(peripheral component interconnect) 버스 등을 포함하지만 이에 제한되는 것은 아니다.

<19> 컴퓨터(110)는 통상적으로 각종 컴퓨터 판독가능 매체를 포함한다. 컴퓨터(110)에 의해 액세스 가능한 매체는 그 어떤 것이든지 컴퓨터 판독가능 매체가 될 수 있고, 이러한 컴퓨터 판독가능 매체는 휘발성 및 비휘발성 매체, 이동식 및 비이동식 매체를 포함한다. 예로서, 컴퓨터 판독가능 매체는 컴퓨터 저장 매체 및 통신 매체를 포함하지만 이에 제한되는 것은 아니다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보를 저장하는 임의의 방법 또는 기술로 구현되는 휘발성 및 비휘발성, 이동식 및 비이동식 매체를 포함한다. 컴퓨터 저장 매체는 RAM, ROM, EEPROM, 플래시 메모리 또는 기타 메모리 기술, CD-ROM, DVD(digital versatile disk) 또는 기타 광 디스크 저장 장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 장치 또는 기타 자기 저장 장치, 또는 컴퓨터(110)에 의해 액세스되고 원하는 정보를 저장할 수 있는 임의의 기타 매체를 포함하지만 이에 제한되는 것은 아니다. 통신 매체는 통상적으로 반송파(carrier wave) 또는 기타 전송 메커니즘(transport mechanism)과 같은 피변조 데이터 신호(modulated data signal)에 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터 등을 구현하고 모든 정보 전달 매체를 포함한다. "피변조 데이터 신호"라는 용어는, 신호 내에 정보를 인코딩하도록 그 신호의 특성을 중 하나 이상을 설정 또는 변경시킨 신호를 의미한다. 예로서, 통신 매체는 유선 네트워크 또는 직접 배선 접속(direct-wired connection)과 같은 유선 매체, 그리고 음향, RF, 적외선, 기타 무선 매체와 같은 무선 매체를 포함한다. 상술된 매체들의 모든 조합이 또한 컴퓨터 판독가능 매체의 영역 안에 포함되는 것으로 한다.

<20> 시스템 메모리(130)는 판독 전용 메모리(ROM)(131) 및 랜덤 액세스 메모리(RAM)(132)와 같은 휘발성 및/또는 비휘발성 메모리 형태의 컴퓨터 저장 매체를 포함한다. 시동 중과 같은 때에, 컴퓨터(110) 내의 구성요소들 사이의 정보 전송을 돋는 기본 루틴을 포함하는 기본 입/출력 시스템(BIOS)(133)은 통상적으로 ROM(131)에 저장되어 있다. RAM(132)은 통상적으로 처리 장치(120)가 즉시 액세스 할 수 있고 및/또는 현재 동작시키고 있는 데이터 및/또는 프로그램 모듈을 포함한다. 예로서, 도 1은 운영 체제(134), 애플리케이션 프로그램(135), 기타 프로그램 모듈(136) 및 프로그램 데이터(137)를 도시하고 있지만 이에 제한되는 것은 아니다.

<21> 컴퓨터(110)는 또한 기타 이동식/비이동식, 휘발성/비휘발성 컴퓨터 저장매체를 포함할 수 있다. 단지 예로서, 도 1은 비이동식 · 비휘발성 자기 매체에 기록을 하거나 그로부터 판독을 하는 하드 디스크 드라이브(141), 이동식 · 비휘발성 자기 디스크(152)에 기록을 하거나 그로부터 판독을 하는 자기 디스크 드라이브(151), CD-ROM 또는 기타 광 매체 등의 이동식 · 비휘발성 광 디스크(156)에 기록을 하거나 그로부터 판독을 하는 광 디스크 드라이브(155)를 포함한다. 예시적인 운영 환경에서 사용될 수 있는 기타 이동식/비이동식, 휘발성/비휘발성 컴퓨터 기억 매체로는 자기 테이프 카세트, 플래시 메모리 카드, DVD, 디지털 비디오 테이프, 반도체(solid state) RAM, 반도체 ROM 등이 있지만 이에 제한되는 것은 아니다. 하드 디스크 드라이브(141)는 통상적으로 인터페이스(140)와 같은 비이동식 메모리 인터페이스를 통해 시스템 버스(121)에 접속되고, 자기 디스크 드라이브(151) 및 광 디스크 드라이브(155)는 통상적으로 인터페이스(150)와 같은 이동식 메모리 인터페이스에 의해 시스템 버스(121)에 접속된다.

<22> 위에서 설명되고 도 1에 도시된 드라이브들 및 이들과 관련된 컴퓨터 저장 매체는, 컴퓨터(110)를 위해, 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 및 기타 데이터를 저장한다. 도 1에서, 예를 들어, 하드 디스크 드라이브(141)는 운영 체제(144), 애플리케이션 프로그램(145), 기타 프로그램 모듈(146), 및 프로그램 데이터(147)를 저장하는 것으로 도시되어 있다. 여기서 주의할 점은 이들 컴퓨트론트가 운영 체제(134), 애플리케이션 프로그램(135), 기타 프로그램 모듈(136), 및 프로그램 데이터(137)와 동일하거나 그와 다를 수 있다는 것이다. 이에 관해, 운영 체제(144), 애플리케이션 프로그램(145), 기타 프로그램 모듈(146) 및 프로그램 데이터(147)에

다른 번호가 부여되어 있다는 것은 적어도 이들이 다른 사본(copy)이라는 것을 나타내기 위한 것이다. 사용자는 키보드(162), 및 마우스, 트랙볼(trackball) 또는 터치 패드와 같은 포인팅 장치(161) 등의 입력 장치를 통해 명령 및 정보를 컴퓨터(110)에 입력할 수 있다. 다른 입력 장치(도시 생략)로는 마이크, 조이스틱, 게임 패드, 위성 안테나, 스캐너, 핸드헬드 PC 또는 기타 기록 태블릿의 터치 감지 스크린 등을 포함할 수 있다. 이들 및 기타 입력 장치는 종종 시스템 버스에 결합된 사용자 입력 인터페이스(160)를 통해 처리 장치(120)에 접속되지만, 병렬 포트, 게임 포트 또는 USB(universal serial bus) 등의 다른 인터페이스 및 버스 구조에 의해 접속될 수도 있다. 모니터(191) 또는 다른 유형의 디스플레이 장치도 비디오 인터페이스(190) 등의 인터페이스를 통해 시스템 버스(121)에 접속될 수 있다. 모니터 외에, 컴퓨터는 스피커(197) 및 프린터(196) 등의 기타 주변 출력 장치를 포함할 수 있고, 이들은 출력 주변장치 인터페이스(195)를 통해 접속될 수 있다.

<23> 컴퓨터(110)는 원격 컴퓨터(180)와 같은 하나 이상의 원격 컴퓨터로의 논리적 접속을 사용하여 네트워크화된 환경에서 동작할 수 있다. 원격 컴퓨터(180)는 퍼스널 컴퓨터, 핸드-헬드 장치, 서버, 라우터, 네트워크 PC, 피어 장치 또는 기타 통상의 네트워크 노드일 수 있고, 도 1에는 메모리 저장 장치(181)만이 도시되어 있지만, 통상적으로 컴퓨터(110)와 관련하여 상술된 구성요소들의 대부분 또는 그 전부를 포함한다. 도 1에 도시된 논리적 접속으로는 LAN(171) 및 WAN(173)이 있지만, 기타 네트워크를 포함할 수도 있다. 이러한 네트워킹 환경은 사무실, 전사적 컴퓨터 네트워크(enterprise-wide computer network), 인트라넷, 및 인터넷에서 일반적인 것이다.

<24> LAN 네트워킹 환경에서 사용될 때, 컴퓨터(110)는 네트워크 인터페이스 또는 어댑터(170)를 통해 LAN(171)에 접속된다. WAN 네트워킹 환경에서 사용될 때, 컴퓨터(110)는 통상적으로 인터넷과 같은 WAN(173)을 통해 통신을 설정하기 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 접속된다. 네트워크화된 환경에서, 컴퓨터(110) 또는 그의 일부와 관련하여 기술된 프로그램 모듈은 원격 메모리 저장 장치에 저장될 수 있다. 예로서, 도 1은 원격 애플리케이션 프로그램(185)이 원격 컴퓨터(180)에 있는 것으로 도시하고 있지만 이에 제한되는 것은 아니다. 도시된 네트워크 접속은 예시적인 것이며 이 컴퓨터들 사이에 통신 링크를 설정하는 기타 수단이 사용될 수 있다는 것을 이해할 것이다.

<25> 자원 복제

<26> 쉽게 이해되듯이, 현대의 머신들은 비교적 짧은 기간에 수천 개의 자원 변경들을 처리할 수 있다. 이러한 자원들을 복제하고, 다양한 신뢰성 및 대역폭의 다양한 네트워크를 통해 접속된 수백 또는 수천 개의 머신들을 통해 이들을 동기 상태로 유지하는 것은 상당한 어렵다.

<27> 낙관적인 멀티-마스터 복제 시스템들은 주어진 복제 그룹에 관여하는 임의의 머신 상에서 복제된 콘텐츠에 대한 제한 없는 변경들을 허가한다. 복제 그룹은 복제 그룹에 관여하는 머신들 상에서 복제되는 한 세트의 자원들을 포함한다. 복제 그룹의 자원들의 세트는 볼룹들을 스팯닝(spanning)할 수 있다. 예를 들어, 복제 그룹은 복제 그룹에 관여하는 한 세트의 머신들 상에서 복제될 수 있는 C:\#DATA, D:\#APPS, 및 E:\#DOCS와 연관된 자원들을 포함할 수 있다. 잠재적으로 충돌하는 변경들은 모든 충돌 상황에 대해 어느 충돌하는 변경이 다른 것들보다 우선 순위를 갖는지를 정의하는 한 세트의 충돌 해결 기준들을 이용하여 복제 시스템의 제어하에 조정된다.

<28> 복제 그룹 멤버 또는 복제 그룹의 멤버는 복제 그룹 상에 포함된 자원들을 복제함으로써 복제 그룹에 관여하는 머신을 포함한다. 콘텐츠 세트(때때로 간단히 "콘텐츠"로 지칭됨)는 복제 그룹 멤버 상에 상주하는 자원들을 포함한다.

<29> "머신"이라는 용어는 단순히 물리적 머신으로 한정되지 않는다. 오히려, 단일 물리적 머신은 다수의 가상 머신을 포함할 수 있다. 여기서 사용되는 바와 같이, 하나의 머신에서 다른 머신으로의 복제는 가상 또는 물리적인 하나의 머신에서 가상 또는 물리적인 다른 머신으로의 동일 복제 그룹의 하나 이상의 멤버의 복제를 의미한다. 단일 물리적 머신은 동일 복제 그룹의 다수의 멤버를 포함할 수 있다. 따라서, 복제 그룹의 멤버들의 복제는 동일 복제 그룹의 둘 이상의 멤버를 포함하는 단일 물리적 머신의 멤버들의 동기화를 수반할 수 있다.

<30> 자원은 객체로서 간주될 수 있다. 각각의 자원은 자원 데이터 및 자원 메타데이터와 연관된다. 자원 데이터는 콘텐츠 및 콘텐츠와 연관된 속성들을 포함할 수 있고, 자원 메타데이터는 동기화 협상 및 충돌 해결에 관련될 수 있는 다른 속성들을 포함한다. 자원 데이터 및 메타데이터는 데이터베이스 또는 다른 적절한 저장소에 저장될 수 있고, 다른 실시예에서는 자원 데이터 및 메타데이터를 저장하기 위해 개별 저장소들이 사용될 수 있다.

<31> 파일 시스템 내의 명명된 파일들에 기초하는 데이터 저장소들을 포함하는 복제 시스템들에서, 자원 데이터는 파일

일 콘텐츠는 물론, 파일 콘텐츠와 관련하여 파일 시스템 상에 저장되는 임의의 파일 속성들을 포함할 수 있다. 파일 속성들은 액세스 제어 리스트(ACL), 생성/수정 시간, 및 파일과 연관된 기타 데이터를 포함할 수 있다. 여기서 사용되는 바와 같이, 파일은 디렉토리들(즉, 자손 파일들 및 디렉토리들 및 조상 디렉토리들과 연관될 수 있는 파일 시스템 객체들) 및 비 디렉토리들을 포함할 수 있다. 문맥이 명확히 다르게 지시하는 않는 한, 파일이라는 용어는 "자원 데이터"로서 해석될 수 있다.

<32> 파일 시스템 내의 명명된 파일들에 기초하지 않는 데이터 저장소들을 포함하는 복제 시스템들(예를 들어, 자원들이 데이터베이스 또는 객체 기반 데이터 저장소에 저장되는 시스템들)에서, 데이터 저장소에 적합한 자원 데이터가 저장된다. 본 명세서 전반에서, 파일 시스템 내의 파일들에 기초하는 복제 시스템들이 때때로 예시를 위해 사용되지만, 여기에 설명되는 본 발명의 사상 또는 범위를 벗어나지 않고, 콘텐츠를 저장할 수 있는 임의의 데이터 저장소가 사용될 수 있다는 것을 인식할 것이다.

<33> 각각의 자원에 대해, 자원 메타데이터는 전역 고유 식별자(GUID), 자원이 삭제되었는지의 여부, 변경의 출처와 함께하는 버전 시퀀스 번호, 변경이 발생한 시간을 나타내는 클럭 값, 및 자원 데이터의 값들을 요약하고 자원 콘텐츠에 대한 시그니처들을 포함할 수 있는 다이제스트와 같은 다른 필드들을 포함할 수 있다. 다이제스트는 예를 들어 복제 동기화 동안 데이터 전달을 바이패스하기 위한 빠른 비교를 위해 사용될 수 있다. 목적 머신 상의 자원이 소스 머신 상의 콘텐츠(예를 들어, 다이제스트에 의해 지시됨)와 동기화되는 경우, 자원 데이터 자체를 전송하지 않고 자원 메타데이터만을 전송함으로써 네트워크 오버헤드가 최소화될 수 있다. 자원 메타데이터의 전송은 목적 머신이 그의 후속 복제 활동들에서 소스 머신 상에 포함된 메타데이터를 반영할 수 있도록 수행된다. 이것은 예를 들어 목적 머신이 후속 복제 활동에서 소스 머신이 되는 것을 허가할 수 있다. 자원 메타데이터는 여기에 설명되는 본 발명의 사상 또는 범위를 벗어나지 않고 자원 데이터와 함께 또는 별도로 저장될 수 있다.

<34> 버전 벡터들은 자원들을 복제할 때 사용될 수 있다. 버전 벡터는 복제 그룹에 관여하는 머신들의 카운터들 또는 클럭들의 전역 세트로서 간주할 수 있다. 복제 그룹에 관여하는 각각의 머신은 그의 현재의 최신 버전, 및 그가 다른 머신들에 관하여 수신한 최신 버전들을 나타내는 버전 벡터를 유지한다. 자원이 머신으로부터 생성, 수정 또는 삭제될 때마다, 자원의 버전은 해당 머신에 대한 현재의 버전 번호에 1을 더한 값과 동일한 버전 번호로 설정될 수 있다. 해당 머신에 대한 버전 벡터는 또한, 그 머신에 대한 버전 번호가 증가한 것을 반영하도록 갱신된다.

<35> 버전 벡터들은 지식의 상태에 관한 정보를 유지한다. 버전 벡터는 머신 식별자를 타임 스탬프에 맵핑할 수 있다. 복제 그룹에 관여하는 각각의 머신은 그 자신의 클럭 벡터를 유지하여, 다른 머신들이 도달한, 그가 아는 클럭 값을 기록할 수 있다. 클럭 벡터들은 버전 시퀀스 번호를 포함할 수 있다. 일 실시예에서는, 각각의 머신에 대해 단일 클럭 값을 유지하는 대신에, 갱신들의 무질서한 처리를 조정하고 버전 벡터들에 대한 기본 세트 동작들을 이용하기 위해 클럭 값을 세트가 유지될 수 있다.

<36> 일 실시예에서, 동기화 동안, 버전 벡터가 자원들의 동기화에 사용하기 위해 전송될 수 있다. 예를 들어, 머신들 A 및 B가 결합(join)과 같은 동기화 활동에 관여하는 경우, 머신 B는 그의 버전 벡터를 A로 전송할 수 있다. B의 버전 벡터를 수신한 후, A는, 만일 존재한다면, B의 버전 벡터에 의해 포함되지 않는(즉, 지배되지 않는) 버전들을 갖는 모든 자원에 대한 변경들을 전송할 수 있다. "버전 벡터가 다른 버전 벡터를 지배한다"라는 문구는 버전 벡터들 간의 대응하는 포함 관계와 동의어이다.

<37> 동기화에서 버전 벡터들의 사용의 예는 "Interval Vector Based Knowledge Synchronization for Resource Versioning"이라는 제목의 미국 특허 출원 번호 10/791,041, "Garbage Collection of Tombstones for Optimistic Replication Systems"라는 제목의 미국 특허 출원 번호 10/779,030, 및 "Granular Control Over the Authority of Replicated Information via Fencing and Unfencing"이라는 제목의 미국 특허 출원 번호 10/733,459에 설명되어 있다.

<38> 도 2는 여기에 설명되는 본 발명의 양태들에 따라 복제 그룹에 관여하는 머신들을 일반적으로 나타내는 블록도이다. 일례로, 업스트림 머신(201) 및 다운스트림 머신(202)이 2개의 자원을 포함하는 복제 그룹에 관여할 수 있다. 이러한 2개의 자원은 예를 들어 문서 디렉토리들(205, 215) 및 헬프 디렉토리들(210, 220)(이들은 특정 순간에 이들 자원이 동일한 자원 데이터를 포함하지 않을 수 있음을, 즉 이들이 동기화되지 않을 수 있음을 지시하기 위해 2개의 머신 상에서 상이한 번호를 부여받음)을 포함할 수 있다.

<39> 주기적으로, 다운스트림 머신(202)은 업스트림 머신(201)으로부터 갱신들을 요청할 수 있으며, 갱신들에 기초하

여 그의 자원들을 갱신할 수 있다. 도 2에는 2개의 머신만이 도시되어 있지만, 업스트림 및 다운스트림 머신들(201, 202)은 많은 다른 머신을 포함하는 복제 시스템의 일부일 수 있다. 하나의 상호작용에서 소스인 머신(때때로 업스트림 머신이라고 함)은 나중에 다른 상호작용에서 목적지(때때로 다운스트림 머신이라고 함)가 될 수 있고, 그 역도 가능하다.

- <40> 다운스트림 머신(202)은 업스트림 머신(201)으로부터 언제 갱신들을 취득할 것인지를 결정하는 갱신 메카니즘(225)을 포함할 수 있다. 다운스트림 머신(202)이 다른 머신(도시되지 않음)의 업스트림 파트너이기도 한 경우, 갱신 메카니즘(225)은 또한 어느 갱신들을 다른 머신에 전송할지를 결정할 수 있다. 갱신 메카니즘은 도 3과 관련하여 더 상세히 후술한다.
- <41> 도 3은 여기에 설명되는 본 발명의 양태들에 따라 자원 복제 시스템에서 동작하도록 구성된 머신을 나타내는 블록도이다. 머신(305)은 갱신 메카니즘(225), 자원들(322) 및 통신 메카니즘(340)을 포함한다. 갱신 메카니즘은 에포크(epoch) 로직(345), 프로토콜 로직(315) 및 다른 동기화 로직(320)을 포함한다.
- <42> 자원들(322)은 자원 데이터를 저장하기 위한 자원 데이터 저장소(325) 및 자원 메타데이터 저장소(330)를 포함한다. 동일 박스 내에 도시되지만, 자원 데이터 저장소(325)는 자원 메타데이터 저장소(330)에 대해 함께 또는 별도의 저장소에 저장될 수 있다. 특히, 자원 메타데이터 저장소(330)는 자원 데이터 저장소(325)에 저장된 자원 데이터 레코드들 각각에 대한 버전들을 포함할 수 있으며, 인터벌 벡터(블록 335)도 포함할 수 있다.
- <43> 자원 메타데이터 저장소(330)는 또한 복제된 폴더들과 연관된 레코드들을 포함할 수 있으며, 각각의 레코드는 (예를 들어, 타임 스탬프를 통해) 언제 그의 각각의 복제된 폴더가 다른 멤버 상의 대응하는 복제된 폴더와의 성공적인 동기화 활동에 최종 연관되었는지를 지시한다. 여기서 사용되는 바와 같이, 복제된 폴더는 콘텐츠 세트와 동의어이다.
- <44> 통신 메카니즘(340)은 갱신 메카니즘(225)이 다른 머신들 상의 다른 갱신 메카니즘들(도시되지 않음)과 통신하는 것을 허가한다. 통신 메카니즘(340)은 네트워크 인터페이스 또는 어댑터(170), 모뎀(172), 또는 도 1과 관련하여 설명된 바와 같은 통신을 설정하기 위한 임의의 다른 수단일 수 있다.
- <45> 갱신 메카니즘(225)은 다른 복제 파트너들과 통신하도록 동작하는 프로토콜 로직(315) 및 후술되는 바와 같이 머신이 다른 머신들과의 동기화 활동들에 관여할 수 있는지를 결정하도록 동작하는 에포크 로직(345)을 포함한다. 다른 동기화 로직(320)은 프로토콜 로직(315)과 다른 동기화 로직(예를 들어, 충돌하는 갱신들의 경우에 무엇을 해야할지, 어떠한 갱신들을 얻어야할지를 어떻게 결정할지 등)을 포함한다. 프로토콜 로직(315), 에포크 로직(345) 및 다른 동기화 로직(320)이 개별 박스들로서 도시되지만, 이들은 전체적으로 또는 부분적으로 결합될 수 있다.
- <46> 여기서 사용되는 바와 같이, 복제 서비스는 도 3과 관련하여 전술한 컴포넌트들 중 임의의 하나 이상을 포함할 수 있다.
- <47> 동작에 있어서, 머신(305)은 다운스트림 머신 및/또는 업스트림 머신으로 동작할 수 있다. 다운스트림 머신은 업스트림 및 다운스트림 머신들 양자가 관여하는 복제 그룹에 대해 업스트림 머신과 접속을 설정할 수 있다. 이것은 예를 들어 통신 메카니즘(340)을 통해 수행될 수 있다. 접속 설정시, 파트너들(즉, 업스트림 및 다운스트림 머신들) 각각은 그의 버전 벡터를 다른 파트너에게 전송할 수 있다. 이어서, 업스트림 머신에서 다운스트림 머신으로 갱신들을 전송하기 위해 세션이 설정된다.
- <48> 세션은 업스트림 머신의 복제된 폴더와 그에 대응하는 다운스트림 머신의 복제된 폴더를 결합하는 데 사용될 수 있다. 세션은 복제 그룹의 각각의 복제된 폴더에 대해 설정될 수 있다. 다수의 폴더에 대한 세션들은 업스트림 및 다운스트림 머신들 간의 단일 접속을 통해 설정될 수 있다.
- <49> 세션에서 모든 갱신들이 처리 또는 포기된 후, 다운스트림 머신은 세션을 종료할 수 있다.
- <50> 다운스트림 머신은 (예를 들어, 통신 메카니즘(340)을 통해) 업스트림 머신이 세션과 연관된 임의의 자원들에 대한 갱신들이 발생할 때 다운스트림 머신에게 통지할 것을 요청할 수 있다. 업스트림 머신이 다운스트림 머신에게 갱신들이 이용 가능함을 통지할 때, 다운스트림 머신은 갱신들에 대한 버전 벡터를 요청할 수 있다. 이에 응답하여, 업스트림 머신은 그의 버전 벡터(때때로 "VVup"라고 함)를 전송한다. VVup은 완전한 버전 벡터 또는 최종 버전 벡터가 전송된 이후의 변경들을 포함하는 버전 벡터를 포함할 수 있다는 점에 유의한다. 다운스트림 머신에게 갱신들이 이용 가능함을 통지하고, 다운스트림 머신이 갱신들을 요청하기를 기다리는 것은, 다운스트림 머신이 다수의 업스트림 파트너로부터의 버전 벡터들로 뜻밖에 넘치지 않도록 2 단계로 수행될 수 있다.

- <51> 다운스트림 머신은 그가 수신한 업스트림 버전 벡터(즉, "VVup")를 사용하여, 그 자신의 버전 벡터와의 세트 차이를 계산하여 그가 모르는 업스트림 머신에 상주하는 버전들을 계산한다. 이어서, 다운스트림 머신은 그 버전들에 관한 메타데이터를 요청할 수 있다. 개신들의 요청시, 다운스트림 머신은 그가 어떤 개신들을 필요로 하는지를 지시하는 엘타 버전 벡터를 포함시킬 수 있다.
- <52> 다운스트림 머신은 투스톤들 또는 라이브 개신들을 별개로 또는 함께 요청할 수 있다. 투스톤은 자원이 삭제되었음을 나타내고, 라이브 개신들은 자원을 삭제하지 않은 개신들을 나타낸다. 몇몇 구현에서 있어서, 다운스트림 머신은 라이브 개신들을 요청하기 전에 투스톤들을 요청할 수 있다. 이것은, 수정된 후 삭제된 자원이 복제파트너 상에서 삭제되기 전에 수정될 필요가 없으므로, 효율을 향상시키기 위해 수행될 수 있다. 또한, 라이브 개신 전에 투스톤을 처리하는 것은 라이브 대체 개신을 처리하기 위한 준비로서 다운스트림 머신의 데이터 저장소(예를 들어, 파일 시스템 내)의 명칭 공간을 소거할 수 있다.
- <53> 개신들을 수신한 후, 다운스트림 머신은 개신들의 처리를 시작하여, 개신들과 연관된 어떠한 자원 데이터 또는 그의 일부를 업스트림 머신으로부터 요청할지를 결정할 수 있다. 이러한 처리는 예를 들어 다수의 컴포넌트(도시되지 않음)로 분할될 수 있는 다른 동기화 로직(340)을 통해 수행될 수 있다. 예를 들어, 개신은 특정 자원의 자원 데이터 또는 그 일부가 변경되었음을 지시할 수 있다. 일 실시예에서, 자원과 연관된 모든 자원 데이터가 다운스트림 머신에 의해 요청될 수 있다. 다른 실시예에서, 변경을 포함하는 자원 데이터의 일부가 다운스트림 머신에 의해 요청될 수 있다. 여기서 사용되는 바와 같이, 자원 데이터를 수반하는 상호작용(예를 들어, 요청, 응답, 개신 등)은 자원과 연관된 자원 데이터의 일부 또는 모두를 수반하는 상호작용을 의미하는 것으로 이해되어야 한다. 예를 들어, 자원 데이터에 대한 요청은 자원과 연관된 자원 데이터의 일부 또는 모두에 대한 요청을 의미할 수 있다.
- <54> 요청될 필요가 있는 자원 데이터를 결정한 후, 다운스트림 머신은 자원 데이터를 요청할 수 있다. 자원 데이터에 대한 요청에 응답하여, 업스트림 머신은 자원과 연관된 자원 데이터를 전송할 수 있다. 요청 및 응답은 다운스트림 머신이 개신될 필요가 있는 것으로 결정한 모든 자원 데이터가 요청된 때까지 계속될 수 있다. 예를 들어 자원이 삭제된 경우에 업스트림 머신은 더 이상 요청된 자원 데이터를 갖지 못할 수 있으므로, 모든 자원 데이터가 전송될 수 있는 것은 아님에 유의한다. 자원 데이터가 전송되지 못할 수 있는 다른 예는 다운스트림 머신에 대한 유일한 유효 변경이 자원이 재명명되거나 메타데이터 속성들이 개신되는 것인 경우이다. 이러한 경우에, 개신을 수신하고 로컬 자원을 재명명하거나 로컬 메타데이터를 개신하는 것은 다운스트림 자원과 업스트림 자원을 동기화하는 데 필요한 전부일 수 있다.
- <55> 예를 들어 복제된 폴더가 삭제되거나, 복구 불가능한 에러가 복제 동안 발생하거나, 복제 시스템이 셧다운되는 경우에 세션이 종료될 수 있다. 그렇지 않은 경우, 설정된 세션은 위의 이벤트들의 모두 또는 일부를 수반하는 후속 동기화 액션들에 사용될 수 있다.
- <56> 개신 메카니즘(225)은 자원 데이터 저장소(325) 내의 자원 데이터의 파일 및 디렉토리 상태들의 추적을 유지하기 위해 데이터베이스(예를 들어, 자원 메타데이터 저장소(330))를 이용할 수 있다. 자원이 개신될 때마다, 자원은 데이터베이스에서 새로운 버전을 할당받을 수 있다.
- <57> 개신 메카니즘(225)은 업스트림 파트너로부터 개신들을 수신하고 로컬 자원들을 개신하여, 이들이 업스트림 파트너 상의 복제된 자원들과 동기화되게 할 수 있다. 자원 메타데이터에 대한 개신들은 자원 메타데이터 저장소(330)에 저장될 수 있고, 자원 데이터에 대한 개신들은 자원 데이터 저장소(325)에 저장될 수 있다. 일 실시예에서, 자원 데이터 저장소(325) 및 자원 메타데이터 저장소(330)는 동일 볼륨 또는 물리적 장치(예를 들어, 하드 드라이브) 상에 포함될 수 있다.
- <58> 자원 데이터는 또한 자원 데이터에 관련된 로컬 파일 활동에 의해 변경될 수 있다. 예를 들어, 복제 파트너는 많은 클라이언트 머신(도시되지 않음)에 대한 파일 서버로서 기능할 수 있다. 클라이언트 머신들은 복제 파트너 상의 자원 데이터(예를 들어, 파일들)에 액세스하여 이들을 수정할 수 있다. 개신 메카니즘(225)은 복제된 자원들에 대한 변경들(예를 들어, 파일들에 대한 변경들)을 모니터링할 수 있는 자원 데이터 변경 모니터(도시되지 않음)를 포함할 수 있다. 복제된 자원이 변경될 때, 자원 변경 모니터는 개신을 지시하도록 자원 메타데이터 저장소(330)를 개신할 수 있다. 이것은 자원 메타데이터 저장소(330)를 자원 데이터 저장소(325)와 동기로 유지하고 복제 그룹 멤버들을 동기화하는 기능을 할 수 있다.
- <59> 콘텐츠 신선도
- <60> 에포크 로직(345)은 복제 그룹의 멤버의 콘텐츠 세트의 콘텐츠 신선도를 결정할 때 사용될 수 있다. 멤버가 복

제 그룹의 멤버들을 갖는 다른 머신들로부터 장기간 분리될 때, 멤버의 자원들은 동기화되지 않고 손상될 수 있다. 멤버가 복제 그룹에 다시 연결되고, 여기에 설명되는 에포크 로직(345)을 사용하지 않는 경우, 머신은 복제 그룹의 다른 멤버들과의 복제를 시작할 수 있다. 멤버 상에 갱신이 존재하지 않는 경우, 멤버는 다른 멤버들로부터 갱신들을 수신할 수 있으며, 어떠한 손상된 자원들도 복제하지 않을 수 있다. 그러나, 멤버 상에 도움이 되지 않는 최근의 갱신들이 존재하는 경우, 이러한 갱신들은 복제될 수 있고, 다른 멤버들 상의 양호한 최신 버전들을 덮어쓸 수 있다. 따라서, 어떤 점에서, 콘텐츠 신선도는 변경들의 적절성에 대한 척도(measure)로서 간주될 수 있다.

- <61> 이것이 발생할 수 있는 일례는 머신이 하드웨어 문제로 인해 장기간(예를 들어, 두 달) 셧다운될 때이다. 머신은 수리 후 온라인 복귀할 수 있으며, 머신 상에서 호스트되는 복제 그룹들의 멤버들은 다른 머신들 상의 파트너들과의 동기화를 시작할 수 있다. 파트너들과의 동기화를 시작할 때, 또는 그 바로 전에, 소정의 애플리케이션(예를 들어, 안티바이러스 애플리케이션)이 손상된 파일들 중 하나 이상을 트래버스하고 이들을 터치할 수 있다(예를 들어, 이들의 최종 수정 시간을 갱신할 수 있다). 여기에 설명되는 양태들이 없다면, 이것은 멤버들 상의 손상된 파일들이 최근에 갱신된 것으로 보이고, 다른 머신들 상의 다른 멤버들로 복제되어 다른 머신들 상의 보다 양호한 버전들을 덮어쓰게 할 수 있다.
- <62> 일 실시예에서, 위의 예가 전형적인 손상 콘텐츠의 복제는 에포크들의 사용을 통해 방지되거나 줄여질 수 있다. 일 실시예에서, 하나의 머신이 마스터로서 지정될 수 있고, 에포크 값이 이 머신 상에서 항상 갱신되고 다른 머신들로 전파될 수 있다. 동기화 동안, 2개의 머신 간의 에포크들이 허용되는 것보다 큰 차이를 갖는 경우, 손상된 콘텐츠를 갖는 머신(예를 들어, 보다 작은 에포크를 갖는 머신)으로부터의 갱신들은 더 최신인 머신으로 복제되지 않을 것이다.
- <63> 양태들에서, 에포크 값들은 머신, 복제된 폴더들 또는 개별 자원들 상에 설정될 수 있다. 에포크 값이 머신 상에 설정되는 경우, 머신 상의 복제된 폴더들 및 자원들은 동일 에포크 값을 갖는 것으로 간주할 수 있다. 에포크 값이 복제된 폴더 상에 설정되는 경우, 폴더와 연관된 모든 복제된 자원은 동일 에포크 값을 갖는 것으로 간주될 수 있다. 머신 상의 다른 복제된 폴더는 상이한 에포크 값을 가질 수 있다. 각각의 자원이 그 자신의 에포크 값을 갖는 경우, 각각의 복제된 폴더 내의 상이한 자원들은 상이한 에포크 값들을 가질 수 있다.
- <64> 에포크 값들의 사용은 손상도가 결정될 수 있도록 하기 위해 동기화 동안 에포크 값들을 전송하는 것을 필요로 할 수 있다. 결과적으로, 자원별로 에포크 값들을 구현하는 것은 복제된 폴더들에 대해 또는 머신별로 에포크 값들을 구현하는 것보다 많은 오버헤드를 부과할 수 있다. 여기에 설명되는 본 발명의 사상 또는 범위를 벗어나지 않고, 에포크 값들은 하나의 머신 상에서 하나의 방식으로(예를 들어, 복제된 폴더별로) 그리고 다른 머신 상에서 다른 방식으로(예를 들어, 자원별로) 설정될 수 있다.
- <65> 다른 실시예에서는, 에포크 값들을 사용하는 대신에, 멤버가 로컬 시스템 클럭을 이용하여 생성된 타임 스탬프를 저장할 수 있다. 로컬 시스템 클럭은 최신으로 유지하기 위해 시간 서버로부터 주기적으로 갱신될 수 있다. 타임 스탬프는 또한 다양한 동기화 활동에 따라 갱신될 수 있다. 일 실시예에서, 복제된 폴더에 대한 타임 스탬프는 복제된 폴더가 다른 멤버의 복제된 폴더와 성공적으로 동기화될 때마다 갱신될 수 있다. 다른 실시예에서, 타임 스탬프는 복제 그룹의 다른 멤버와 양호한 접속이 설정될 때마다 갱신될 수 있다.
- <66> 또 다른 실시예에서, 타임 스탬프는 동기화 활동에 따라 하루에 한 번 또는 소정의 다른 선택된 빈도로 갱신될 수 있다. 그 기간 동안 성공적인 동기화 활동이 발생하는 경우, 타임 스탬프가 갱신될 수 있다. 성공적인 동기화 활동은 복제된 폴더를 다른 멤버의 복제된 폴더와 성공적으로 동기화하는 것, 다른 멤버와 양호한 접속을 설정하는 것, 다른 멤버와 양호한 접속을 유지하는 것(예를 들어, "하트 비트" 메시지들에 의해 또는 달리 지시됨), 업스트림 파트너로부터 버전 벡터를 얻는 것, 버전 벡터를 다운스트림 파트너에게 전송하는 것, 또는 임의의 다른 동기화 활동을 포함할 수 있다.
- <67> 타임 스탬프들과 함께, 각각의 복제된 폴더는 그의 메타데이터 저장소에 관련 레코드(예를 들어, CONTENT_SET_RECORD)를 가질 수 있다. 레코드에 포함된 다른 필드들 중에서, 레코드는 예를 들어 레코드의 대응하는 복제된 폴더의 신선도를 반영하기 위해 타임 스탬프를 포함할 수 있다.
- <68> 머신 상에서 호스트되는 하나의 복제된 폴더가 최신인 반면, 그 머신 상에서 호스트되는 다른 복제된 폴더는 손상된 것일 수 있다는 점에 유의한다. 이것은 하나의 복제된 폴더가 그의 복제 그룹 내의 다른 멤버들과 동기화 할 수 있었던 반면, 다른 복제된 폴더는 그의 복제 그룹 내의 다른 멤버들과 동기화할 수 없었기 때문에 발생할 수 있다.

- <69> 복제된 폴더를 호스트하는 멤버가 다른 멤버에 접속하거나 다른 멤버로부터의 접속 요청을 승인하려고 시도하기 전에, 멤버는 임의의 다른 복제 활동에 관여하기 전에 복제된 폴더의 타임 스탬프가 최신인지(예를 들어, 현재 시각으로부터 설정 가능한 날수 내인지)를 검사할 수 있다. 그의 타임 스탬프가 최신인 경우, 멤버는 접속하거나 접속 요청을 허가하고 동기화 활동들을 계속할 수 있다. 그렇지 않은 경우, 멤버는 다른 멤버에 접속하려고 시도하지 않거나, 임의의 접속 요청을 거절할 수 있으며, 다른 멤버와의 임의의 추가적인 복제 활동의 개시를 중지할 수 있다.
- <70> 다른 실시예에서, 멤버의 콘텐츠가 손상된 경우, 멤버는 접속 요청을 거절하는 대신에 접속 요청을 승인하지만, 동기화 세션에 참여하기를 거절할 수 있다. 일 실시예에서, 동기화 세션에 대한 참여 거절은 접속 요청의 거절 대신에 그리고 접속 요청의 거절과 관련하여 여기에 설명되는 임의의 시간에 발생할 수 있다.
- <71> 다른 실시예에서, 손상된 콘텐츠를 갖는 멤버는 초기화에 실패할 수 있으며, 따라서 이후 멤버는 다른 멤버들에 접속하려고 시도하지도 않고, 임의의 접속 요청을 승인하지도 못할 것이다.
- <72> 접속 요청의 승인을 거절하는 멤버는 공지된 에러 코드를 반환할 수 있으며, 따라서 다른 멤버는 그 멤버의 콘텐츠가 손상되었음을 알게 된다. 또한, 멤버는 이벤트(예를 들어, EVENT_DFSR_CS_STALE)를 로깅할 수 있다.
- <73> 오래된 타임 스탬프를 가진 복제 그룹의 멤버는 손상된 콘텐츠를 갖는 것으로 간주될 수 있다. 복제 멤버의 콘텐츠가 손상된 것으로 간주되기 전에 허용될 수 있는 로컬 시스템 클러프 타임 스탬프 간의 최대 차이는 시스템 관리자, 사용자 등(이하, 때때로 "시스템 관리자"라고 함)에 의해 설정될 수 있다. 이러한 최대 차이는 때때로 "손상 값"으로 지칭된다.
- <74> 손상의 잘못된 표시를 피하기 위해, 성공적인 동기화 활동들에 응답하여 타임 스탬프가 생성되는 빈도는 손상 값에 대응하는 기간 동안 적어도 한 번 발생하도록 선택될 수 있다. 예를 들어, 멤버가 다른 멤버에 대해 양호한 접속을 유지한 경우, 타임 스탬프는 멤버가 양호한 접속을 유지한 동안 매일 생성될 수 있다. 손상 값이 30일이고, 타임 스탬프가 매일 생성되는 경우, 손상의 잘못된 표시가 발생하는 것은 가능하지 않을 것이다.
- <75> 손상 값에 대응하는 시간 동안 적어도 하나의 생성 가능성을 보증하지 않는 생성 빈도를 선택하는 것은 잘못된 손상 표시를 유발할 수 있다. 예를 들어, 타임 스탬프가 30일 이상마다 한 번씩만 생성되고, 손상 값이 30일인 경우, 타임 스탬프가 생성되지 않는 성공적인 동기화 활동(즉, 양호한 접속)이 발생했을 수 있다.
- <76> 손상 값에 비해 매우 빈번하게 타임 스탬프를 생성하는 것과 덜 빈번하게 타임 스탬프를 생성하는 것 사이의 균형(예를 들어, 오버헤드)이 존재함을 인식할 것이다. 일 실시예에서, 타임 스탬프가 생성되는 빈도는 하루에 한 번인 반면(성공적인 동기화 활동이 하루 동안 발생한 경우), 손상 값은 30일로 설정된다. 다른 실시예들에서는, 예를 들어 구현 및 시스템 구성에 따라 다른 빈도들 및 손상 값들이 선택될 수 있다.
- <77> 또한, 투스톤 수집 기간 이하로 손상 값을 설정함으로써, 다양한 문제가 방지되거나 줄여질 수 있다. 투스톤 수집 기간은 멤버가 자원이 삭제되었음을 지시하는 메타데이터 자원 레코드를 삭제하기 전에 대기하는 기간이다. 통상적으로, 멤버는 투스톤이 복제 그룹 내의 다른 멤버들로 전파할 수 있도록 레코드의 삭제를 미룬다. 멤버가 충분한 기간 동안 복제 그룹의 다른 멤버들로부터 분리되는 경우, 자원에 대한 투스톤이 생성된 후에 메타데이터 자원 레코드가 삭제되었을 수 있다. 여기에 설명되는 메카니즘들이 없다면, 손상된 멤버 상에 자원이 존재하는 경우(그리고, 도 3과 관련하여 설명된 동기화가 발생하는 경우), 자원은 손상된 멤버로부터 복제되어 복제 그룹의 다른 멤버들로 재유입될 수 있다.
- <78> 이러한 거동을 방지하거나 줄이기 위하여, 손상 값은 투스톤 기간 이하로 설정될 수 있다. 투스톤 기간과 비교되는 손상 값이 작으면 작을수록, 삭제된 자원에 대한 투스톤이 또한 모든 손상되지 않은 멤버로 전파되기 전에 삭제될 가능성이 적어진다. 멤버가 투스톤 기간 내에 다른 멤버와 성공적으로 동기화하는 한, 멤버는 투스톤을 수신하고 멤버의 대응 자원을 삭제용으로 표기할 가능성이 커질 것이다. 멤버가 이 기간 내에 다른 멤버와 성공적으로 동기화하지 못하는 경우, 멤버 상의 콘텐츠는 손상된 것으로 표기될 수 있으며, 손상된 콘텐츠가 복제 그룹의 다른 멤버들로 재유입되지 않는 것을 보증하기 위해 후술하는 다른 메카니즘들이 취해질 수 있다.
- <79> 복제된 폴더가 손상될 때, 시스템 관리자는 인바운드 접속이 유효할 때 복제된 폴더가 다른 멤버와 동기화하지 못하게 하거나 다시 동기화할 수 있도록 복제된 폴더를 디스에이블링하거나 리인에이블링할 수 있다. 복제된 폴더를 리인에이블링하는 것은 복제된 폴더가 초기 동기화되게 할 수 있다. 이러한 초기 동기화 동안, 3개의 복제 전략 중 적어도 하나가 후술하는 바와 같이 복제된 폴더들을 동기화하는 데 이용될 수 있다.
- <80> 하나의 전략에서, 복제 파트너들 중 하나(예를 들어, 손상된 콘텐츠를 가진 파트너)는 관독 전용(또는

슬레이브)으로 설정될 수 있다. 요컨대, 판독 전용 멤버 상의 임의의 갱신은 폐기된다. 이 멤버 상에서 파일이 생성되면, 이는 삭제된다. 기존 파일이 삭제되는 경우, 이는 재활된다(예를 들어, 복구된다). 파일이 갱신되면, 갱신은 폐기되고, 최초 버전, 또는 다른 파트너로부터의 새로운 버전이 재설치된다. 일반적으로, 이것은 판독 전용이 아닌 파트너(예를 들어, 마스터)로부터의 모든 갱신이 판독 전용 파트너로 복제되게 한다.

- <81> 제2 전략에서는, 자원들의 병합이 발생한다. 병합에 있어서, 업스트림 파트너로부터의 자원들은 손상 파트너 상의 상이한 모든 자원에 대해 손상 파트너로 복제된다. 손상 파트너가 자원을 포함하지 않는 경우, 그 자원은 손상 파트너로 복제된다. 손상 파트너가 업스트림 파트너가 포함하지 않는 자원을 포함하는 경우, 이 자원은 손상 파트너 상에서 삭제되지 않으며, 손상 파트너가 다른 멤버의 업스트림 파트너인 경우에 손상 파트너로부터 복제될 수 있다.
- <82> 제3 전략에서, 도 3과 관련하여 전술한 바와 같은 동기화가 발생할 수 있다. 일반적으로, 이 전략에서, 보다 최근의 수정 일자를 가진 자원들이 복제되고, 덜 최근인 수정 일자를 가진 다른 멤버들 상의 대응 자원들을 덮어쓴다.
- <83> 복제 그룹의 멤버들 모두가 손상될 수 있다. 이 경우, 복제 멤버들 중 하나가 주 멤버로 설정될 수 있다. 멤버가 주 멤버로 지정되는 경우, 이 멤버는 자원들을 그의 다운스트림 파트너들에 복제할 수 있다. 멤버가 주 멤버로 지정되지 않는 경우, 이 멤버는 자원들을 업스트림 파트너와 성공적으로 동기화할 때까지 자원들을 그의 다운스트림 파트너들에 복제하지 못할 수 있다. 멤버가 다른 멤버에 성공적으로 복제한 후에, 이 멤버는 도 3과 관련하여 설명된 바와 같은 표준 복제 활동들을 재개할 수 있다.
- <84> 동기화 동안 어떤 자원들이 승리하는지(그리고 복제되는지)를 제어하는 펜스 값들이 설정될 수 있다. 펜스 값들은 시스템 관리자가 자원들의 다른 특성들과 무관하게 복제 그룹의 다른 멤버들에 복제되는 멤버의 자원들을 지시하는 것을 가능하게 한다. 예를 들어, 보다 높은 펜스 값이 할당된 자원은 다른 멤버 상의 대응 자원보다 오래된 경우에도 복제될 수 있다.
- <85> 일 실시예에서, 복제 그룹의 모든 멤버가 손상된 경우에 자동 복구가 수행된다. 주 멤버는 모든 멤버가 손상되는 경우에 주 멤버가 되도록 시스템 관리자에 의해 사전 지정되었을 수 있다. 대안으로, 멤버들의 다른 특성들(예를 들어, 위치, 존재할 경우에 계층 구조 순서, 최신 타임 스탬프 등)을 이용하여, 어느 멤버가 주 멤버로 지정되는지를 결정할 수 있다.
- <86> 다른 실시예에서는, 복제 그룹의 모든 멤버가 손상될 때 수동 조정이 수행될 수 있다. 이러한 조정에 있어서, 시스템 관리자는 모든 멤버가 손상되었고 동기화를 재개하기 위해 한 멤버가 주 멤버로 설정될 필요가 있다는 것을 보고받을 수 있다. 이어서, 시스템 관리자는 어느 멤버가 주 멤버인지를 지시할 수 있다.
- <87> 복제 그룹 멤버가 신뢰성 있는 자원들 또는 단지 아웃바운드 접속들(즉, 이 멤버는 업스트림 파트너를 갖지 않음)을 갖는 경우, 이 멤버는 또한, 장기간 다른 멤버들로부터 분리되는 경우에 손상된 것으로 간주할 수 있다. 신뢰성 있는 자원들은 복제 그룹의 모든 다른 멤버들 상의 대응 자원들(신뢰성 있는 것으로 지정된 다른 자원들은 제외)을 무시하도록 지정된 자원들이다. 일 구현에서, 이것은 각각의 신뢰성 있는 자원에 특수 펜스 값을 할당함으로써 수행될 수 있다. 콘텐츠 세트의 자원들(예를 들어, 특정 멤버 상의 자원들)은, 예를 들어 하나의 멤버 상에서 백업 복구가 수행되고, 모든 다른 멤버가 이와 동일한 콘텐츠 세트를 갖는 것이 요구될 때, 신뢰성 있는 것으로 지정될 수 있다.
- <88> 콘텐츠 세트의 자원들은 하나의 멤버가 자원 손실로부터 복구될 때 신뢰성 없는 것으로 지정될 수 있다. 신뢰성이 없는 경우, 멤버의 자원들은 업스트림 파트너들로부터의 자원들로 절하된다. 특정 자원에 대해, 자원 데이터가 업스트림 파트너 상에 포함된 것과 동일한 경우, 자원 데이터는 자원에 대한 자원 메타데이터가 상이한 경우에도 업스트림 파트너로부터 다운로드될 필요가 없다. 콘텐츠 세트 내의 자원들은 예를 들어 복제 그룹의 멤버의 자원들이 손실되거나 오염된 때, 자원들이 백업으로부터 복구된 때, 그리고 자원들이 멤버로부터 복제되지 않는 것이 요구될 때, 신뢰성 없는 것으로 지정될 수 있다.
- <89> 때때로, 멤버의 자원 메타데이터는 오염 또는 손실 또는 삭제될 것이다. 이러한 일이 발생할 경우, 자원 메타데이터를 저장하는 데이터베이스가 삭제되고 자원 데이터로부터 재구축될 수 있다. 이 경우, 멤버의 콘텐츠 세트의 자원들은 또한 신뢰성 없는 것으로 지정될 수 있다.
- <90> 자원 및 그에 대응하는 업스트림 파트너 상의 자원이 신뢰성 있는 것으로 또는 신뢰성 없는 것으로 지정되지 않은 경우, 어느 자원이 승리하고 복제되는지를 결정하기 위해 정상적인 충돌 해결법이 이용될 수 있다.

- <91> 손상된 멤버가 신선해질 필요가 있을 때, 손상된 복제된 폴더가 최신이 되도록 강제하기 위한 인터페이스가 시스템 관리자에게 표시될 수 있다. 일 실시예에서, 인터페이스는 손상된 멤버를 신선하게 하기 위해 타임 스탬프를 개신할 수 있다.
- <92> 시스템의 건강이 모니터링되는 시스템들에서, 건강 모델은 손상 상태의 복제된 폴더를 경고 상태로 간주할 수 있고, 보고서에 경고를 표시할 수 있다. 전술한 바와 같이, 손상 멤버는 시스템 관리자의 조정을 필요로 하지 않을 수 있으며, 따라서 일 실시예에서 손상 상태는 중요한 에러가 아니라 경고로 표시될 수 있다.
- <93> 도 4는 여기에 설명되는 본 발명의 양태들에 따라 로컬 머신 상의 콘텐츠가 복제 그룹의 다른 멤버들과 동기화 할 수 있을 만큼 충분히 신선한지를 결정할 때 발생할 수 있는 액션들을 일반적으로 나타내는 흐름도이다. 블록 405에서, 액션들이 시작된다.
- <94> 블록 410에서, 파트너(업스트림 또는 다운스트림)로부터 접속 요청이 수신된다. 블록 415에서, 요청을 수신한 멤버는 전술한 바와 같이 그의 로컬 자원들이 손상되었는지를 결정한다. 블록 420에서, 로컬 자원들이 손상된 경우, 액션들은 블록 425에서 계속되고, 그렇지 않은 경우에는 액션들은 블록 430에서 계속된다. 블록 425에서, 접속이 거절되고, 로컬 자원들이 손상되었음을 지시하는 메시지가 요청자에게 반환될 수 있다.
- <95> 블록 430에서, 접속 요청이 승인되고, 다른 동기화 활동들(예를 들어, 도 3과 관련하여 설명된 활동들)이 수행 될 수 있다. 블록 435에서, 동기화가 최근에 발생했음을 지시하도록 타임 스탬프가 개신될 수 있다.
- <96> 블록 440에서 액션들이 종료된다.
- <97> 도시되지는 않았지만, 멤버는 다른 멤버와의 접속을 요청하기 전에 그의 로컬 자원들이 손상되었는지를 결정할 수 있다. 그러한 경우, 멤버는 접속 요청을 그만둘 수 있고, 그렇지 않은 경우 멤버는 접속을 요청하고, 다른 동기화 활동들을 수행하며, 그의 타임 스탬프를 개신할 수 있다.
- <98> 도 5는 여기에 설명되는 본 발명의 양태들에 따라 멤버가 손상된 콘텐츠를 갖는 것으로 결정되고 콘텐츠를 복구하도록 슬레이브로서 설정될 때 발생할 수 있는 액션들을 일반적으로 나타내는 흐름도이다. 블록 505에서, 액션들은 초기 동기화 단계에서 시작된다.
- <99> 블록 510에서, 멤버 상에 국지적으로(locally) 포함되지만, 로컬 파트너가 동기하고 있는 업스트림 파트너 상에 포함되지 않은 임의 자원들이 삭제된다.
- <100> 블록 515에서, 국지적으로 포함되지 않고 업스트림 파트너 상에 포함된 임의 자원들이 추가된다.
- <101> 블록 520에서, 업스트림 파트너와 동일하지 않은 임의 자원들이 취득되고, 로컬 자원들이 그에 따라 개신된다.
- <102> 블록 525에서, 액션들이 종료된다. 액션들이 종료된 후, 로컬 파트너는 초기 동기화 단계를 지난 것으로 간주 될 수 있고, 이후 도 4와 관련하여 설명된 바와 같이 자원들을 동기화할 수 있다.
- <103> 도 6은 여기에 설명되는 본 발명의 양태들에 따라 멤버가 손상된 콘텐츠를 갖는 것으로 결정되고 업스트림 파트너와 콘텐츠를 병합하도록 설정될 때 발생할 수 있는 액션들을 일반적으로 나타내는 흐름도이다. 블록 605에서, 액션들은 초기 동기화 단계에서 시작된다.
- <104> 블록 610에서, 업스트림 파트너와 동일하지 않은 임의의 자원들이 취득되고, 로컬 자원들이 이에 따라 개신된다.
- <105> 블록 615에서, 국지적으로 포함되지 않고 업스트림 파트너 상에 포함된 임의 자원들이 추가된다.
- <106> 블록 620에서, 액션들이 종료된다. 액션들이 종료된 후, 로컬 파트너는 초기 동기화 단계를 지난 것으로 간주 되고, 이후 도 4와 관련하여 설명된 바와 같이 자원들을 동기화할 수 있다. 이것은, 전술한 바와 같이 투스톤이 또한 삭제된 다른 멤버들 상에서 삭제된 자원들의 동기화를 포함할 수 있다.
- <107> 도 4-6과 관련하여 설명된 액션들은 복제 시스템들에서 자원 신선도를 검사할 때 취해질 수 있는 모든 액션을 포함하는 것이 아님을 이해해야 한다. 또한, 액션들이 특정 순서로 발생하는 것으로 설명되지만, 다른 실시예 들에서는, 여기에 설명되는 본 발명의 사상 또는 범위를 벗어나지 않고, 액션들 중 일부는 동시에 발생하거나, 다른 순서로 수행될 수 있다.
- <108> 위의 상세한 설명으로부터 알 수 있듯이, 양태들은 콘텐츠 신선도 및 자원 복제 시스템들에 관련하여 설명되었다. 여기에 설명되는 본 발명의 양태들의 다양한 변형 및 대안적인 구성들이 가능하지만, 이를 양태의 소정의

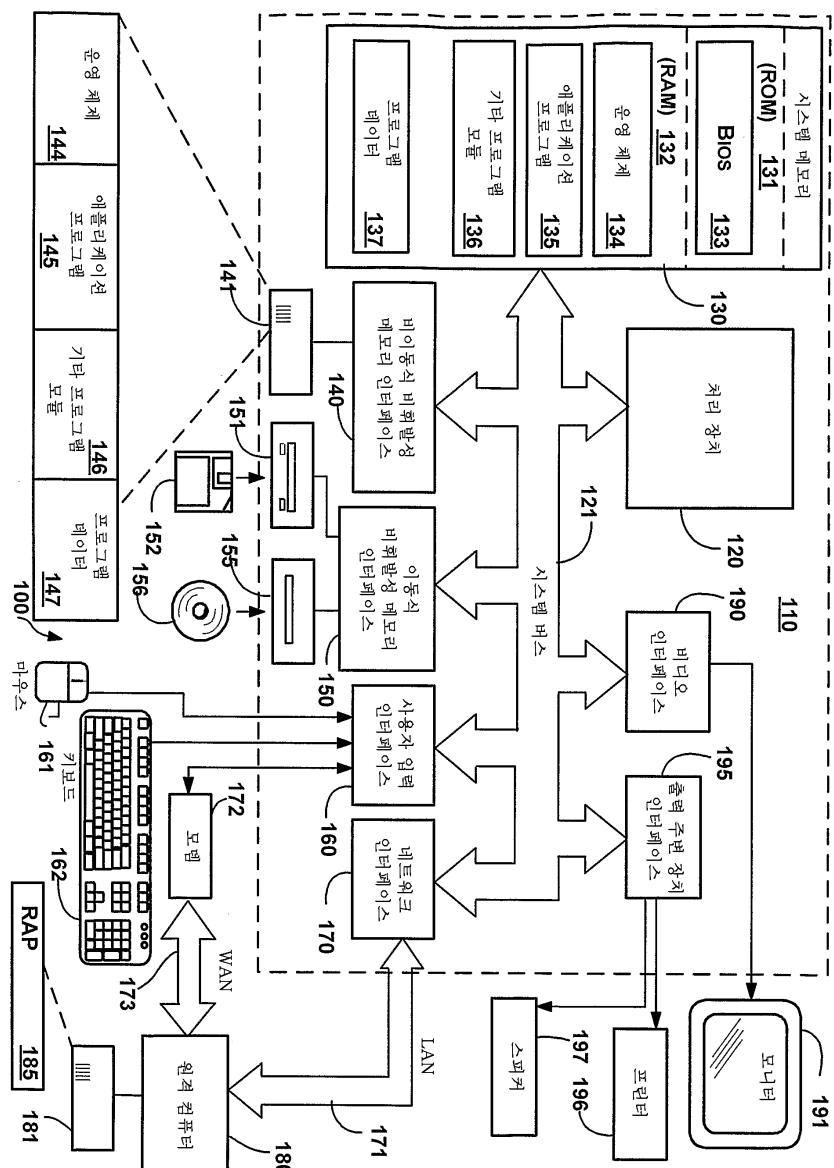
실시예들이 도면들에 도시되고 위에서 상세히 설명되었다. 그러나, 청구 발명의 양태들을 개시되는 특정 형태들로 한정하고자 하는 의도는 없으며, 이와 달리 여기에 설명되는 본 발명의 다양한 양태의 사상 및 범위 내에 있는 모든 변형, 대안 구성 및 균등물을 포함하는 것을 의도한다.

도면의 간단한 설명

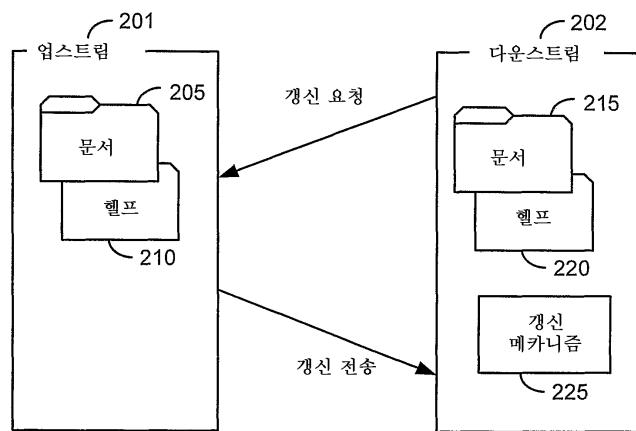
- <8> 도 1은 여기에 설명되는 본 발명의 양태들이 통합될 수 있는 컴퓨터 시스템을 나타내는 블록도.
- <9> 도 2는 여기에 설명되는 본 발명의 양태들에 따라 복제 그룹에 관여하는 머신들을 일반적으로 나타내는 블록도.
- <10> 도 3은 여기에 설명되는 본 발명의 양태들에 따라 자원 복제 시스템에서 동작하도록 구성된 머신을 나타내는 블록도.
- <11> 도 4는 여기에 설명되는 본 발명의 양태들에 따라 로컬 머신 상의 콘텐츠가 복제 그룹의 다른 멤버들과 동기화 할 수 있을 만큼 충분히 신선한지를 결정할 때 발생할 수 있는 액션들을 일반적으로 나타내는 흐름도.
- <12> 도 5는 여기에 설명되는 본 발명의 양태들에 따라 멤버가 손상된 콘텐츠를 갖는 것으로 결정되고 콘텐츠를 복구하기 위해 슬레이브로서 설정될 때 발생할 수 있는 액션들을 일반적으로 나타내는 흐름도.
- <13> 도 6은 여기에 설명되는 본 발명의 양태들에 따라 멤버가 손상된 콘텐츠를 갖는 것으로 결정되고 콘텐츠를 업스트림 파트너와 병합하도록 설정될 때 발생할 수 있는 액션들을 일반적으로 나타내는 흐름도.

도면

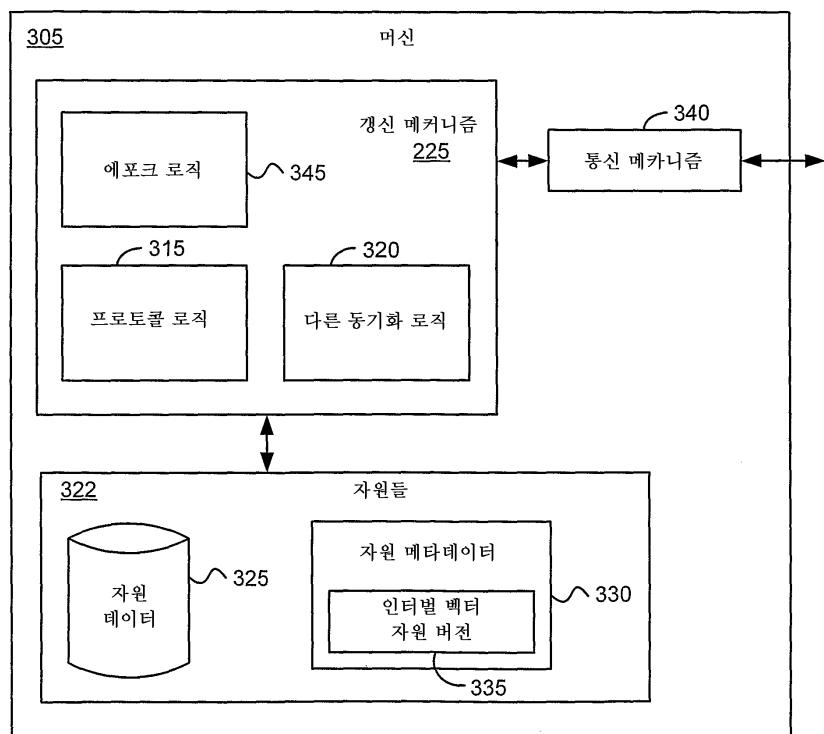
도면1



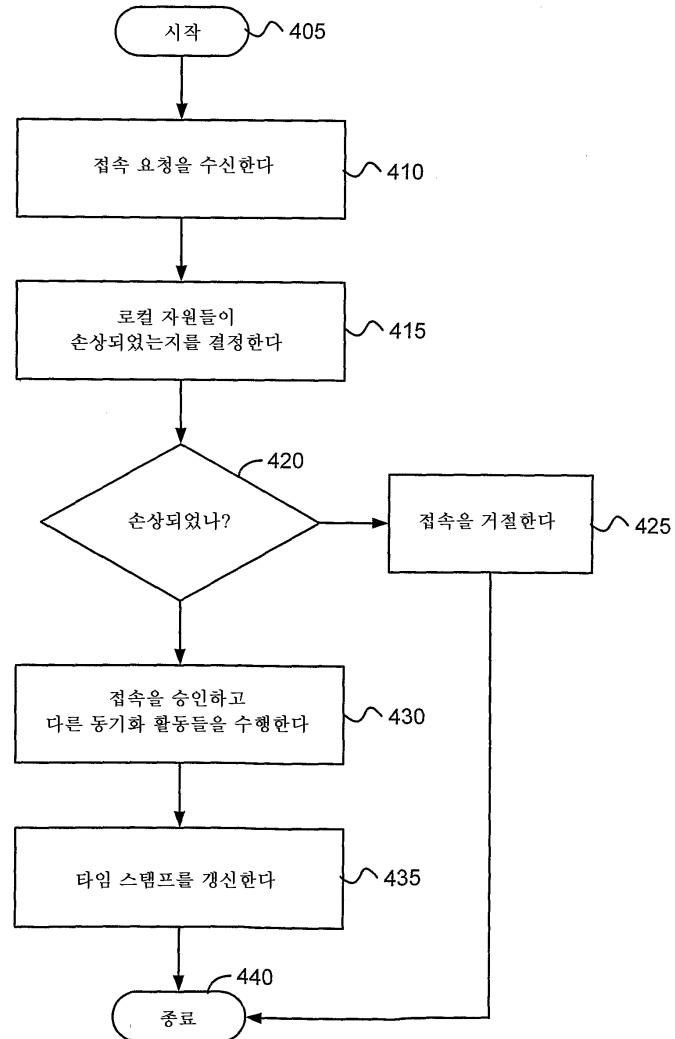
도면2



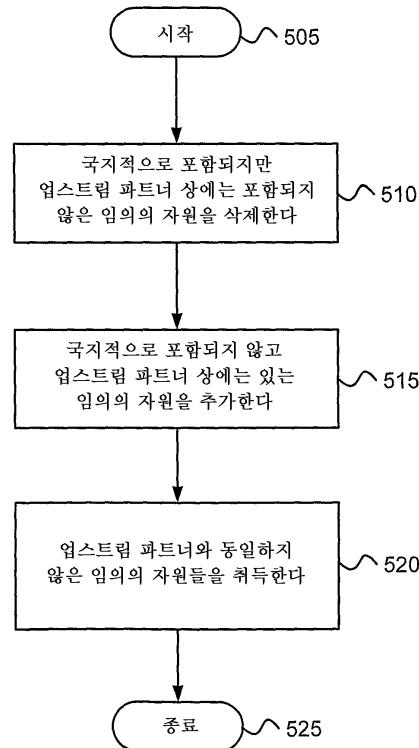
도면3



도면4



도면5



도면6

