



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2018년08월24일
(11) 등록번호 10-1881625
(24) 등록일자 2018년07월18일

(51) 국제특허분류(Int. Cl.)
G06F 9/06 (2018.01) G06F 15/16 (2018.01)
G06F 9/30 (2018.01) G06F 9/44 (2018.01)
(21) 출원번호 10-2013-7034791
(22) 출원일자(국제) 2012년05월30일
심사청구일자 2017년04월28일
(85) 번역문제출일자 2013년12월27일
(65) 공개번호 10-2014-0041604
(43) 공개일자 2014년04월04일
(86) 국제출원번호 PCT/US2012/039963
(87) 국제공개번호 WO 2013/002937
국제공개일자 2013년01월03일
(30) 우선권주장
13/171,446 2011년06월29일 미국(US)
(56) 선행기술조사문헌
JP2007140845 A
(뒷면에 계속)

(73) 특허권자
마이크로소프트 테크놀로지 라이선싱, 엘엘씨
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
마이크로소프트 웨이
(72) 발명자
프리스 로버트
미국 워싱턴주 98052-6399 레드몬드 원 마이크로
소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마
이크로소프트 코포레이션
생비 아쉬빈쿠마르
미국 워싱턴주 98052-6399 레드몬드 원 마이크로
소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마
이크로소프트 코포레이션
(74) 대리인
제일특허법인(유)

전체 청구항 수 : 총 19 항

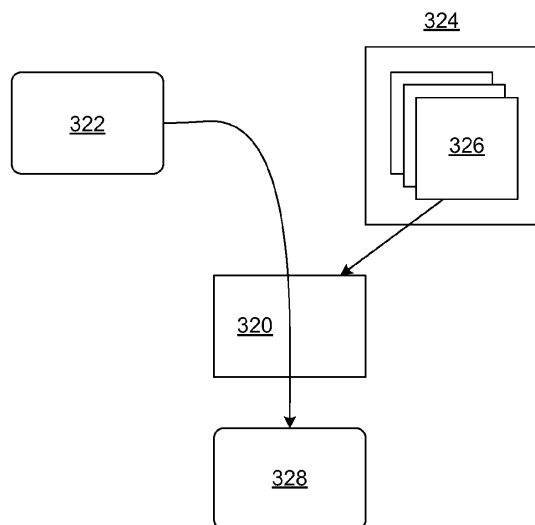
심사관 : 유진태

(54) 발명의 명칭 **가상 머신 이주 도구**

(57) 요약

애플리케이션들을 컴퓨터 클라우드들로 이주시키기 위한 도구들 및 기술들이 본 명세서에서 설명된다. 도구는 임의의 애플리케이션을 컴퓨터 클라우드의 특정 구현으로 이주시키는 데 사용될 수 있다. 도구는 이주 규칙들의 라이브러리를 사용하고, 규칙들을 선택된 애플리케이션에 적용하고, 프로세스에서 이주 출력을 생성할 수 있다. 이주 출력은 권고 정보, 교정된 코드, 패치(patch) 등일 수 있다. 상이한 클라우드 플랫폼들을 위한 상이한 규칙 세트들이 존재할 수 있으며, 이는 애플리케이션이 상이한 클라우드들로 이주되는 것을 가능하게 한다. 규칙들은 광범위한 애플리케이션들 및 애플리케이션을 이주시키기 위한 대응하는 수정 액션들을 설명할 수 있다. 규칙들은 애플리케이션, 코드 또는 호출, 저장소, 데이터베이스 인스턴스, 데이터베이스와의 상호작용, 애플리케이션을 호스팅하는 운영 체제 등의 시맨틱 거동을 지정할 수 있다.

대표도 - 도6



(56) 선행기술조사문헌

US20030188036 A1

US20110035755 A1

US8321857 B1

US20070256058 A1

US20110055377 A1

Frey 외 1명. 'Model-Based Migration of Legacy Software Systems ~'. Proceedings of the First International Conference on Cloud Computing, GRIDs, and Virtualization, 2010.11., pp.155-158.

명세서

청구범위

청구항 1

인터넷을 통한 클라이언트에 의한 액세스를 위해 임의의 애플리케이션을 호스팅하는 애플리케이션 호스팅 클라우드에 애플리케이션을 이주시키는 방법으로서,

상기 애플리케이션 호스팅 클라우드로 이주시킬 타겟 애플리케이션을 선택하는 단계 - 상기 애플리케이션 호스팅 클라우드는 애플리케이션이 마치 하드웨어 플랫폼 상에서 실행되는 것처럼 실행될 수 있는 클라우드 기반 가상 플랫폼을 제공하고, 상기 타겟 애플리케이션은 이주(migration) 전에 특정 컴퓨터 호스트 플랫폼 상에서 실행됨 - 와,

상기 타겟 애플리케이션을 이주 도구(migration tool)로 전달하는 단계 - 상기 이주 도구는 하나 이상인 복수의 이주 규칙을 포함하는 규칙 라이브러리에 대한 액세스를 제공받고, 각각의 복수의 이주 규칙은 제각기의 상이한 애플리케이션 호스팅 클라우드에 대응하고, 상기 복수의 이주 규칙은 코드 규칙, 운영 체제 규칙, 시맨틱(semantics) 규칙, 데이터베이스 규칙 또는 인스톨러(installer) 규칙 중 적어도 하나를 설명함 - 와,

상기 규칙 라이브러리로부터 상기 타겟 애플리케이션에 적용 가능한 것으로 판정된 하나 이상의 이주 규칙을 식별하기 위해 상기 이주 도구를 이용하여 상기 타겟 애플리케이션의 소스 파일과 구성 파일을 파싱하고 분석하는 단계 - 하나의 이주 규칙은 특정 네트워크 서비스에 대한 참조를 상기 애플리케이션 호스팅 클라우드에 의해 제공되는 유사한 서비스로 변환함 - 와,

상기 특정 컴퓨터 호스트 플랫폼을 위해 구성되는 것에서 상기 애플리케이션 호스팅 클라우드를 위해 적어도 부분적으로 구성되는 것으로 상기 타겟 애플리케이션의 상기 소스 파일과 상기 구성 파일을 변경하기 위해 출력의 형태를 생성하도록 상기 식별된 하나 이상의 이주 규칙을 상기 타겟 애플리케이션에 적용하는 단계를 포함하는 방법.

청구항 2

제1항에 있어서,

상기 애플리케이션 호스팅 클라우드는 상기 애플리케이션을 인스턴스화하고 상기 애플리케이션의 인스턴스를 종료시킴으로써 상기 애플리케이션 호스팅 클라우드에서 호스팅되는 상기 애플리케이션에 대한 부하 균형화(load balancing)를 수행하고, 식별된 규칙은 상기 부하 균형화와 관련된 시맨틱 거동을 포함하는 조건을 포함하는 방법.

청구항 3

제2항에 있어서,

상기 타겟 애플리케이션은 인스톨 패키지를 포함하고, 상기 파싱하고 분석하는 단계는 상기 인스톨 패키지를 여는 것과 상기 타겟 애플리케이션의 구성 설정을 식별하는 것을 포함하고, 식별된 규칙은 구성 설정을 변경하는 방법.

청구항 4

제1항에 있어서,

상기 하나 이상의 식별된 규칙은 상기 애플리케이션 호스팅 클라우드 내에 지속되는 데이터와 관련된 규칙을 포

합하는
방법.

청구항 5

제4항에 있어서,

상기 하나 이상의 식별된 규칙을 적용하는 단계는 상기 애플리케이션 호스팅 클라우드에 의해 처리되는 기능을 수행하는 상기 타겟 애플리케이션의 부분을 식별하는

방법.

청구항 6

제1항에 있어서,

상기 하나 이상의 식별된 이주 규칙을 적용하는 단계는, 또한, 상기 애플리케이션 호스팅 클라우드에서의 실행을 가능하게 하기 위한 상기 타겟 애플리케이션에 대해 제안된 변경을 식별하는 보고서를 생성하는

방법.

청구항 7

제1항에 있어서,

상기 하나 이상의 식별된 규칙 중 하나는 SQL(structure queried language) 코드 및 코드 변경의 설명을 포함하고, 상기 하나 이상의 식별된 이주 규칙을 적용하는 상기 단계는 상기 설명과 매칭되는 상기 타겟 애플리케이션의 코드의 일부를 변경하기 위해 상기 하나 이상의 식별된 규칙 중 상기 하나를 적용하는 단계를 포함하고, 변경 전의 상기 코드는 상기 애플리케이션 호스팅 클라우드에서 올바르게 실행될 수 없는

방법.

청구항 8

컴퓨터와,

상기 컴퓨터와 연결되어 상기 컴퓨터로 하여금 상기 컴퓨터에 의해 실행가능한 이주 도구를 구현하도록 하는 정보를 저장하는 저장 하드웨어를 포함하되,

상기 이주 도구는,

PaaS(platform-as-a-service) 클라우드에 이주될 임의의 애플리케이션에 대해 적용될 수 있는 구성, 코드, 운영 체제, 데이터베이스 또는 인스톨 속성 중 적어도 하나와 시맨틱을 설명하는 이주 규칙을 포함하는 규칙 라이브러리 - 상기 규칙 라이브러리는 상기 PaaS 클라우드의 호스팅 기능과 중복되는 애플리케이션 로직을 식별하는 하나 이상의 규칙을 더 포함함 - 와,

프론트엔드 애플리케이션과 상기 프론트엔드 애플리케이션과 협업하도록 프로그래밍된 백엔드 데이터 저장 애플리케이션으로 구성된 타겟 웹 애플리케이션의 코드 파일과 구성 파일을 파싱하는 하나 이상의 파서(parser) - 상기 파서는 상기 타겟 웹 애플리케이션에 적용가능한 규칙을 식별함 - 를 포함하고,

상기 하나 이상의 규칙은 상기 타겟 웹 애플리케이션에 적용되는 경우 상기 PaaS 클라우드의 상기 타겟 웹 애플리케이션의 실행을 가능하게 하거나 향상시키도록 상기 타겟 웹 애플리케이션에 행해진 변경을 식별하는데 사용되는 출력을 생성하고,

상기 변경은 상기 PaaS 클라우드의 호스팅 기능과 중복되는 것으로 식별된 규칙을 적용함으로써 식별된 애플리

케이션 로직을 상기 웹 애플리케이션으로부터 제거하는 것을 포함하는 장치.

청구항 9

제8항에 있어서,

상기 규칙의 일부는 SQL 호출 또는 SQL 선언문을 설명하고,

상기 하나 이상의 규칙을 상기 타겟 웹 애플리케이션에 적용하는 것은 변경된 호출 또는 선언문이 PaaS 클라우드에서 실행될 수 있도록 상기 타겟 웹 애플리케이션의 코드를 변경하는 것을 포함하는

장치.

청구항 10

제9항에 있어서,

상기 변경된 호출 또는 선언문은 실행될 경우에 상기 PaaS 클라우드의 데이터베이스 서비스를 인보킹하고 (invoke),

상기 데이터베이스 서비스는 상기 PaaS 클라우드에서 실행되는 임의의 애플리케이션에 대해 이용가능한 상기 PaaS 클라우드의 일부를 포함하고,

상기 데이터베이스 서비스는 상기 애플리케이션에 대해 격리되고 스케일링 가능한 데이터베이스 인스턴스를 각각 제공하는

장치.

청구항 11

제8항에 있어서,

상기 하나 이상의 규칙은 상기 타겟 웹 애플리케이션에 적용되는 경우, 로컬 파일 시스템 상의 저장소를 특정하는 상기 타겟 웹 애플리케이션의 저장 로직을 식별하고, 상기 저장 로직은, 상기 PaaS 클라우드의 영구 데이터 저장소를 사용하도록 상기 로직을 교정하거나 상기 타겟 웹 애플리케이션의 저장 로직을 식별하는 상기 로직을 식별하고, 상기 영구 데이터 저장소를 권고하는 정보를 출력하는

장치.

청구항 12

제8항에 있어서,

상기 타겟 웹 애플리케이션은 가상 머신 이미지에 저장되고, 상기 이주 도구는 상기 가상 머신 이미지를 적재하며, 상기 하나 이상의 파서는 상기 규칙을 적용하도록 상기 타겟 웹 애플리케이션을 파싱하는

장치.

청구항 13

제8항에 있어서,

상기 타겟 웹 애플리케이션은 데이터베이스 스키마(schema)를 포함하고 표준 배포 패키지 포맷에 따른 배포 패

키지에 저장되며, 상기 이주 도구는 상기 표준 배포 패키지 포맷을 판독하는 컴포넌트를 갖는 장치.

청구항 14

애플리케이션을 PaaS 클라우드에 이주시키는 방법으로서,

상기 PaaS 클라우드로 이주시킬 타겟 애플리케이션을 선택하는 단계 - 상기 타겟 애플리케이션은 이주되기 전에 특정 컴퓨터 호스트 플랫폼 상에서 실행됨 - 와,

상기 타겟 애플리케이션을 이주 도구로 전달하는 단계 - 상기 이주 도구는 하나 이상인 복수의 이주 규칙을 포함하는 규칙 라이브러리를 포함하고, 상기 복수의 이주 규칙의 적어도 하나는 상기 PaaS 클라우드에 특정되며, 상기 복수의 이주 규칙은 코드 규칙, 운영 체제 규칙, 시맨틱 규칙, 데이터베이스 규칙 또는 인스톨러 규칙 중 적어도 하나를 설명함 - 와,

상기 규칙 라이브러리로부터 상기 PaaS 클라우드에 특정된 복수의 이주 규칙에 액세스하는 단계 - 상기 복수의 이주 규칙은 임의의 애플리케이션에 적용되는 경우 상기 애플리케이션에 의한 소프트웨어 제품의 사용 또는 상기 소프트웨어 제품에 대한 참조를 식별하며, 상기 소프트웨어 제품은 (i) 라이선스 기반으로 상기 애플리케이션에 의해 사용될 수 있고, (ii) 상기 PaaS 클라우드에 의한 라이선싱을 제공받을 수 있으며, 상기 복수의 이주 규칙은 상기 PaaS 클라우드에 호스팅되는 것으로 구성될 경우에 상기 애플리케이션에 행해질 변경을 식별하기 위한 것임 - 와,

상기 규칙 라이브러리로부터의 상기 복수의 이주 규칙을 상기 PaaS 클라우드에서 실행되도록 아직 구성되지 않은 애플리케이션에 적용하는 단계 - 상기 애플리케이션은 상기 PaaS 클라우드가 아닌 시스템에 의해 제공된 상기 소프트웨어 제품 중 하나를 사용하도록 구성되며, 상기 적용하는 단계는 상기 애플리케이션을 인스톨 및 컴파일링하여 출력의 형태를 생성하기 위해 사용되도록 구성되는 파일을 파싱하는 단계를 포함함 - 와,

상기 복수의 이주 규칙과 상기 파일에 따라 상기 출력으로부터, 상기 PaaS 클라우드에 의해 제공되는 상기 소프트웨어 제품을 사용하도록 변경됨으로써 상기 애플리케이션이 상기 소프트웨어 제품과 관련하여 더 낮은 라이선싱 비용을 갖도록 변경될 수 있는지를 판정하고, 상기 판정에 기초하여 상기 PaaS 클라우드에 의해 제공되는 소프트웨어 제품을 사용하기 위해 대응하는 변경 또는 변경의 권고를 생성하는 단계를 포함하는

방법.

청구항 15

제14항에 있어서,

추가적인 이주 규칙이 온-클라우드(non-cloud) 네트워크 자원의 사용으로부터 상기 PaaS 클라우드에 의해 제공된 네트워크 자원의 사용으로 변환될 수 있는 상기 애플리케이션의 코드를 식별하는

방법.

청구항 16

제15항에 있어서,

상기 온-클라우드 네트워크 자원은 부하 균형화 자원, 데이터베이스 서비스 자원, 데이터 저장 서비스 또는 스케일링 자원을 포함하는

방법.

청구항 17

제14항에 있어서,

상기 애플리케이션은 웹 프론트엔드를 적어도 부분적으로 포함하고, 상기 변경 또는 변경의 권고는 상기 웹 프론트엔드에 의해 클라이언트 상태가 어떻게 유지되는지와 관련되는 방법.

청구항 18

제14항에 있어서,

상기 복수의 규칙은 PaaS 클라우드 내에 지속되는 데이터와 관련된 규칙을 포함하는 방법.

청구항 19

제14항에 있어서,

상기 애플리케이션에 적용되는 경우에, 상기 복수의 규칙 중 하나의 규칙은 구현될 경우, 상기 애플리케이션과 연관된, 상기 소프트웨어 제품 중 대응하는 하나의 라이선싱 비용을 제거할 수 있는 권고 또는 변경을 제공하는 방법.

발명의 설명

기술 분야

배경 기술

[0001] 최근, 때때로 PaaS(Platform as a Service)로서 지칭되는 컴퓨트 클라우드들의 이용 및 가용성이 증가해왔다. 컴퓨트 클라우드들의 예들은 Windows Azure(TM), Amazon EC2(TM), Bungee Connect(TM), Google App Engine(TM) 등이다. 이러한 컴퓨트 클라우드들은 통상적으로 많은 테넌트(tenant)를 호스팅하며, 테넌트들 각각은 통상적으로 클라이언트 브라우저들에 의해 액세스되는 그들 자신의 격리된 웹 애플리케이션들 또는 웹 서비스들을 실행한다. 테넌트의 애플리케이션들은 종종 가상 머신들(VM들)에서 실행된다. 컴퓨트 클라우드들은 애플리케이션들에 투명하도록 의도된 방식으로 조건들 및 요구들의 변화를 처리할 수 있는 실행 환경을 제공한다. 예를 들어, 착신되는 요청들의 부하의 균형화, 네트워크 대역폭, 처리 자원들, 저장소의 준비, 애플리케이션들의 스케일링(예를 들어, 인스턴스들의 수의 조정), 가상 머신들 및 애플리케이션 인스턴스들의 재배포 등을 포함한다. 공유 컴퓨팅 클라우드들은 운영자 엔티티에 의해 관리되며, 이는 테넌트들이 주로 그들의 애플리케이션들에 관련되게 한다.

[0002] 그러나, 실행 환경으로서의 컴퓨팅 클라우드는 원래 컴퓨팅 클라우드 상에서 실행되도록 설계되지 않은 애플리케이션들에 맞지 않는, 이익들 및 한계들 양자를 포함하는 특징들을 가질 수 있다. 예를 들어, 원래 특정 non-클라우드(non-cloud) 자원들(예를 들어, 관계형 데이터베이스들) 및 아마도 다양한 소프트웨어 및 하드웨어 설비들을 이용하는 특정 운영 체제들 상에서 실행되도록 설계된 3층 웹 애플리케이션을 고려한다. 애플리케이션은 부하의 변동을 처리하기 위한 내장 논리를 갖는 웹 프론트엔드를 구비할 수 있다. 프론트엔드는 비즈니스 논리를 구현하는 그리고 로컬 파일 저장소 및 데이터베이스와 같은 백엔드 저장소와 상호작용하는 중간 계층과 인터페이스할 수 있다. 이 애플리케이션은 클라우드에서 필요하지 않은 자기 스케일링을 위한 시맨틱스(semantics)를 가질 수 있다. 애플리케이션은 그 자신의 데이터베이스 계층 및 클라우드에서 필요하지 않은 부수적인 관리 소프트웨어를 가질 수 있다. 애플리케이션은 클라우드에 의한 제어와 충돌하는 운영 체제 구성 설정들을 가질 수 있다(일부 클라우드들은 운영 체제를 필요로 하지 않을 수도 있다). 애플리케이션이 컴퓨팅 클라우드에서 효율적으로 실행될 수 있게 하기 위해 애플리케이션의 양태들이 변경, 제거 또는 추가되는 것이 필요할 수 있다.

[0003] 아래에서 논의되는 기법들은 애플리케이션들 및 가상 머신들을 컴퓨팅 클라우드들로 이주시키기 위한 도구들과 관련된다.

발명의 내용

[0004] 아래의 요약은 아래의 상세한 설명에서 논의되는 일부 개념들을 소개하기 위해서만 포함된다. 이 요약은 포괄적이지 않고 청구된 발명 대상의 범위를 정의하는 것을 의도하지 않으며, 이 범위는 마지막에 제공되는 청구범위에 의해 설명된다.

도면의 간단한 설명

- [0005] 도 1은 범용 컴퓨팅 클라우드를 나타낸다.
- 도 2는 범용 컴퓨팅 클라우드의 다른 뷰를 나타낸다.
- 도 3은 2개의 예시적인 컴퓨팅 클라우드 아키텍처를 나타낸다.
- 도 4는 다른 컴퓨팅 클라우드 아키텍처를 나타낸다.
- 도 5는 타겟 애플리케이션의 컴퓨팅 클라우드로의 예시적인 이주를 나타낸다.
- 도 6은 이주 도구를 나타낸다.
- 도 7은 규칙들 또는 이주 라이브러리의 뷰를 나타낸다.
- 도 8은 이주 도구에 의해 수행되는 프로세스를 나타낸다.
- 도 9는 이주 도구에 의해 사용되는 관독 도구들의 예시적인 세트를 나타낸다.

발명을 실시하기 위한 구체적인 내용

- [0006] 수반되는 특징들 중 다수가 첨부 도면들과 관련하여 고려되는 아래의 상세한 설명을 참조하여 아래에 설명된다.
- [0007] 본 설명은 첨부 도면들을 고려하여 아래의 상세한 설명을 읽을 때 더 잘 이해될 것이며, 여기에서 동일한 참조 번호들은 첨부된 설명 내의 동일한 요소들을 지시하는 데 사용된다.
- [0008] 애플리케이션들을 컴퓨팅 클라우드들로 이주시키기 위한 도구들 및 기술들이 본 명세서에서 설명된다. 도구는 임의의 애플리케이션을 컴퓨팅 클라우드의 특정 구현으로 이주시키는 데 사용될 수 있다. 도구는 이주 규칙들의 라이브러리를 사용하고, 규칙들을 선택된 애플리케이션에 적용하고, 프로세스에서 이주 출력을 생성할 수 있다. 이주 출력은 권고 정보, 교정된 코드, 패치(patch) 등일 수 있다. 상이한 클라우드 플랫폼들을 위한 상이한 규칙 세트들이 존재할 수 있으며, 이는 애플리케이션이 상이한 클라우드들로 이주되는 것을 가능하게 한다. 규칙들은 광범위한 애플리케이션들 및 애플리케이션을 이주시키기 위한 대응하는 수정 액션들을 설명할 수 있다. 규칙들은 애플리케이션, 코드 또는 호출, 저장소, 데이터베이스 인스턴스, 데이터베이스와의 상호작용, 애플리케이션을 호스팅하는 운영 체제 등의 시맨틱 거동을 지정할 수 있다.
- [0009] 개요
- [0010] 후술하는 실시예들은 애플리케이션들을 컴퓨팅 클라우드들로 이주시키기 위한 이주 도구들과 관련된다. 설명은 컴퓨팅 클라우드들의 설명으로부터 시작되고, 여러 가지 예가 이어질 것이다. 예시적인 애플리케이션이 설명될 것이다. 이어서, 이주 도구들, 이주 규칙들 및 이주를 위한 프로세스들을 포함하는 이주를 위한 도구들 및 기술들이 설명될 것이다.
- [0011] 컴퓨팅 클라우드들
- [0012] 도 1은 범용 컴퓨팅 클라우드(100)를 나타낸다. 데이터 네트워크(102)가 컴퓨팅 클라우드(100)를 구성하는 다양한 컴퓨터들(도시되지 않음) 사이의 접속을 제공한다. 일반적으로, 많은 수의 컴퓨터가 가상 머신들을 호스팅하고, 가상 머신들은 격리된 테넌트 애플리케이션들을 호스팅한다. 다양한 클라우드 서비스들(104)은 통신 규, 부하 균형화 등과 같은 기능들을 제공할 수 있다. 클라우드 플랫폼(106)은 테넌트들이 그들의 애플리케이션들을 업로드 및 관리할 수 있게 해주는 테넌트용 인터페이스로서 작용할 수 있다. 클라우드 플랫폼(106)은 또한 테넌트들의 애플리케이션들을 랩핑(wrapping) 및 관리하여, 각각의 애플리케이션에 대한 컴퓨팅 환경을 효

과적으로 제공할 수 있다. 클라우드 기반구조(108)는 비용청구 및 관리 요소들을 포함할 수 있다. 예를 들어, 클라우드 기반구조(108)는 애플리케이션들 및/또는 컴퓨팅 클라우드(100)의 부하 변화들을 처리하기 위해 컴퓨터들을 온라인화 및 오프라인화할 수 있다. 클라우드 저장소(110)는 다양한 형태들, 예를 들어 각각의 테넌트에 의해 제어되고 구성되는 데이터베이스들의 인스턴스들을 제공하는 관계형 데이터베이스 서비스, 간단한 블로브(blob)(이진 대형 객체) 저장소, 테이블 저장소, 파일 시스템 저장소 등의 형태를 취할 수 있다.

[0013] 도 2는 범용 컴퓨팅 클라우드(100)의 다른 뷰를 나타낸다. 이 뷰에서, 테넌트들(120, 122)은 클라우드에 의해 호스팅되는 각각의 애플리케이션(124, 126)을 갖는다. 컴퓨팅 클라우드(100)는 애플리케이션들(124, 126)에 대한 컴퓨터 환경들(128, 130)을 관리하는 패브릭(125)을 갖는다. 패브릭(125)은 게스트 운영 체제, 저장 서비스 등을 갖는 VM들을 실행하는 많은 컴퓨터를 가질 수 있다. 애플리케이션들(124, 126)은 웹 기반 액세스를 위해 통상적인 다양한 컴포넌트들을 포함할 수 있으며, 컴퓨팅 클라우드(100)에 의해 제공되는 자원들을 사용할 수 있다. 컴퓨터 환경들(128, 130)은 (아마존 머신 이미지들(AMI들)에 의해 구성되는 바와 같은) Amazon EC2 인스턴스들, (Microsoft Azure에서와 같은) 역할들, (Google App Engine에서와 같은) 관리되는 코드 환경들을 갖는 샌드박스 및 시뮬레이션된 부분 운영 체제들 등과 유사할 수 있다. 컴퓨터 환경들(128, 130)은 부하, 네트워크 트래픽, 예상치 못한 고장 등과 같은 현재의 조건들에 따라 패브릭(125)에 의해 "확장"될 수 있다. 그러한 확장은 필요에 따라 계산 자원들(하드웨어, VM, 서비스 인스턴스, 데이터베이스 인스턴스 등)을 투명하게 추가 또는 제거하는 것을 포함할 수 있다.

[0014] 애플리케이션들(124, 126)은 테넌트들(120, 122)에 의해 업로드 및 구성된다. 애플리케이션들(124, 126)은 컴퓨팅 클라우드(100)에 의해 관리되는 바와 같이 실행되며, 클라이언트들(128)은 브라우저들 또는 다른 타입의 클라이언트 소프트웨어를 이용하여 애플리케이션들(124, 126)의 인스턴스들에 액세스한다. 애플리케이션의 관점에서, 애플리케이션은 플랫폼 상에서 실행되고 있으며, 컴퓨팅 클라우드(100)의 활동은 대부분 투명하다는 점에 유의한다. 애플리케이션들은 기반 하드웨어, 데이터 네트워크 또는 애플리케이션과 클라이언트 사이의 클라우드 계층에 대한 어떠한 관계도 없이 통신 프로토콜들을 통해 액세스된다.

[0015] 도 3은 2개의 예시적인 컴퓨팅 클라우드 아키텍처를 나타낸다. 클라우드 아키텍처(250)는 Amazon EC2 클라우드의 일 버전이다. 애플리케이션 개발 및 배치는 테넌트 클라이언트에 의해 처리된다. 클라우드는 머신 이미지들 및 주문형 인스턴스들의 형태의 클라우드 컴퓨팅 서비스들을 제공한다. 애플리케이션들은 가상 머신들 내의 게스트 운영 체제들에서 호스팅된다. 가상 머신들 및 데이터베이스들의 인스턴스들은 지원 서비스들로부터의 요구에 따라 클라우드에 의해 제공된다. 큐 서비스가 가상 머신들과 애플리케이션 인스턴스들 간의 통신을 용이하게 할 수 있다. 이러한 컴포넌트들이 어떻게 동작하고 협동하는지에 대한 상세들은 다른 곳에서 입수할 수 있다.

[0016] 클라우드 아키텍처(252)는 Google App Engine 클라우드의 일 버전이다. 애플리케이션을 생성하고 배치하기 위해 다양한 개발 도구들이 사용된다. App Engine 자체는 다른 곳에 충분히 기록되어 있다. 하나의 중요한 특징은, 애플리케이션이 배치되면, App Engine이 자동으로 스케일링을 처리하며, 자원들 및/또는 인스턴스들이 필요에 따라 추가 및 제거된다는 것이다. 다양한 지원 서비스들이 애플리케이션들에 의해 액세스될 수 있다. 계정 서비스, 데이터 테이블 서비스 등이 애플리케이션들에 의해 이용되며, 이러한 자원들도 클라우드에 의해 스케일링 및 관리된다.

[0017] 도 4는 Microsoft Azure의 일 버전에 대한 컴퓨팅 클라우드 아키텍처(254)를 나타낸다. 관리되는 코드를 갖는 개별 스케일링 가능 컴포넌트들인 역할들이 제공된다. 작업자 역할들은 일반화된 개발을 위한 것이며, 웹 역할에 대한 배경 처리를 수행할 수 있다. 웹 역할들은 웹 서버를 제공하며, 하이퍼텍스트 전송 프로토콜(HTTP) 또는 HTTP 보안(HTTPS) 엔드포인트를 통해 웹 요청들을 청취하고 응답한다. VM 역할들은 테넌트 정의 구성들(예로서, 자원, 게스트 운영 체제)에 따라 인스턴스화된다. 운영 체제 및 VM 갱신들은 클라우드에 의해 관리된다. 웹 역할 및 작업자 역할은 테넌트의 제어하에 있는 가상 머신인 VM 역할에서 실행된다. 저장 및 SQL 서비스들은 역할들에 의한 사용에 유효하다. 다른 클라우드들에서와 같이, 스케일링, 부하 균형화 등을 포함하는 하드웨어 및 소프트웨어 환경 또는 플랫폼은 클라우드에 의해 처리된다.

[0018] 요건대, PaaS 타입의 컴퓨팅 클라우드들에서는, 클라우드 컴퓨팅 플랫폼 자체가 대부분의 관리 기능들을 처리한다. 플랫폼은 운영 체제 패치들의 적용, 시스템 또는 데이터베이스 소프트웨어의 새로운 버전들의 인스톨, 컴퓨터들 및 VM들의 온라인화, VM들의 이주, 네트워크 대역폭의 할당 등과 같은 것들을 자동으로(그리고 테넌트들에 투명하게) 처리할 수 있다. 그러나, (다음 섹션에서 설명되는) 애플리케이션들의 소정의 시맨틱 거동과 교차할 수 있는 이러한 투명한 관리는 패칭, 하드웨어 고장, 과부하 및 다른 이유들로 인한 애플리케이션 이용 불

가능성을 제거할 수 있다. 더욱이, 물리 및 가상 머신들의 제어 내에 있는 클라우드는 애플리케이션 스케일링을 처리하며, 클라우드는 적절한 레벨의 자원들이 임의의 주어진 시간에 이용가능한 것을 보증한다. 컴퓨팅 클라우드 플랫폼들은 다른 특징들을 가질 수 있다. 예를 들어, 브라우저 기반 개발 도구들, 클라우드 내의 호스팅되는 실행시간 환경에 대한 무결함 배치(즉, 클라우드에 액세스하는 클라이언트로부터 애플리케이션을 배치하고 개시하는 능력), 테넌트용 웹 기반 관리 및 모니터링 도구들, 현금 지불 방식(pay-as-you-go)의 비용 청구 등을 포함한다.

[0019] 애플리케이션 이주

[0020] 전술한 바와 같이, 원래 컴퓨팅 클라우드에서 실행되도록 생성되지 않은 애플리케이션은 컴퓨팅 클라우드의 아키텍처 및 서비스들에 의해 영향을 받을 수 있는 설계 특징들(시맨틱스), 코드 특성들 및 구성 특징들을 가질 수 있다. 애플리케이션은 클라우드 환경에서 중복되는 부하 균형화 및 스케일링과 같은 기능을 가질 수 있다. 애플리케이션은 클라우드 환경에서 에러, 데이터 손실 또는 다른 장애를 유발할 수 있는 특징들도 가질 수 있다. 애플리케이션을 클라우드 환경으로 이주시킬 때, 호환성, 신뢰성, 효율, 비용 최소화, 적절한 인스톨 등을 위해 행해질 수 있는 또는 행해져야 하는 변경들이 종종 존재한다.

[0021] 도 5는 타겟 애플리케이션(280)의 컴퓨팅 클라우드로의 예시적인 이주를 나타낸다. 타겟 애플리케이션(280)은 원래 특정 운영 체제 및 맞춤형 기반구조, 예를 들어 인 하우스 정보 기술(IT) 환경을 위해 설계된 머신 또는 호스트 기반 애플리케이션이다. 타겟 애플리케이션(280)은 클라이언트 요청들을 처리하는 웹 서버들(282)의 프론트엔드를 포함하는 3층 아키텍처를 갖는다. 맞춤형 부하 균형화기(284)가 클라이언트 요청들을 웹 서버들(282) 사이에 분산시킨다. 중간 계층은 타겟 애플리케이션(280)의 논리 및 주요 기능을 처리하는 애플리케이션 서버들(286)을 포함한다. 중간 계층은 SQL 서버(288)에 의해 관리되는 데이터베이스 내에 데이터 및 애플리케이션 상태를 저장한다. SQL 서버는 장애 극복 및 백업을 위해 데이터베이스 미러를 유지한다. 애플리케이션 서버들(286)은 SQL 호출 등을 이용하여 SQL 서버(288)와 인터페이스한다. 부하 균형화기(290)는 웹 서버들(282)과 애플리케이션 서버들(286) 간의 상호작용을 균형화한다. 타겟 애플리케이션(280)은 전술한 요소들 중 임의의 요소의 인스턴스들을 추가함으로써 스케일링하기 위한 맞춤형 논리를 구비할 수 있다. 더욱이, 타겟 애플리케이션(280)이 실행되는 컴퓨터 플랫폼들을 관리하는 관리 소프트웨어의 계층이 존재할 수 있다. 이 계층은 백업, 시스템 갱신, 줌비 프로세스 또는 시스템의 재개, 호스트 컴퓨터들 사이의 가상 머신들의 이주, 장애 극복 시스템들로의 재지향 등을 수행할 수 있다.

[0022] 도 5의 아랫부분은 이주된 애플리케이션(292)을 나타낸다. 이주된 버전은 후술하는 다양한 방식으로 변경될 수 있다. 예를 들어, (이주된 애플리케이션(292)의 가시성 없이) 컴퓨팅 클라우드에 의해 부하 균형화 메커니즘(294)이 제공될 수 있다. 이주된 애플리케이션(292)의 웹 서버들(296)은 제거된 HTTP 서버들일 수 있으며, HTTP 요청들을 처리하기 위해 컴퓨팅 클라우드에 의존할 수 있다. 또는, 웹 서버들(296)은 클라우드에 의해 인스턴스화되고 관리될 수 있다. 클라우드는 또한 이주된 애플리케이션 서버들(298)에 대한 부하 균형화 및 스케일링을 제공할 수 있다. 이주된 애플리케이션(292)의 데이터 계층은 여전히 SQL 명령문들 및 (아마도 변경된) 논리를 사용하지만, 애플리케이션 데이터는 이제 클라우드 관리 데이터베이스 인스턴스(300)로부터 저장되고 서빙된다. 이주된 애플리케이션(292)을 클라우드 내에 인스톨하는 테넌트는 여전히 데이터베이스를 구성하고 그의 요구들을 지정할 수 있지만, 데이터베이스는 일반적으로 사용자 제공 스키마 등에 따라, 클라우드 내의 다른 테넌트들에 대한 데이터베이스들을 제공(및 격리)하는 데이터베이스 서비스에 의해 제공된다. 이주 도구 및 애플리케이션에 대한 다른 가능한 변경들의 상세들이 다음에 설명될 것이다.

[0023] 도 6은 이주 도구(320)를 나타낸다. 이주 도구는 하나 이상의 컴퓨터에서 실행되며, 선택된 애플리케이션(322)에 대한 이주 분석을 수행하여, 아마도 선택된 애플리케이션(322)을 변경하고/하거나, 개발자로 하여금 선택된 애플리케이션(322)을 수동으로 변경하는 것을 가능하게 하기 위한 정보를 출력한다. 도구는 각각의 클라우드에 대한 이주 규칙들(326)의 세트들을 갖는 이주 라이브러리(324)를 사용한다. 예를 들어, 한 세트의 이주 규칙들(326)을 포함한다. 사용자가 이주 도구(320)를 사용하고 있을 때, 타겟 클라우드 플랫폼이 선택되며, 이주 도구(320)는 이주 규칙들(326)의 대응하는 세트를 이용하여 이주 출력(328)을 생성한다. 이주 출력(328)은 교정된 소스 코드, 선택된 애플리케이션(322)에 적용될 패치들, 코드, 시맨틱 또는 아키텍처 변경을 원하는 보고서들 또는 그러한 출력들의 조합일 수 있다.

[0024] 도 7은 규칙들 또는 이주 라이브러리(324)의 뷰를 나타낸다. 설명된 바와 같이, 상이한 컴퓨팅 클라우드 플랫폼들에 대한 상이한 이주 규칙 세트들이 존재할 수 있다. 예시적인 규칙 세트(326A)는 코드 규칙들(350), 운영 체제 규칙들(352), 시맨틱 규칙들(354), SQL 또는 데이터베이스 규칙들(356), 인스톨러 규칙들(358) 등을 포함

할 수 있다.

[0025] 코드 규칙들(350)은 <조건><액션> 형태의 규칙들을 포함할 수 있다. 조건은 코드 명령문의 신택스, 포함되거나 배제되어야 하는 특정 라이브러리, 특정 저장 타입 또는 위치, 경로 등을 지정할 수 있다. 공통 코드 패턴들도 지정될 수 있다. 액션들은 변할 수 있다. 일부 액션들은 코드를 변경하거나 사전 정의된 코멘트를 삽입할 수 있다. 다른 액션들은 이주 보고 로그에 출력을 추가할 수 있다. 코드 규칙들(350)은 특정 호출들 또는 메소드들의 세트를 인식하고, 이들을 클라우드 고유 애플리케이션 프로그래밍 인터페이스(API)로 변환할 수 있다. 규칙은 특정 라이선스 서버 또는 라이선스 라이브러리에 대한 호출을 인식할 수 있다. 규칙은 타겟 컴퓨팅 클라우드에서 이용 가능하지 않은 네트워크 서비스(예를 들어, 활성 디렉토리(TM))와 관련된 코드도 인식할 수 있다. 대응하는 수정 액션들, 보고들 및/또는 교정들이 포함될 수 있다.

[0026] 타겟 애플리케이션이 (이주 후에 클라우드 내의 VM 게스트의 형태로) 특정 운영 체제를 위해 형성되는 환경들을 위해 운영 체제 구성 규칙들(352)의 세트가 제공될 수 있다. 일부 예들에서, 운영 체제 규칙들은 운영 체제와 인터페이스하는 애플리케이션 코드와 관련될 수 있다. 컴퓨팅 클라우드가 테넌트로 하여금 특정 운영 체제를 지정 또는 인스톨하게 하는 예들에서, 규칙들은 운영 체제를 직접 검사할 수 있다. 예를 들어, 관련 컴퓨팅 클라우드가 게스트 운영 체제 갱신들을 자동으로 처리하는 경우, 게스트 운영 체제들은 자동 갱신들을 디스에이블하도록 구성되어야 한다. 허가들 또는 특정 사용자 계정들이 변경되거나 추가될 수 있다. 일부 클라우드들은 특정 운영 체제 버전들(릴리스 버전, 32 대 64 비트 버전 등)만을 지원할 수 있으며, 따라서 규칙들은 운영 체제 요구를 식별할 수 있고, 액션들은 운영 체제의 변경, 운영 체제의 업그레이드 또는 이를 행할 필요성에 대한 플래깅(flagging)을 포함할 수 있다. 일부 클라우드들에서는, 일관성을 위해, 운영 체제 시간대 설정치들을 특정 시간대 설정치, 예를 들어 UTC(Coordinated Universal Time)로 설정하는 것이 권장될 수 있는데, 그 이유는 애플리케이션 인스턴스들 또는 VM들이 다수의 지리적 시간대에 걸쳐 실행될 수 있기 때문이다. 다시, 대응하는 수정들 또는 패치들이 규칙들에 포함될 수 있다.

[0027] 시맨틱 규칙들(354)은 타겟 애플리케이션의 아키텍처 또는 설계 양태들을 지정할 수 있다. 예를 들어, 전술한 바와 같이, 일부 애플리케이션 특징들은 컴퓨팅 클라우드에서 쓸모없게 될 수 있다. 시맨틱 규칙들(354)은 부하 균형화기, 스케일링 논리, 데이터 백업 또는 미러링 등을 인식하기 위한 단서들을 지정할 수 있다. 단서들은 구성된 명단, 키워드 인식, 또는 재고 또는 개방 소스 컴포넌트들의 공지된 표시기들로부터 나올 수 있다. 단서들은 또한 코드를 컴파일하거나 코드의 특징들을 분석 또는 프로파일링하는 것을 포함할 수 있는 자동 코드 분석으로부터 나올 수 있다. 인식할 시맨틱 특성들은 (클라우드에 의해 처리될) 재개 논리, 자원 사용 및 장애 극복 논리를 포함할 수 있다. 일부 클라우드들에서는, VM들이 클라우드에 의해 뜻대로 이동(정지 및 재시동)될 수 있으므로, 규칙들은 로컬 저장소에 의존하는 애플리케이션의 부분들을 인식하고, 애플리케이션의 상태를 신뢰성 있게 유지하기 위해 클라우드 기반 영구 저장소의 사용을 권할 수 있다. 일 실시예에서, 규칙은 애플리케이션 컴포넌트의 하나의 인스턴스에 의해 저장된 상태가 클라우드에 의해 자동으로 개시된 새로운 인스턴스들에 의해 인식되어야 한다는 것을 인식할 수 있다. 다른 시맨틱 규칙들은 애플리케이션의 인스턴스가 예를 들어 그의 호스트 VM의 회복시에 그가 인터럽션 이전의 그의 상태와 일치하는 상태를 갖는 것을 확인하는 것을 가능하게 하기 위해 호스트 운영 체제가 언제 슬립 또는 중지 상태(또는 재개)에 들어갔는지를 인식하기 위한 훅들(hooks)을 추가할 수 있다(또는 추가를 제안할 수 있다).

[0028] 저장소 또는 데이터베이스 규칙들(356)은 컴퓨팅 클라우드의 일부인 데이터베이스 서비스의 인스턴스일 수 있는 통상의 데이터베이스 서버로부터의 클라우드 기반 데이터베이스 서버로의 시프팅과 관련된 규칙들을 포함할 수 있다. 이러한 규칙들은 특정 데이터베이스로부터의 클라우드 내의 블로브 저장소, 클라우드 내의 키 값 저장소, 테이블 서비스로부터의 단순 데이터 테이블 등과 같은 다른 형태의 저장소로의 시프팅도 포함할 수 있다. 일반적으로, 전술한 바와 같이, 애플리케이션의 저장 전략을 애플리케이션을 호스팅하는 VM 상의 저장으로부터 클라우드 기반 저장으로 시프트하려고 시도하는 규칙들이 존재할 수 있다. 다른 데이터베이스 규칙들(356)은 특정 SQL 호출, 데이터베이스 미러링 논리를 찾을 수 있다. 데이터베이스 트랜잭션들 동안 더 빈번한 접속 검사를 행하기 위해 접속 규칙이 포함될 수 있다. 예를 들어, 일부 클라우드 기반 데이터베이스 서비스들은 새로운 인스턴스들을 자주 스핀업하고, 애플리케이션의 데이터베이스의 오래된 인스턴스들을 셧다운시킬 수 있으며, 원래의 애플리케이션은 접속이 코드의 범위를 통해 이용 가능하게 유지되는 반면에 클라우드로 이주될 때 접속 검사가 유용한 것으로 가정할 수 있다. 다른 실시예에서, 클라우드 기반 데이터베이스들은 다수의 테이블에 걸치는 트랜잭션들을 보증하지 않을 수 있으며, 규칙은 다수의 테이블과 관련된 SQL 트랜잭션들을 플래깅할 수 있다. 지원되지 않거나 필요하지 않은 SQL 호출들도 인식될 수 있다.

[0029] 인스톨러 규칙들(358)은 타겟 애플리케이션의 인스톨 패키지 포맷을 검사하고, 타겟 클라우드 상의 인스톨과 관

련된 규칙들을 적용할 수 있다. 예를 들어, 컴퓨팅 클라우드가 애플리케이션을 호스팅하는 VM들을 통해 제어를 실행하는 경우, 다양한 인스톨 컴포넌트들이 재배포되는 것이 필요할 수 있고, 원래의 애플리케이션이 존재하는 것으로 가정한 컴포넌트들(예를 들어, 어셈블리들 또는 라이브러리들)이 이주된 애플리케이션의 인스톨 프로세스 내에 실제로 포함되는 것이 필요할 수 있다. 일부 애플리케이션 컴포넌트들의 설정들은 클라우드 기반 환경 내에 인스톨될 때 변경되는 것이 필요할 수 있다. 일 실시예에서는, 전체 인스톨 패키지가 클라우드와 호환되지 않은 것으로서 플래깅될 수 있다. 다른 실시예에서, 인스톨 규칙은 애플리케이션의 인스톨 패키지를 하나의 포맷으로부터 타겟 클라우드와 호환되는 포맷으로 변환할 수 있다. 규칙은 또한 애플리케이션을 인스톨하기 위해 클라우드에 액세스하는 데 필요한 증명서들을 추가할 수 있다. 다시, 대응하는 액션들, 수정 및/또는 권고가 규칙들에 포함될 수 있다.

[0030] 다른 타입의 규칙들도 포함될 수 있다. 일 실시예에서, 이주 라이브러리는 관련 클라우드들에 대한 비용 정보를 포함할 수 있다. 그러한 정보는 클라우드에서 비용이 어떻게 발생하는지 그리고 클라우드 자원들의 다양한 유닛들에 대한 비용들을 설명할 수 있다. 규칙들은 또한 비용 정보에 액세스하여, 타겟 클라우드 내의 애플리케이션의 잠재적 비용들에 대한 분석을 수행할 수 있다. 그러한 분석이 포함될 수 있다. 일 실시예에서, 비용 정보는 타겟 클라우드에서의 라이선싱 권리들 또는 기회들에 대한 정보를 포함할 수 있다. 이러한 규칙들은 적용시에 애플리케이션의 컴포넌트들(예를 들어, 게스트 운영 체제, 데이터베이스 인스턴스 등)에 대한 새로운 라이선싱 배열들, 또는 라이선스 요구들을 커버하고, 따라서 라이선스에 대해 별도로 지불할 필요성을 없애는 클라우드의 운영자(또는 다른 판매자들)로부터의 라이선스 제의들을 찾기 위한 권고를 이주 보고서에 추가할 수 있다.

[0031] 전문한 다양한 규칙들은 한정적이 아니며, 다른 특정 규칙들 및 다른 카테고리의 규칙들이 사용될 수 있다. 더욱이, 규칙은 컴퓨팅 클라우드에서의 요구들 및 선호들에 대한 정보를 나타내는 편리한 형태일 뿐이다. 본 명세서에서 사용되는 바와 같은 "규칙"이라는 용어는 임의의 애플리케이션의 원래 조건 또는 그의 환경, 및 애플리케이션이 특정 컴퓨팅 클라우드에서 실행되는 경우에 관련되는 특정 컴퓨팅 클라우드의 대응하는 양태를 설명하는(임의의 애플리케이션들에 적용 가능한) 임의의 정보를 포함하도록 정의된다. 본 명세서에서 사용되는 바와 같은 "규칙"은 또한 규칙에 의해 지정되는 조건이 타겟 애플리케이션 내에 존재하는 것으로 결정될 때 취해질 수 있는 임의의 액션을 포함하도록 정의되며, 임의의 액션은 사용될 수 있는 패치들 또는 보고서들 또는 다른 정보의 실제 변경, 생성을 포함한다. 따라서, 실제로, 규칙들은 선언 또는 논리 언어에서의 명령문들, 스크립팅 언어를 포함하는 통상의 절차 코드, 컴파일된 코드 등을 포함하는 다양한 형태들을 취할 수 있다. 규칙들의 타입 및 특성은 규칙 세트들에 대응하는 특정 컴퓨팅 클라우드들에 따라 규칙 세트(326)마다 다를 수 있다. 일 실시예에서는, 하나의 규칙 세트(326)만이 존재하며, 이주 라이브러리(324)는 컴퓨팅 클라우드의 하나의 구현만을 위한 것이다. 또 다른 실시예에서, 규칙들은 실행 가능 도구의 일부로서 구현된다.

[0032] 도 8은 이주 도구(320)에 의해 수행되는 프로세스를 나타낸다. 단계 380에서, 이주 도구(320)의 사용자는 컴퓨팅 클라우드로 이주될 타겟 소프트웨어 또는 애플리케이션을 지정한다(도구 및 규칙들은 임의의 애플리케이션에 적용될 수 있도록 설계된다). 단계 382에서, 이주 도구(320)는 타겟 애플리케이션에 액세스한다. 이것은 패키지 포맷을 열거나, 소스 코드 파일들 및 구성 파일들을 판독하거나, VM 이미지를 적재하거나, 애플리케이션을 조사하기 위한 다른 수단을 포함할 수 있다. 이주 도구(320)는 요소들 사이의 관계, 종속성, 관련 파일(예를 들어, 명단 및 구성된 스크립트) 등을 식별할 수 있다. 단계 384에서, 규칙들의 라이브러리로부터 관련 규칙 세트가 로딩된다. 단계 386에서, 애플리케이션의 다양한 컴포넌트들이 파싱되고, 관련 규칙들이 적용된다. 단계 388에서, 출력이 생성된다. 출력은 패치, 코드 수정, 권고 등을 포함하는 전문한 다양한 형태들을 취할 수 있다.

[0033] 도 9는 이주 도구(320)에 의해 사용되는 판독 도구들(400)의 예시적인 세트를 나타낸다. 애플리케이션의 콘텐츠에 액세스하기 위해, 이주 도구(320)는 VM 이미지 적재기(402), 코드 파서(404), 인스톨 패키지 검사기(406), 어셈블리 판독기(408), 스키마 판독기(410), 스크립트 판독기(412) 또는 파서, 컴파일러, 소프트웨어 개발 환경 및/또는 이주되는 애플리케이션의 타입과 관련된 임의의 다른 공지 기술들을 이용할 수 있다. VM 이미지 적재기(402)는 VM 이미지 포맷을 판독하고 이미지를 파일 시스템 내에 적재하도록 구성될 수 있다.

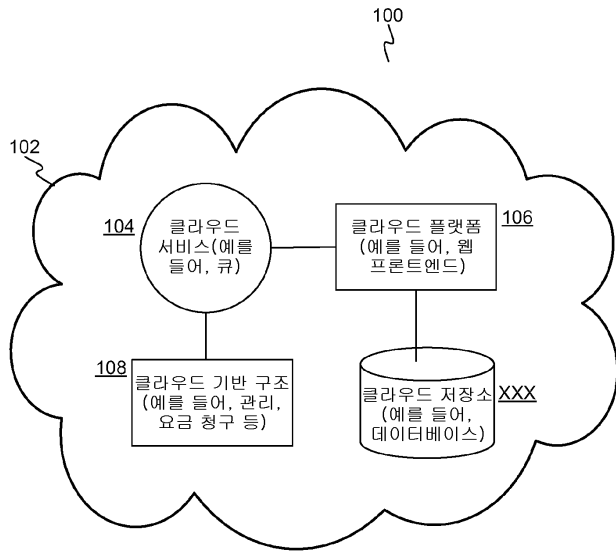
[0034] 결론

[0035] 전문한 실시예들 및 특징들은 휘발성 및 비휘발성 컴퓨터 또는 장치 판독 가능 매체에 저장되는 정보의 형태로 구현될 수 있다. 이것은 적어도 광학 저장 장치(예로서, 콤팩트 디스크 판독 전용 메모리(CD-ROM)), 자기 매체, 플래시 판독 전용 메모리(ROM), 또는 디지털 정보를 저장하는 임의의 현재 또는 미래의 수단과 같은 매체

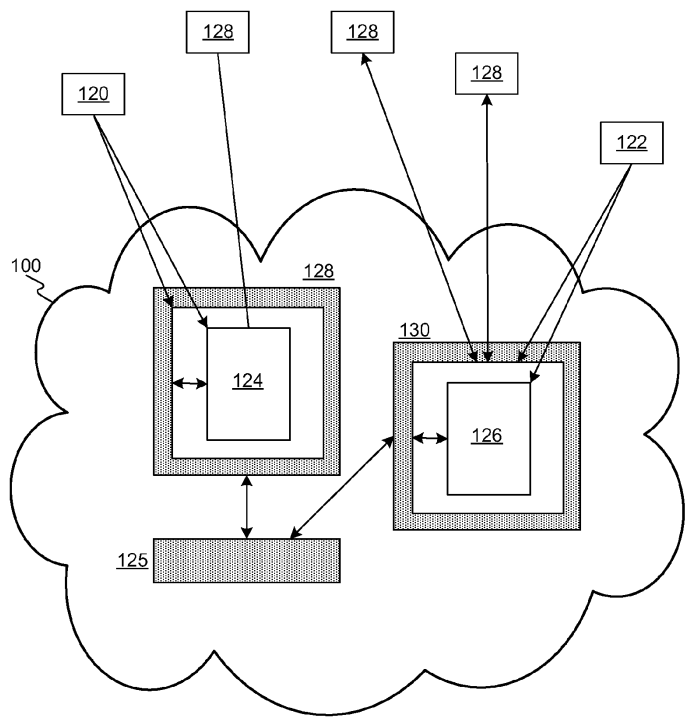
를 포함하는 것으로 간주된다. 저장되는 정보는 머신 판독 가능 명령어(예를 들어, 컴파일된 실행 가능 이진 코드), 소스 코드, 바이트 코드, 또는 컴퓨팅 장치들이 전술한 다양한 실시예들을 수행하는 것을 가능하게 하거나 구성하는 데 사용될 수 있는 임의의 다른 정보의 형태일 수 있다. 이것은 또한 적어도 일 실시예를 수행하는 프로그램의 실행 동안 중앙 처리 유닛(CPU) 명령어들과 같은 정보를 저장하는 랜덤 액세스 메모리(RAM) 및/또는 가상 메모리와 같은 휘발성 메모리는 물론 프로그램 또는 실행 파일이 로딩되고 실행되는 것을 가능하게 하는 정보를 저장하는 비휘발성 매체를 포함하는 것으로 간주된다. 실시예들 및 특징들은 휴대용 장치, 워크스테이션, 서버, 이동 무선 장치 등을 포함하는 임의 타입의 컴퓨팅 장치에서 수행될 수 있다.

도면

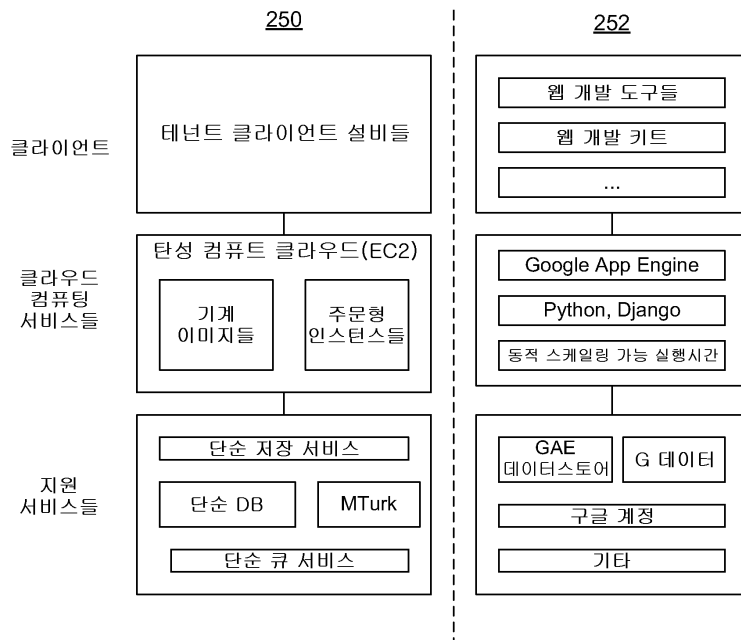
도면1



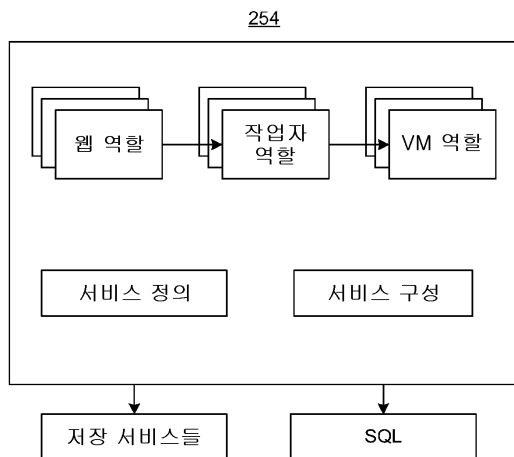
도면2



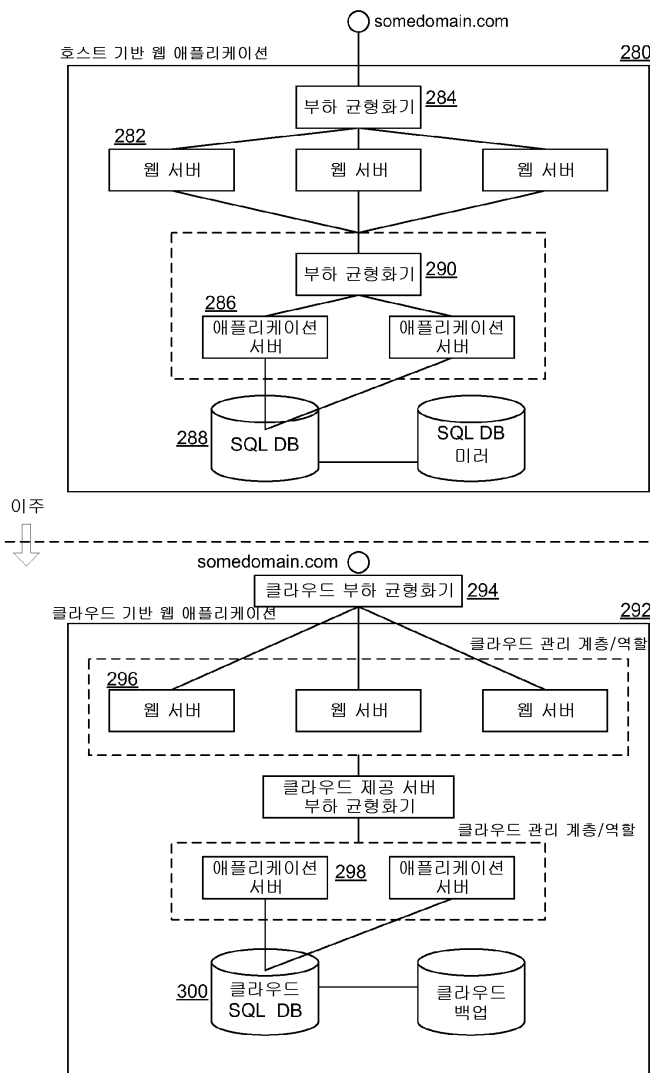
도면3



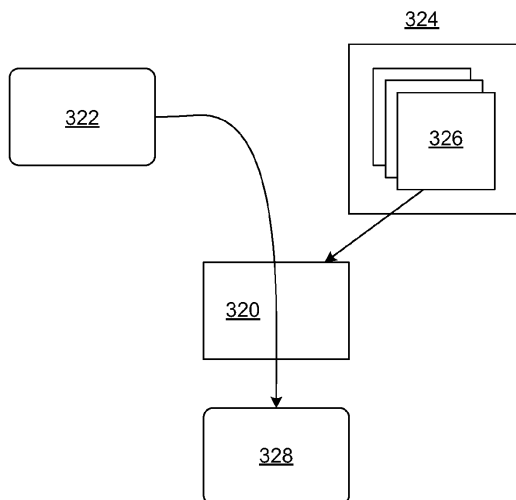
도면4



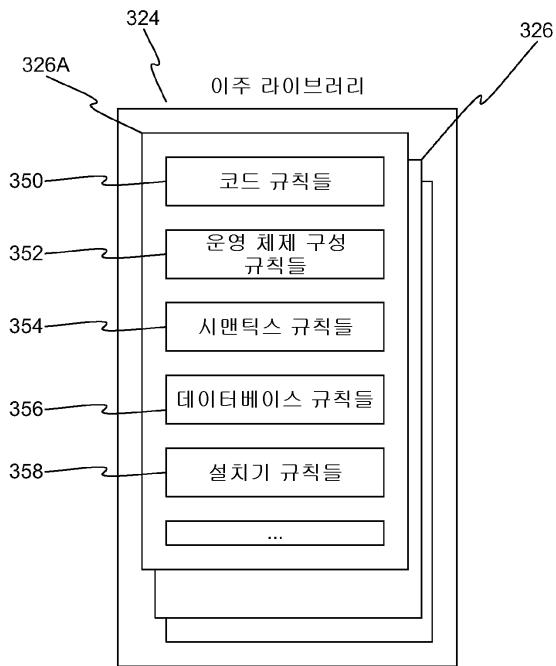
도면5



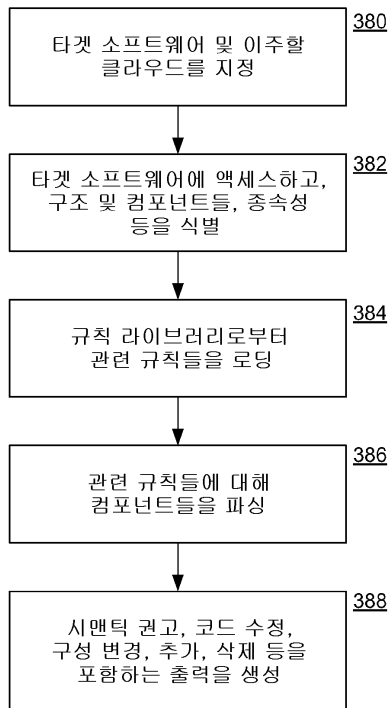
도면6



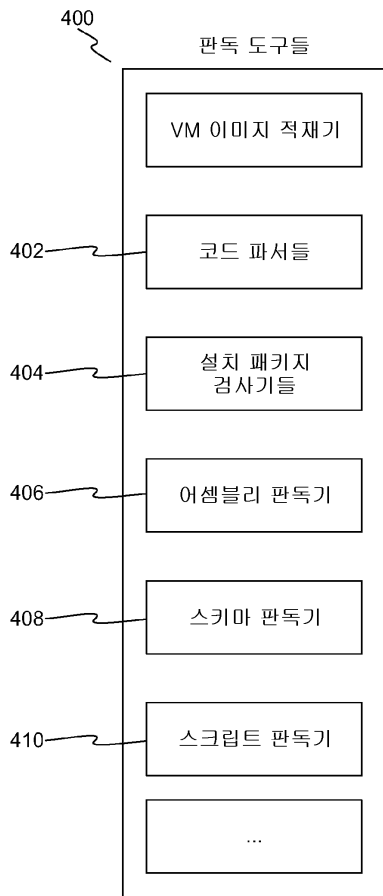
도면7



도면8



도면9



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 16

【변경전】

상기 논-클라우드 자원

【변경후】

상기 논-클라우드 네트워크 자원