

(72) 발명자

커터네크 맥심

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

샌더스 코레이

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

시예드 사드

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

굽타 가우라브

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

필로 아이반 산타 마리아

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

하산 아크람

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

류와스카르 수산트

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

아자드 우메르

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

샤 애쉬쉬

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

펠리거 토드

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

특허청구의 범위

청구항 1

실행 시에, 클라우드-컴퓨팅 네트워크의 패브릭 컨트롤러와 상기 클라우드-컴퓨팅 네트워크에서 실행되는 서비스 애플리케이션 간의 인터랙션을 용이하게 하는 방법을 수행하도록 구현된 컴퓨터-실행 가능 명령어를 갖고 있는 하나 이상의 컴퓨터-판독 가능 매체에 있어서, 상기 방법은

상기 서비스 애플리케이션의 하나 이상의 롤 인스턴스(role instance)를 포함하는 제 1 업데이트 도메인(UD)을 선택하는 단계 - 상기 하나 이상의 롤 인스턴스는 온라인 상태에서 동작함 -,

테넌트 잡(tenant job)의 실행을 위해 상기 제 1 UD를 준비하는 단계 - 상기 제 1 UD의 준비 단계는

(a) 상기 테넌트 잡을 실행하라는 상기 패브릭 컨트롤러의 의도를 상기 서비스 애플리케이션 내의 관리 롤에게 통지하는 단계, 및

(b) 상기 테넌트 잡에 의해 영향을 받는 상기 하나 이상의 롤 인스턴스의 내부 상태가 상기 테넌트 잡에 의해 영향을 받지 않는 상기 서비스 애플리케이션의 부분에 복제되는지 여부의 결정에 대한 상기 관리 롤로부터 응답을 수신하는 단계를 포함함 -,

상기 제 1 UD의 상기 하나 이상의 롤 인스턴스의 오프라인 상태로의 사용 불가를 시작하는 단계, 및

상기 제 1 UD에서 상기 테넌트 잡을 실행하게 하는 단계를 포함하는 컴퓨터-판독 가능 매체.

청구항 2

제 1 항에 있어서,

상기 방법은, 상기 테넌트 잡의 실행이 완료되면, 상기 제 1 UD의 상기 하나 이상의 롤 인스턴스를 상기 온라인 상태로 복원하는 단계를 더 포함하는 컴퓨터-판독 가능 매체.

청구항 3

제 2 항에 있어서,

상기 제 1 UD의 상기 하나 이상의 롤 인스턴스를 상기 온라인 상태로 복원하는 단계는

상기 테넌트 잡에 의한 영향을 받는 상기 하나 이상의 롤 인스턴스가 기능하는지를 확인하는 단계, 및

상기 하나 이상의 롤 인스턴스에서 상기 테넌트 잡의 실행이 완료되었음을 상기 관리 롤에게 통지하여, 상기 서비스 애플리케이션이 상기 하나 이상의 롤 인스턴스의 이용을 재개하게 하는 단계를 포함하는 컴퓨터-판독 가능 매체.

청구항 4

제 3 항에 있어서,

상기 하나 이상의 롤 인스턴스의 오프라인 상태로 사용 불가하게 하는 단계는 상기 제 1 UD 내에서 상기 하나 이상의 롤 인스턴스를 호스팅하는 일련의 노드들을 작동하지 않게 하는 단계를 포함하고, 상기 하나 이상의 롤 인스턴스를 상기 온라인 상태로 복원하는 단계는 상기 제 1 UD 내에서 상기 하나 이상의 롤 인스턴스를 호스팅하는 상기 일련의 노드들을 작동하게 하는 단계를 포함하는 컴퓨터-판독 가능 매체.

청구항 5

제 4 항에 있어서,

상기 일련의 노드들 각각은 상기 서비스 애플리케이션의 상기 하나 이상의 물 인스턴스를 실행할 수 있는 물리적 기계 또는 가상 기계를 나타내며, 상기 물 인스턴스들은 상기 서비스 애플리케이션의 기능을 지원하는 각각의 컴포넌트 프로그램들을 나타내는 컴퓨터-판독 가능 매체.

청구항 6

제 1 항에 있어서,

상기 방법은 상기 테넌트 잡을 실행하기 위한 물 인스턴스들의 제 2 UD를 선택하는 단계를 더 포함하며, 상기 제 1 UD 및 제 2 UD는 상호 배타적인 관계이며, 그 각각은 상기 클라우드-컴퓨팅 네트워크를 통한 테넌트 잡의 전파 시에 개별 단계를 나타내는 컴퓨터-판독 가능 매체.

청구항 7

테넌트 잡을 실행할 때 최우선 순위 잡(high-priority job)을 스케줄링하는 컴퓨터 방법에 있어서, 상기 방법은 서비스 애플리케이션에서 상기 테넌트 잡 실행 지시를 수신하는 단계,

상기 서비스 애플리케이션의 하나 이상의 물 인스턴스를 포함하는 업데이트 도메인(UD)을 식별하는 단계 - 상기 하나 이상의 물 인스턴스는 온라인 상태에서 동작함 -,

상기 테넌트 잡의 실행을 위해 상기 UD를 준비하는 단계,

상기 UD의 상기 하나 이상의 물 인스턴스의 오프라인 상태로의 사용 불가를 시작하는 단계,

최우선 순위 잡 구현 지시를 수신하는 단계 - 상기 최우선 순위 잡은 우선 순위 계획 안에서 상기 테넌트 잡을 선점하기 위해 사전 결정됨 -,

상기 최우선 순위 잡에 상기 테넌트 잡의 배치를 양보하는 단계 - 상기 양보 프로세스는

(a) 상기 하나 이상의 물 인스턴스를 상기 온라인 상태로 복원하는 일련의 축소된 작업들을 실행하도록 상기 서비스 애플리케이션에 지시하는 단계,

(b) 상기 테넌트 잡을 보류하는 단계, 및

(c) 상기 하나 이상의 물 인스턴스에서 상기 최우선 순위 잡의 실행을 시작하는 단계를 포함함 -,

상기 최우선 순위 잡의 실행을 완료하면, 상기 테넌트 잡의 배치를 재개하는 단계 - 상기 재개 프로세스는

(a) 상기 UD의 상기 하나 이상의 물 인스턴스의 상기 오프라인 상태로의 사용 불가를 재시작하는 단계, 및

(b) 상기 하나 이상의 물 인스턴스에서 상기 테넌트 잡의 실행을 허용하는 단계를 포함함 -, 및

상기 테넌트 잡의 실행이 완료되면, 상기 UD의 상기 하나 이상의 물 인스턴스를 상기 온라인 상태로 복원하는 단계를 포함하는 컴퓨터 방법.

청구항 8

제 7 항에 있어서,

상기 테넌트 잡의 실행을 위해 상기 UD를 준비하는 단계는

상기 서비스 애플리케이션 내의 관리 물에게 상기 테넌트 잡을 실행시키려는 의도를 통지하는 단계, 및

상기 테넌트 잡에 의해 영향을 받는 상기 하나 이상의 물 인스턴스의 내부 상태가 상기 테넌트 잡에 의해 영향을 받지 않는 상기 서비스 애플리케이션의 부분에 복제된다고 결정될 경우 긍정적인 응답을 상기 관리 물로부터 수신하는 단계, 또는

상기 테넌트 잡에 의해 영향을 받는 상기 하나 이상의 롤 인스턴스의 내부 상태가 상기 하나 이상의 롤 인스턴스에 국한된다고 결정될 경우 부정적인 응답을 상기 관리 롤로부터 수신하는 단계를 포함하는 컴퓨터 방법.

청구항 9

제 8 항에 있어서,

상기 방법은

상기 관리 롤로부터 상기 긍정적인 응답을 수신하면, 상기 하나 이상의 롤 인스턴스에서 상기 테넌트 잡이 실행 되도록 하는 단계, 및

상기 관리 롤로부터 상기 부정적인 응답을 수신하면,

(a) 상기 테넌트 잡을 실행하라는 지시가 고객으로부터 시작되었을 경우, 상기 내부 상태의 복제를 허용하기 위해서 상기 하나 이상의 롤 인스턴스에서 상기 테넌트 잡의 실행을 지연하는 단계, 및

(b) 상기 테넌트 잡을 실행하라는 지시가 플랫폼으로부터 시작되었을 경우, 상기 하나 이상의 롤 인스턴스에서 상기 테넌트 잡의 실행을 진행하는 단계를 더 포함하는 컴퓨터 방법.

청구항 10

서비스 애플리케이션의 부분들로의 테넌트 잡의 점진적인 배치를 조정하는 방법을 실행하는 컴퓨터 시스템에 있어서,

상기 컴퓨터 시스템은 컴퓨터 저장 매체에 결합된 처리 장치를 포함하고, 상기 컴퓨터 저장 매체는 상기 처리 장치에 의해 실행 가능한 복수의 컴퓨터 소프트웨어 컴포넌트들을 저장하며, 상기 소프트웨어 컴포넌트들은

상기 서비스 애플리케이션의 하나 이상의 롤 인스턴스들 - 상기 롤 인스턴스들은 상기 서비스 애플리케이션의 기능을 지원하는 상기 컴포넌트 프로그램들을 나타냄 -,

상기 테넌트 잡의 배치를 감독하는 테넌트-변경 관리자 - 상기 배치를 감독하는 프로세스는 상기 테넌트 잡의 배치의 지시 전달, 상기 하나 이상의 롤 인스턴스의 오프라인 상태로의 사용 불가 시작, 상기 하나 이상의 롤 인스턴스에서 상기 테넌트 잡의 실행 허용, 및 상기 하나 이상의 롤 인스턴스의 온라인 상태로의 복원 시작을 포함함 -, 및

상기 하나 이상의 롤 인스턴스의 내부 상태를 모니터링하는 상기 서비스 애플리케이션의 관리 롤 - 상기 테넌트 잡의 배치의 지시를 수신하면, 상기 관리 롤은 상기 테넌트 잡에 의해 영향을 받는 상기 하나 이상의 롤 인스턴스의 모니터링된 내부 상태가 상기 테넌트 잡에 의해 영향을 받지 않는 상기 서비스 애플리케이션의 부분에 복제되는지 여부를 결정하고 상기 내부 상태의 가용성의 표시를 상기 테넌트-변경 관리자에게 전달하는 담당을 함 - 을 포함하는 컴퓨터 시스템.

명세서

배경 기술

[0001] 일반적으로, 분산 서비스 애플리케이션은 (다양한 노드들을 통해) 클라우드-컴퓨팅 네트워크에서 호스팅되어 주로 서비스-애플리케이션 컴포넌트들의 중복분(redundancy)을 통해 고가용성(high availability)을 촉진시키는 것을 목적으로 한다. 이런 서비스 애플리케이션은 보통 서비스-애플리케이션 컴포넌트들의 그룹을 포함하는 부분들로 나뉜다. 이런 부분들을 사용하여 서비스 애플리케이션을 호스팅하는 노드들의 업데이트 또는 보수 중에 전체 서비스 애플리케이션이 이용 불가가 되지 않도록 보장하게 된다. 동작 시에, 이런 부분들이 개별적으로 오프라인이 되므로, 다수의 서비스 애플리케이션의 부분들이 계속 온라인으로 동작하게 된다. 따라서, 이상적으로, 사용자는 서비스 애플리케이션의 가용성의 일시적 감소(lapse)를 접하지 않게 된다.

[0002] 현재, 서비스 애플리케이션의 분할은 위치-관련 기준(예컨대, 데이터 센터 내의 공통 노드에 있는 서비스-애플리케이션 컴포넌트들) 또는 기능-관련 기준(한 번에 중단될 수 없는 특정 롤(role)을 실행하는 모든 서비스-애플리케이션 컴포넌트들)에 기반한다. 서비스-애플리케이션 컴포넌트들을 부분들로 그룹화하는 이런 기준은 상태 비보존형(stateless) 애플리케이션(즉, 그 컴포넌트들의 외부에 저장된 데이터에 의해 좌우되는 휘발성 소프트

웨어)에서는 어느 정도 효과가 있다. 그러나, 이런 기준은 서비스-애플리케이션 컴포넌트들에서 지속되는 내부 상태를 유지하는 상태 보존형 애플리케이션에서는 효과가 없다. 즉, 상기의 기준을 사용하면 서비스-애플리케이션 컴포넌트들의 현재 내부 상태를 고려하지 못하게 되어, 이전에 형성된 상태가 오프라인이 될 때 서비스 애플리케이션을 이용하지 못할 수 있다. 따라서, 내부 상태를 유지하는 분산 서비스 애플리케이션들에 대한 고가용성을 가능하게 하는 기술한 문제점을 다루는 기술이 본원에서 소개된다.

발명의 내용

과제의 해결 수단

- [0003] 본 요약은 아래의 상세한 설명에서 추가적으로 설명되는 일련의 개념을 간략화된 형태로 소개하기 위한 것이다. 본 요약은 특허청구된 대상의 핵심적인 특징 또는 필수적인 특징을 밝히기 위한 것이 아니며, 특허청구된 대상의 범위를 결정하는 데 일조하기 위한 것도 아니다.
- [0004] 본 발명의 실시예는 내부 상태를 유지하는 서비스 애플리케이션들 또는 그 부분들에 대한 고가용성을 촉진하는 시스템, 방법, 및 컴퓨터 저장 매체에 관한 것이다. 일반적으로, 이런 “상태 보존형” 서비스 애플리케이션은 클라우드-컴퓨팅 네트워크(이하 “클라우드”)내에서 노드들(예컨대, 물리적 기계 또는 가상 기계)에 걸쳐 분산되는 부분들을 포함하고, 노드들은 하나 이상의 데이터 저장소에 수용된다. 보통, 클라우드는 클라우드의 기본적인 기능을 지원하는 컴퓨터 리소스의 건강 상태(health)의 정비 및 관리를 둘러싼 다양한 임무를 일반적으로 책임지는 패브릭 컨트롤러를 구비한다.
- [0005] 예를 들어, 패브릭 컨트롤러는 상태 보존형 서비스 애플리케이션의 부분들이 속한 노드들을 목표화(targeting) 및 재구성(reconfiguring)을 함으로써 클라우드에서 실행되는 상태 보존형 서비스 애플리케이션들에 대한 업데이트를 지휘하는 책임을 질 수 있다. 이와 같은 업데이트 도중에, 어떤 상태 보존형 서비스 애플리케이션도 실행 불능으로 되지 않도록 하기 위해 패브릭 컨트롤러는 노드들을 목표화할 때 서비스-애플리케이션 부분들의 가용성을 고려한다. 또한, 패브릭 컨트롤러는 서비스-애플리케이션 부분들의 내부 상태를 고려하여 목표화할 노드들을 결정함으로써, 업데이트 중에 어떤 내부 상태도 전체적으로 오프라인이 되지는 않도록 보장하려 한다.
- [0006] 예시적인 실시예에서, 패브릭 컨트롤러는 상태 보존형 애플리케이션이 온라인으로 유지되고 완전히 기능하도록 노드들의 재구성을 스케줄링하기 위해 업데이트를 지휘하는 소프트웨어 컴포넌트(본원에서는 “테넌트(tenant)-변경 관리자”라 함)에 의해 좌우된다. 동작 시에, 테넌트-변경 관리자는 패브릭 컨트롤러와 상태 보존형 서비스 애플리케이션들의 특화된 컴포넌트들 사이에서 인터랙션하는 프로토콜을 이용한다. 이런 특화된 컴포넌트들(본원에서 “관리 롤”이라고 함)은 상태 보존형 서비스 애플리케이션의 롤의 인스턴스들의 경우와 같이, 상태 보존형 서비스 애플리케이션들의 기능 부분들의 내부 상태를 모니터링하도록 상태 보존형 서비스 애플리케이션 내에서 구현된다.
- [0007] 테넌트-변경 관리자에 의해 용이해지는 이와 같은 인터랙션은, 상태 보존형 서비스 애플리케이션 전체의 일정한 고가용성을 유지시키기 위해 롤 인스턴스들의 내부 상태의 가용성을 고려함으로써 상태 보존형 서비스 애플리케이션 내의 변경 동작들의 적절한 시퀀싱을 보장한다. 예를 들어, 내부 상태-가용성의 이러한 고려 대상에는 변경 동작 중에 내부 상태를 추론하고 이 정보를 테넌트-변경 관리자에게 보고하는 관리 롤이 포함된다. 테넌트-변경 관리자는 상태 보존형 서비스 애플리케이션의 특정 롤 인스턴스들에 대한 변경 동작, 또는 “테넌트 잠”을 구현하기 위한 시작-시간을 설정하기 위해 이런 정보를 이용한다. 따라서, 테넌트-변경 관리자와 관리 롤 간의 인터랙션은 분산된 롤 인스턴스들의 개별적인 내부 상태를 고려함으로써 상태 보존형 서비스 애플리케이션의 진정한 가용성을 파악하게 된다.

도면의 간단한 설명

- [0008] 첨부된 도면을 참조하여 본 발명의 실시예가 이하에서 상세하게 설명된다.
- 도 1은 본 발명의 실시예의 구현에 사용하기에 적합한 예시적인 컴퓨팅 환경의 블록도이다.
- 도 2는 데이터 센터의 노드들 간에 분산된 서비스 애플리케이션의 동작을 수용하고 지원하도록 구성된, 본 발명의 실시예의 구현에 사용하기 적합한 데이터 센터를 도시하는 블록도이다.

도 3은 본 발명의 실시예에 따른, 데이터 센터 내에 테넌트 잡을 점진적으로(incrementally) 구현하는 업데이트 도메인을 형성하기 위해 그룹화된 물들의 인스턴스가 있는 예시적인 호스팅 환경을 도시하는 블록도이다.

도 4는 본 발명의 실시예에 따른, 상태 보존형 노드와 상태 비보존형 노드 양쪽 모두에 관한 프로세스 흐름의 그래픽 표현이다.

도 5는 본 발명의 실시예에 따른, 테넌트 잡을 구현하는 개별 동작들을 실행하기 위한 다양한 컴포넌트들을 포함하는 클라우드-컴퓨팅 네트워크를 도시하는 블록도이다.

도 6은 본 발명의 실시예에 따른, 패브릭 컨트롤러와 서비스 애플리케이션 간의 인터랙션을 묘사하는 데이터 센터의 예시적인 토폴로지의 그래픽 표현이다.

도 7은 본 발명의 실시예에 따른, 클라우드-컴퓨팅 네트워크의 패브릭 컨트롤러와 서비스 애플리케이션 간의 인터랙션을 용이하게 하기 위한 전반적인 방법을 도시하는 순서도이다.

도 8은 본 발명의 실시예에 따른, 테넌트 잡을 실행할 때 최우선 순위 잡을 스케줄링하는 전반적인 방법을 도시하는 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0009] 본 발명의 실시예의 대상이 법정 요구조건을 충족시키도록 구체적으로 설명한다. 그러나, 설명 자체가 본 특허의 범위를 제한하고자 하는 것은 아니다. 오히려, 발명가들은 특허청구된 대상이 그 밖의 현재 또는 미래의 기술과 관련하여, 본 특허에서 설명된 것들과 상이한 단계들, 또는 유사한 단계들의 조합을 포함하도록 다른 방식으로 구현될 수 있다는 점을 고려했다.

[0010] 일반적으로, 본 발명의 실시예는 상태 보존형 서비스 애플리케이션들이 고객들에 대한 고가용성을 유지할 수 있게 함으로써 상태 보존형 서비스 애플리케이션의 업데이트와 보수를 지원하기 위한 클라우드-컴퓨팅 네트워크 내의 메커니즘들을 소개한다. 특히, 이러한 메커니즘들은 상태 보존형 서비스 애플리케이션의 부분들(본원에서 “롤 인스턴스들”이라고 함)에 대한 업그레이드, 구성 변경, 스케일링 업 또는 다운, 및 재배포 절차 동안 이런 상태 보존형 서비스 애플리케이션의 지속적인 가용성을 유지할 수 있다. 예시적인 실시예에서, 이런 메커니즘은 상태 보존형 서비스 애플리케이션들과, 상태 보존형 서비스 애플리케이션의 롤 인스턴스들의 라이프 사이클을 관리하는 패브릭 컨트롤러 간의 인터랙션에 의해 좌우된다. 아래에서 상세하게 논의되겠지만, 일례로, 이 메커니즘은 업데이트와 보수를 지능적으로 스케줄링하기 위해 인터랙션하는 패브릭 컨트롤러에 있는 테넌트-변경 관리자 및 상태 보존형 서비스 애플리케이션 내에 있는 관리 롤(들)을 포함한다.

[0011] 본원에서 사용된 바와 같이, “상태 보존형 서비스 애플리케이션”이라는 용어는 롤 인스턴스들 또는 프로그램 컴포넌트들 각각에 상태가 내장된 서비스 애플리케이션들을 일반적으로 지칭한다. 예를 들어, 은행 업무를 실행하도록 특정 온라인 인터페이스가 노출된 금융 기관의 서비스 애플리케이션에서 상태는 고객 데이터(예컨대, 계좌 정보)를 포함할 수 있고, 이는 서비스 애플리케이션의 롤 인스턴스들 안에 유지된다. 즉, 고객 데이터는 원격 위치에 업로드되는 것이 아니라 서비스-애플리케이션 롤 인스턴스들에서 지역적으로 호스팅된다. 따라서, 서비스 애플리케이션이 클라우드-컴퓨팅 네트워크 내의 다양한 노드들에 걸쳐 분산될 때, 고객-제공 계좌 정보, 또는 “상태”도 클라우드-컴퓨팅 네트워크의 노드들에 걸쳐 분산된다. 또한, 서비스 애플리케이션을 가지고 은행 업무를 보는 많은 고객들이 있기 때문에, 각각의 롤 인스턴스는 조금씩 다른 내부 상태를 갖고 있을 수 있다.

[0012] 따라서, 클라우드-컴퓨팅 네트워크의 기저를 이루는 플랫폼의 가용성 및 건강 상태를 알면서, 그 내부에서 실행되는 분산 서비스 애플리케이션들의 내부 상태를 이해하는 것은 정비 및 보수 작업의 타이밍을 스케줄링하는 데 도움이 된다. 예를 들어, 정비 및 복구 작업을 실행할 때, 서비스 애플리케이션의 부분들이 오프라인이 될 수 있다. 이런 부분들을 오프라인이 되게 하는 것은, 특정 내부 상태의 복사본이 온라인에 남아 있지 않으면 전체 서비스 애플리케이션에 부정적인 영향을 미치거나 전체 서비스 애플리케이션을 사용 불가능하게 할 수도 있다. 따라서, 본 발명의 실시예는 각각의 내부 상태의 복사본이 온라인으로 유지되도록 다양한 내부 상태들의 위치를 파악하고 정비 및 복구 작업들을 조정하는 메커니즘을 포함한다.

[0013] 그 결과, 내부-상태 위치의 이해를 통한 정비 및 복구 작업의 조정은 롤 인스턴스들의 내부 상태의 고신뢰도를 얻는 데 일조한다. 일 실시예에서, 특정 상태를 지역적으로 호스팅하는 롤 인스턴스들 중에 어느 하나가 오프라인이 되도록 하기 위해서 조정은 동일한 롤의 복수의 인스턴스들에 걸쳐 특정 상태를 복제하는 것을 포함할 수 있다. 예를 들어, 롤이 현재 데이터 센터에 분산된 다섯 개의 인스턴스들을 가지고 있는 경우, 롤 인스턴스들

중 특정 하나가 고객이 지속하기 원하는 상태를 저장하고 있으면, 특정 롤 인스턴스가 오프라인으로 되기 전에 그 저장된 상태가 다섯 개의 인스턴스들 중 다른 인스턴스(들)에 복사, 전파, 및 유지된다. 이와 같은 방식으로, 다양한 노드들에서의 다양한 롤 인스턴스들에 걸친 저장 상태의 복제는 특정 상태의 신뢰도를 보장하게 된다. 따라서, 정비 및 복구 작업들의 범위 밖에서 특정 롤 인스턴스가 작동하지 않는 경우에도, 다른 롤 인스턴스가 작동하지 않는 롤 인스턴스의 저장된 상태를 지속하게 된다.

[0014] 정비 및 복구 작업을 고려하여 상태를 복제하는 여러 다른 실시예들이 설명되었지만, 본원에 설명된 프로세스를 통해 다른 종류의 적절한 서비스 애플리케이션들이 업데이트될 수 있고, 본 발명의 실시예는 본원에 설명된 상태 보존형 서비스 애플리케이션들에 제한되지는 않는다는 점을 이해하고 인식할 필요가 있다. 예를 들어, 클라우드-컴퓨팅 네트워크의 기저 플랫폼 내에서 테넌트 잡을 실행할 때 비보존형 서비스 애플리케이션도 유사한 방식으로 다뤄질 수 있다. 본원에서 사용된 바와 같이, “비보존형 서비스 애플리케이션”이란 용어는 저장 데이터를 관리할 수 있는 다른 데이터 센터와 같은 외부 엔티티에 그 저장 데이터를 전가하는 클라우드-컴퓨팅 네트워크에 의해 호스팅되는 서비스를 일반적으로 지칭한다. 이러한 데이터 센터는 상이한 물리적 기계들에 대해 상이한 상태들을 유지하고 있지만, 많은 중복본(예컨대, 다수의 기계들에 걸친 다수의 복사본들)를 생성함으로써 상태들을 관리할 수 있다.

[0015] 따라서, 일 양태에서, 본 발명의 실시예는 실행 시에, 클라우드-컴퓨터 네트워크의 패브릭 컨트롤러와 클라우드-컴퓨터 네트워크에서 실행되는 서비스 애플리케이션 간의 인터랙션을 용이하게 하는 방법을 수행하도록 구현된 컴퓨터-실행 가능 명령어를 갖고 있는 하나 이상의 컴퓨터-판독 가능 매체에 관한 것이다. 상기 방법은 서비스 애플리케이션의 한 개 이상의 롤 인스턴스를 포함하는 제 1 업데이트 도메인(UD)을 선택하는 단계를 포함한다. 일반적으로, 롤 인스턴스들은 온라인 상태로 동작하고, 서비스 애플리케이션의 기능을 지원하는 개별 컴포넌트 프로그램들(예컨대, 서비스 애플리케이션의 롤의 단일 복제)을 나타낸다.

[0016] 상기 방법 테넌트 잡(예컨대, 플랫폼으로부터 시작된 업데이트, 고객으로부터 시작된 업데이트, 플랫폼으로부터 시작된 보수, 또는 고객으로부터 시작된 보수)의 실행을 위해 제 1 UD를 준비하는 단계를 포함할 수 있다. 예시적인 실시예에서, 제 1 UD의 준비 단계는 적어도 다음의 단계들, 서비스 애플리케이션 내의 관리 롤에게 테넌트 잡을 실행하려는 패브릭 컨트롤러의 의도를 통지하는 단계, 및 테넌트 잡에 의해 영향을 받는 롤 인스턴스(들)의 내부 상태가 테넌트 잡에 의해 영향을 받지 않는 서비스 애플리케이션의 일부분에 복제되는지 여부의 결정에 대해 관리 롤로부터 응답을 수신하는 단계를 포함한다. 실시예에서, 롤-인스턴스(들) 내부 상태를 검토하고 및/또는 다른 행동을 취하여 관리 롤에 의해 응답이 생성된다. 이런 다른 행동은, 예를 들어, 클라우드-컴퓨팅 네트워크의 인프라 내에 구축된 채널들을 사용한 추가 정보(예컨대, 테넌트의 롤들의 상태) 검색을 포함할 수 있다. 제 1 UD의 롤 인스턴스(들)가 준비되면, 이런 롤 인스턴스(들)의 오프라인 상태로의 사용 불가가 시작된다. 일반적으로, 롤 인스턴스(들)의 사용 불가 단계는 제 1 UD 내에서 하나 이상의 롤 인스턴스를 호스팅하는 일련의 노드들이 작동되지 않게 하는 단계를 포함한다. 롤 인스턴스(들)가 오프라인 상태를 취하면 테넌트 잡이 제 1 UD에서 실행되게 된다.

[0017] 한편, 상기 방법은, 테넌트 잡의 실행이 완료되면 제 1 UD의 롤 인스턴스(들)를 온라인 상태로 복원하는 단계를 포함할 수 있다. 예시적인 실시예에서, 제 1 UD의 인스턴스(들)를 온라인 상태로 복원하는 단계는 적어도 다음의 단계들, 테넌트 잡에 의한 영향을 받는 롤 인스턴스(들)가 기능하는지를 확인하는 단계, 및 롤 인스턴스(들)에서 테넌트 잡의 실행이 완료되었음을 관리 롤에게 통지하여, 서비스 애플리케이션이 태스크들을 실행하기 위해 롤 인스턴스(들)의 이용을 재개하게 하는 단계를 포함한다. 일반적으로, 제 1 UD 내의 롤 인스턴스(들)를 온라인 상태로 복원하는 단계는 롤 인스턴스(들)를 호스팅하는 일련의 노드들이 작동되게 하는 단계를 포함한다. 제 1 UD 내의 롤 인스턴스(들)를 온라인 상태로 복원하면, 상기 방법은 테넌트 잡을 실행하는 롤 인스턴스들의 제 2 UD를 선택함으로써 계속될 수 있다. 통상적으로, 제 1 UD 및 제 2 UD는 상호 배타적인 관계이며, 그 각각은 클라우드-컴퓨팅 네트워크를 통한 테넌트 잡의 전파 시에 개별 단계를 나타낼 수 있다.

[0018] 다른 양태로, 본 발명의 실시예는 테넌트 잡을 실행할 때 최우선 순위 잡을 스케줄링하는 컴퓨터 방법에 관한 것이다. 상기 컴퓨터 방법은 서비스 애플리케이션에서 테넌트 잡을 실행하라는 표시를 수신하고 서비스 애플리케이션의 하나 이상의 롤 인스턴스를 포함하는 업데이트 도메인(UD)을 식별하는 논리적인 단계의 실행을 포함한다. 이 때, 롤 인스턴스(들)는 온라인 상태에서 동작한다. 테넌트 잡의 실행을 위해 UD가 준비되고, 이어서 UD의 인스턴스(들)가 오프라인 상태로 사용 불가해진다. 예시적인 실시예에서, 테넌트 잡의 실행을 위해 UD를 준비하는 프로세스는 서비스 애플리케이션 내의 관리 롤에게 테넌트 잡을 실행시키려는 의도를 통지하는 단계, 및 관리 롤로부터 긍정적인 응답을 수신하거나 부정적인 응답을 수신하는 단계를 포함하는 다양한 논리 단계를 포

함한다.

- [0019] 통상적으로, 관리 롤이 테넌트 잡에 의해 영향을 받는 롤 인스턴스(들)의 내부 상태가 테넌트 잡에 의해 영향을 받지 않는 서비스 애플리케이션의 일부분에 복제된다는 결정을 하면 긍정적인 응답이 수신된다. 관리 롤로부터 긍정적인 응답을 테넌트-변경 관리자(패브릭 컨트롤러 내에서 동작함)가 수신하면, 테넌트 잡이 롤 인스턴스(들)에서 실행될 것이다. 대신에, 관리 롤이 테넌트 잡에 의해 영향을 받는 롤 인스턴스(들)의 내부 상태가 이런 롤 인스턴스에 국한된다는 결정을 하면 부정적인 응답이 수신된다. 테넌트-변경 관리자가 관리 롤로부터 부정적인 응답을 수신하면, 많은 작업들이 일어날 수 있다. 일 실시예에서, 테넌트 잡을 실행하기 위한 표시가 고객으로부터 시작되었으면, 내부 상태를 복제하기 위해 롤 인스턴스(들)에서의 테넌트 잡의 실행이 지연될 수 있다. 다른 실시예에서, 테넌트 잡을 실행하기 위한 표시가 플랫폼으로부터 시작되었으면, 롤 인스턴스(들)에서 테넌트 잡의 실행이 진행된다.
- [0020] 추후에, 최우선 순위 잡을 구현하라는 표시를 수신한다. 본원에서 사용되는 바와 같이, “최우선 순위” 라는 용어는 제한하는 것으로 여겨지지는 않으며, 우선 순위 계획 안에서 테넌트 잡을 선점하기(preempt) 위해 사전 결정된 임의의 잡을 나타낼 수 있다. 최우선 순위 잡을 구현하라는 표시를 수신하면, 최우선 순위 잡에 테넌트 잡의 배치를 양보한다. 예시적인 실시예에서, 양보 프로세스는 롤 인스턴스(들)를 온라인 상태로 복원하는 일련의 축소된(truncated) 작업들을 실행하도록 서비스 애플리케이션에 지시하는 단계, 테넌트 잡을 보유하는 단계, 및 롤 인스턴스(들)에서 최우선 순위 잡의 실행을 시작하는 단계를 포함하는 다양한 논리 단계를 포함한다.
- [0021] 최우선 순위 잡의 실행을 완료하면, 테넌트 잡의 배치가 재개된다. 예시적인 실시예에서, 재개 프로세스는 UD의 롤 인스턴스(들)의 오프라인 상태로의 사용 불가를 재시작하는 단계, 및 롤 인스턴스(들)에서 테넌트 잡의 실행을 허용하는 단계를 포함하는 다양한 논리 단계를 포함한다. 테넌트 잡의 실행이 완료되면, UD의 롤 인스턴스(들)는 온라인 상태로 복원될 수 있다. 예시적인 실시예에서, 롤 인스턴스(들)를 온라인 상태로 복원하는 프로세스는 테넌트 잡에 의해 영향을 받는 롤 인스턴스(들)가 기능하는지를 확인하는 단계, 및 롤 인스턴스(들)에서 테넌트 잡의 실행이 완료되었음을 관리 롤에게 통지하여, 서비스 애플리케이션이 다양한 태스크들을 실행하기 위해 롤 인스턴스(들)의 이용을 재개하게 하는 단계를 포함하는 다양한 논리 단계들을 포함한다.
- [0022] 또 다른 양태로, 본 발명의 실시예는 서비스 애플리케이션의 부분들에서 테넌트 잡의 배치를 조정하는 방법을 실행하는 컴퓨터 시스템에 관한 것이다. 일반적으로, 컴퓨터 시스템은 컴퓨터 저장 매체에 결합된 처리 장치를 포함하고, 컴퓨터 저장 장치는 처리 장치에 의해 실행 가능한 복수의 컴퓨터 소프트웨어 컴포넌트들을 저장한다. 처음에, 컴퓨터 소프트웨어 컴포넌트들은 서비스 애플리케이션의 하나 이상의 롤 인스턴스, 테넌트-변경 관리자, 관리 롤, 호스트 에이전트, 유선 프로토콜, 스케줄링 컴포넌트, 및 노드-상태 드라이버를 포함한다. 실시예에서, 롤 인스턴스들은 서비스 애플리케이션의 기능을 지원하는 컴포넌트 프로그램들을 나타낸다. 테넌트-변경 관리자는 통상적으로 테넌트 잡의 배치를 지시하도록 구성된다. 배치 지시 프로세스는 일반적으로 테넌트 잡의 배치의 표시 전달, 롤 인스턴스(들)의 오프라인 상태로의 사용 불가 시작, 롤 인스턴스(들)에서 테넌트 잡의 실행 허용, 및 롤 인스턴스(들)의 온라인 상태로의 복원 시작을 포함한다.
- [0023] 서비스 애플리케이션의 컴포넌트인 관리 롤은 통상적으로 롤 인스턴스(들)의 내부 상태를 모니터링하도록 구성된다. 실시예에서, 테넌트 잡의 배치의 표시를 수신하면, 관리 롤은 테넌트 잡에 의해 영향을 받는 롤 인스턴스(들)의 모니터링된 내부 상태가 테넌트 잡에 의해 영향을 받지 않는 서비스 애플리케이션의 일부분에 복제되는지를 결정하는 담당을 한다. 또한, 관리 롤은 내부 상태의 가용성의 표시를 테넌트-변경 관리자에 전달하도록 구성될 수 있다.
- [0024] 통상적으로 호스트 에이전트는 롤 인스턴스(들)를 호스팅하는 노드에 있다. 동작 시에, 호스트 에이전트는 유선 프로토콜을 통해서 서비스 애플리케이션에 롤 인스턴스(들)의 예상 상태를 알린다. 본원에서 사용된 바와 같이, “예상 상태”란 용어는 일반적으로 롤 인스턴스들이 오프라인 상태 또는 온라인 상태를 추정하는 목표를 나타낸다. 스케줄링 컴포넌트는 일반적으로 우선 순위 계획에 따라 최우선 순위 잡에 의해 테넌트 잡이 중단되게 하도록 구성된다. 또한, 노드-상태 드라이버는 일반적으로 롤 인스턴스(들)를 오프라인 상태로 이용 불가능하게 하고, 테넌트 잡을 실행하고, 테넌트-변경 관리자의 지시 하에 롤 인스턴스(들)를 온라인으로 복원하도록 구성된다.
- [0025] 앞서 언급한 바와 같이, 서비스 애플리케이션의 모든 롤 인스턴스들을 동시에 오프라인이 되게 하는 것이 아니라, 테넌트 잡들을 보통 단계 별로 나누어 서비스 애플리케이션의 롤 인스턴스들의 그룹에서 점진적으로 실행되게 한다. 따라서, 본 발명의 실시예는 호스팅 환경 또는 데이터 센터 내에서의 업데이트 도메인의 식별에 관한 것이다. 본원에서 이용된 바와 같이, “업데이트 도메인”이라는 용어는 제한하고자 하는 것이 아니며, 일반적

으로 서비스 애플리케이션의 하나 이상의 롤 인스턴스를 호스팅하며 동시에 오프라인이 될 수 있는 일련의 노드들(예컨대, 물리적 기계 또는 가상 기계)의 서술을 지칭한다. 특정 실시예에서, 업데이트 도메인으로 설명되는 일련의 노드들은 동시에 이용 불가능해질 수 있는 노드들을 나타내며(예컨대, 소프트웨어를 설치하기 위해서 이 노드들을 의도적으로 오프라인이 되게 함), 한편 이 노드들에 의존하는 서비스 애플리케이션들은 이용 가능하게 유지되도록 보장된다. 즉, 일련의 노드들 상에서 실행되는 롤 인스턴스들이 업데이트 도메인 밖에 있는 복제본을 갖기 위해서, 테넌트 잠의 실행 시에, 업데이트 도메인의 일련의 노드들은 통상적으로 동시에 기능하지 못하게 될 수 있는 한정된 수의 노드들을 포함한다. 따라서, 서비스 애플리케이션의 소유자에게 서비스 애플리케이션의 최소 개수의 롤 인스턴스들만이 동시에 동작하지 않도록 보장하기 위해 업데이트 도메인이 설정된다.

[0026] 업데이트 도메인의 구축은 일반적으로 호스팅 환경이 부과하는 다양한 규칙에 따른다. 예를 들어, 공통 업데이트 도메인에 연결될 수 있는 롤 A의 많은 인스턴스들에 제약이 있을 수 있다. 또는, 롤 A의 인스턴스들을 갖고 있는 공통 업데이트 도메인에 연결될 수 있는 롤 B의 많은 인스턴스들에 대한 제약이 존재할 수 있다. 일반적으로, 본원에서 사용된 바와 같이, “롤들”은 서비스 애플리케이션의 기능적인 부분의 템플릿 설명을 제공한다. 예를 들어, 온라인-쇼핑 서비스 애플리케이션의 경우에, 롤 A(구입 주문을 받기 위해 제 1 GUI를 제공하는 태스크가 부여됨), 롤 B(구입 주문을 실행하는 태스크가 부여됨), 및 롤 C(구입 주문의 상태를 전달하기 위해 제 2 GUI를 제공하는 태스크가 부여됨)의 세 개의 롤들로 각각 나뉘어지는 세 개의 코어 태스크가 있을 수 있다. 예시적인 실시예에서, 각각의 롤은 서비스 애플리케이션의 특정 종류의 프로그램 컴포넌트를 나타낸다. 통상적으로, 서비스 모델은 데이터 센터 내에서 각각의 롤의 인스턴스를 얼마나 많이 배치하는지를 기술하고, 각각의 인스턴스는 특정 종류의 프로그램 컴포넌트, 또는 롤의 복제본이다. 즉, 각각의 롤은 각각의 종류의 프로그램 컴포넌트의 인스턴스들의 모음을 나타내고, 서비스 애플리케이션은 기능을 수행하는 많은 종류의 프로그램 컴포넌트들을 가질 수 있다.

[0027] 롤들은 종종 그에 적용되는 구성 설정을 갖고 있다. 일례로, 롤의 구성 설정은 롤의 모든 인스턴스가 공유하는 공동 설정을 포함할 수 있다. 다른 예로, 구성 설정은 롤의 각각의 인스턴스에 특정된 개별 설정을 포함할 수 있다. 이런 개별 설정은 롤의 인스턴스에 저장되는 상태를 포함할 수 있고, 여기서 상태는 그 롤 인스턴스에 특수하다. 상기에서 언급한 온라인-쇼핑 서비스와 관련하여, 롤 B의 인스턴스에 저장된 상태는 특정 클라이언트에 의한 특정 구입 주문의 세부 사항일 수 있다.

[0028] 본 발명의 실시예들의 개관을 간략하게 설명하였으며, 본 발명의 실시예들을 구현하기에 적합한 예시적인 운영 환경이 아래에서 설명된다.

[0029] **운영 환경**

[0030] 특히, 처음에는 도 1을 참조하면, 본 발명의 실시예를 구현하기 위한 예시적인 운영 환경이 도시되고, 일반적으로 컴퓨팅 장치(100)로 지칭된다. 컴퓨팅 장치(100)는 적합한 컴퓨팅 환경의 일례에 불과하며, 본 발명의 사용 또는 기능의 범위에 관하여 어떠한 제한을 제시하려고 의도하지 않는다. 컴퓨팅 장치(100)는 도시된 컴포넌트들 중 임의의 하나 또는 그 컴포넌트들의 임의의 조합과 관련하여 어떠한 의존성 또는 요건을 갖는 것으로서 해석되어서는 안 된다.

[0031] 본 발명은 컴퓨터, 또는 PDA(personal data assistant)나 다른 핸드헬드 장치 등의 다른 기계에 의해 실행되는 프로그램 모듈을 비롯한 컴퓨터-실행가능 명령어를 포함하는 컴퓨터 코드 또는 기계-사용 가능 명령어의 일반적인 컨텍스트에서 설명될 수 있다. 일반적으로, 루틴, 프로그램, 객체, 컴포넌트, 데이터 구조 등을 포함하는 프로그램 모듈은 특정 태스크를 수행하거나 특정 추상 데이터 유형을 구현하는 코드를 말한다. 본 발명은 핸드헬드 장치, 소비자 가전, 범용 컴퓨터, 특수 컴퓨팅 장치 등을 포함하는 다양한 시스템 구성에서 실시될 수 있다. 또한, 본 발명은 통신 네트워크를 통해 연결되어 있는 원격 처리 장치들에 의해 태스크가 수행되는 분산 컴퓨팅 환경에서 실시될 수도 있다.

[0032] 도 1을 참조하면, 컴퓨팅 장치(100)는 메모리(112), 하나 이상의 프로세서(114), 하나 이상의 프레젠테이션 컴포넌트(116), 입/출력(I/O) 포트(118), 입/출력 컴포넌트(120) 및 예시적인 전원(122)을 직접 또는 간접적으로 연결하는 버스(110)를 포함한다. 버스(110)는 (주소 버스, 데이터 버스 또는 이들의 조합과 같은) 하나 이상의 버스일 수 있는 것을 나타낸다. 도 1의 각종 블록은 명확성을 위해 선으로 도시되어 있지만, 실제로는, 다양한 컴포넌트의 윤곽을 그리는 것이 그렇게 명확하지는 않으므로, 비유적으로, 더 정확하게는 선은 희색이고 흐릿한 것이다. 예를 들어, 디스플레이 장치와 같은 프레젠테이션 컴포넌트가 I/O 컴포넌트인 것으로 생각할 수 있다. 또한, 프로세서는 메모리를 갖는다. 본 발명가들은 이것이 기술의 본질임을 인식하며, 도 1의 다이어그램이 단지 본 발명의 하나 이상의 실시예에 관련하여 사용될 수 있는 예시적인 컴퓨팅 장치의 도시일 뿐임을 반복한다.

"워크스테이션", "서버", "랩탑", "핸드헬드 장치" 등과 같은 카테고리들은 모두 도 1의 범주 내에서 고려되며 "컴퓨팅 장치"로 참조되기 때문에, 이런 사이에 구분을 짓지 않는다.

[0033] 컴퓨팅 장치(100)는 통상적으로 각종 컴퓨터-관독 가능 매체를 포함한다. 컴퓨터-관독 가능 매체는 컴퓨팅 장치(100)가 액세스할 수 있는 사용 가능한 임의의 매체일 수 있으며, 휘발성 및 비휘발성 매체, 이동식 및 비이동식 매체를 포함한다. 예를 들어, 그러나 제한 없이, 컴퓨터-관독 가능 매체는 컴퓨터 저장 매체 및 통신 매체를 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터-관독 가능 명령어, 데이터 구조, 프로그램 모듈 또는 그 밖의 데이터 등의 정보를 저장하기 위한 임의의 방법 또는 기술로 구현되는 휘발성 및 비휘발성 매체, 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 원하는 정보를 저장하는 데 사용되고 컴퓨팅 장치(100)가 액세스 가능한 임의의 기타 매체를 포함하지만, 이에 제한되는 것은 아니다. 통신 매체는 통상적으로 컴퓨터-관독 가능 명령어, 데이터 구조, 프로그램 모듈 또는 반송파 또는 기타 전송 메커니즘과 같은 변조 데이터 신호(modulated data signal)의 기타 데이터를 구현하고, 임의의 정보 전달 매체를 포함한다. "변조 데이터 신호"라는 용어는, 신호의 정보를 인코딩하도록 그 신호의 특성들 중 하나 이상을 설정 또는 변경시킨 신호를 의미한다. 예를 들어, 그러나 제한 없이, 통신 매체는 유선 네트워크 또는 직접 배선 연결과 같은 유선 매체, 그리고 음향, RF, 적외선, 기타 무선 매체와 같은 무선 매체를 포함한다. 또한, 상기의 임의의 조합은 컴퓨터-관독 가능 매체의 범위 안에 포함되어야 한다.

[0034] 메모리(112)는 휘발성 및/또는 비휘발성 메모리 형태의 컴퓨터-저장 매체를 포함한다. 메모리는 이동식, 비이동식, 또는 이들의 조합일 수 있다. 예시적인 하드웨어 장치는 고체 상태 메모리, 하드 드라이브, 광 디스크 드라이브 등을 포함한다. 컴퓨팅 장치(100)는 메모리(112) 또는 I/O 컴포넌트(120)와 같은 다양한 엔티티로부터 데이터를 관독하는 하나 이상의 프로세서를 포함한다. 프레젠테이션 컴포넌트(들)(116)는 사용자 또는 다른 장치에 데이터 표시를 제공한다. 예시적인 프레젠테이션 컴포넌트는 디스플레이 장치, 스피커, 인쇄 컴포넌트, 진동 컴포넌트 등을 포함한다.

[0035] 컴퓨팅 장치(100)는 I/O 포트(118)를 통해 I/O 컴포넌트들(120)을 포함한 다른 장치에 논리적으로 연결되고, I/O 컴포넌트들의 일부는 내장될 수 있다. 예시적인 컴포넌트로는 마이크로폰, 조이스틱, 게임 패드, 위성 안테나, 스캐너, 프린터, 무선 장치 등을 들 수 있다.

[0036] **데이터 센터의 아키텍처**

[0037] 이제 도 2에서, 서비스 모델에 따른 서비스 애플리케이션의 컴포넌트 프로그램들, 또는 물 인스턴스들의 동작을 수용하고 지원하도록 구성된 데이터 센터(200)를 도시하는, 본 발명의 실시예에 따른 블록도가 도시된다. 도 2에 도시된 데이터 센터(200)는 호스팅 환경의 적절한 일부분의 일례에 불과하며 본 발명의 실시예의 사용 또는 기능의 범위에 관한 임의의 제한을 제시하기 위한 것이 아님을 당업자라면 이해하고 인식할 것이다. 데이터 센터(200)는 도시된 리소스들 중 임의의 하나 또는 리소스들의 임의의 조합과 관련하여 어떠한 의존성 또는 요건을 갖는 것으로서 해석되어서는 안 된다. 또한, 도 2의 각종 블록은 명확성을 위해 선으로 도시되어 있지만, 실제로는, 다양한 컴포넌트의 윤곽을 그리는 것이 그렇게 명확하지는 않으므로, 비유적으로, 더 정확하게는 선은 회색이고 흐릿한 것이다.

[0038] 데이터 센터(200)는 네트워크 클라우드(240)를 통해 상호 연결된 다양한 리소스들을 포함한다. 본원에 설명된 바와 같이, 이런 리소스는 소프트웨어 컴포넌트들(예컨대, 패브릭 컨트롤러(295))은 물론 랙 A(205), 랙 B(210), 랙 C(215), 탭 랙 스위치(230), 파워 스트립(235), 서버(270), 컴퓨팅 장치(280), 업데이트 도메인(251), 및 업데이트 도메인(252)과 같은 유형(tangible) 하드웨어 요소들을 포함할 수 있다. 네트워크 클라우드(240)는 다양한 물리적 리소스들에 분산되어 배치될 수 있는 서비스 애플리케이션들의 인스턴스들이 그들 간의 통신을 구축하기 위해 다른 인스턴스의 위치를 인식하도록 이런 리소스를 상호 연결한다. 또한, 네트워크 클라우드(240)는 서비스 애플리케이션의 인스턴스들, 및 서비스 모델이 필요로 하는 기타 임의의 요소들을 연결하는 채널 상의 이런 통신을 용이하게 한다. 네트워크는, 제한 없이, 하나 이상의 LAN(local area network) 및/또는 WAN(wide area network)을 포함할 수 있다. 이런 네트워킹 환경은 사무실, 전사적 컴퓨터 네트워크, 인터넷, 및 인터넷에서 흔하다. 따라서, 네트워크는 본원에서 더 이상 설명되지 않는다.

[0039] 일반적으로, 데이터 센터(200)는 컴포넌트 프로그램들을 실행하는 처리 장비에서 독립적으로 실행되는 서비스 애플리케이션의 복수의 컴포넌트 프로그램들, 또는 물 인스턴스들을 지닌 분산 시스템을 수용하고, 이들은 네트워크 클라우드(240)를 통해 상호 연결된다. 때때로, 이런 계산 리소스들, 또는 노드들은 다양하게 업데이트될

수 있다. 업데이트 도메인들(251 및 252)은 일반적으로 서비스 애플리케이션의 롤 인스턴스들이 데이터 센터(200)의 리소스들에 분산되고 기저 플랫폼의 업데이트 중에 모두가 동시에 오프라인이 되지는 않을 것이라는 보장 집합을 제공한다. 따라서, 서비스 애플리케이션의 동작 시에 일관성이 보존된다.

[0040] 데이터 센터(200)의 구성으로 돌아와서, 서버(270), 컴퓨팅 장치(280), 오류(fault) 도메인(251 및 252), 및 서로 동작 가능하게 연결된 블레이드들(예컨대, 블레이드 A(260), 블레이드 B(261), 블레이드 C(262))을 갖고 있는 랙 A(205), B(210), 및 C(215) 각각은, 예를 들어, 도 1과 관련하여 기술한 컴퓨팅 장치(100)와 같은 임의의 종류의 컴퓨팅 장치일 수 있다. 예를 들어, 그러나 제한 없이, 서버(270), 컴퓨팅 장치(280), 블레이드 A(260), 블레이드 B(261), 및 블레이드 C(262) 각각은 개인용 컴퓨터, 데스크탑 컴퓨터, 랩탑 컴퓨터, 핸드헬드 장치, 모바일 핸드셋, 소비자 가전 등일 수 있다. 실시예에서, 데이터 센터(200)는 상기에 열거된 많은 물리적 리소스들을 포함할 수 있고, 서비스 애플리케이션을 실행할 수 있는 다양한 컴퓨팅 장치들, 또는 기타 기계들을 포함할 수 있다.

[0041] 또한, 상기의 물리적 리소스들 중 한 개 이상은 데이터 센터(200)의 유선 또는 무선 네트워크 경로를 통해 서비스 애플리케이션을 배치, 액세스, 및 관리하기 위해 그 곳에 위치한 패브릭 컨트롤러(295)를 호스팅할 수 있다. 그러나, 본 발명의 실시예는 도 2에 도시된 이런 물리적 리소스들에서의 구현에 제한되지는 않고 실시예의 범위 내에서 임의의 여러 다른 종류의 컴퓨팅 장치들 및 장비에서 구현될 수 있음을 알 필요가 있다. 즉, 데이터 센터(200)의 도시된 리소스들은 단지 논의 목적을 위한 예시적인 구성을 묘사하고 있으므로, 따라서, 컴퓨팅 업계에 알려진 리소스들의 임의의 적절한 레이아웃이 사용될 수도 있고 본 발명에서 고려되기도 한다.

[0042] 특히, 데이터 센터(200)의 예시적인 리소스들은 업데이트 도메인의 개념을 소개하는 기능을 한다. 상기에서 논의한 바와 같이, 업데이트 도메인은 데이터 센터(200) 내에서 테넌트 잡의 실행 시에 이용 불가능하게 되는 노드들의 모음, 또는 집합을 나타낸다. 상기에서 논의한 바와 같이, 노드란 블레이드(예컨대, 블레이드 A(260), 블레이드 B(261), 블레이드 C(262)), 컴퓨터(예컨대, 컴퓨팅 장치(280)), 기계(예컨대, 서버(270)), 또는 데이터 센터(200)의 서비스 애플리케이션의 컴포넌트 프로그램들, 또는 인스턴스들을 실행시킬 수 있는 임의의 기타 장치를 지칭할 수 있다. 따라서, 업데이트 도메인을 추상화하는 이점은 테넌트 잡의 어느 한 단계의 실행 시에 데이터 센터 내에서 어떤 리소스 그룹들이 함께 오프라인이 되는지와, 다른 리소스들이 오프라인이 되는 것과는 어떻게 구분되는지를 이해하게 되는 점이다.

[0043] 다양한 다른 구성의 데이터 센터(200) 및 업데이트 도메인들(251 및 252)을 설명하였지만, 당업자라면 노드들에서 실행되는 톨들의 신원에 기반하여 일련의 노드들을 그룹화하는 다른 종류의 적절한 업데이트 도메인들을 추상화할 수 있고, 본 발명의 실시예들이 본원에 도시된 업데이트 도메인들(251 및 252)에 제한되지 않음을 이해하고 인식할 필요가 있다.

[0044] **서비스 애플리케이션의 업데이트 도메인**

[0045] 이제 도 3으로 가서, 예시적인 호스팅 환경(300)의 그래픽 표현이 본 발명의 실시예에 따라 도시된다. 특히, 호스팅 환경(300)은 패브릭 컨트롤러(295), 서비스 애플리케이션(350), 및 패브릭 컨트롤러(295)와 서비스 애플리케이션(350)의 동작을 지원하는 데이터 센터(200)를 포함한다. 상기에서 논의된 바와 같이, 데이터 센터(200)는 서비스 애플리케이션(350)의 롤 A, B, 및 C의 중복본에 부분적으로 기반하는 업데이트 도메인들(305 및 310)로 구성된 리소스들을 포함하고, 톨들은 인스턴스들(311, 312, 313, 321, 322 및 323)에 위치한다. 일반적으로, 업데이트 도메인들(305 및 310) 각각은 업데이트 계획, 또는 “잡 목록”에 따라 동시에 이용 불가능해지도록 의도적으로 유도된 하나 이상의 노드를 서술한다. 실시예에서, 잡 목록(도 6의 도면 부호(610) 참조)은 예정된 및/또는 계류 중인(pending) 테넌트 잡들을 우선 순위에 따라 조직한다. 아래에서 보다 상세하게 설명되는 것처럼, 서비스 애플리케이션(350)이 다가오는 테넌트 잡을 수신할 준비를 할 수 있도록 잡 목록의 적어도 일부가 서비스 애플리케이션(350)의 관리 롤에 노출된다.

[0046] 본원에 사용된 대로, “테넌트 잡”이라는 용어는 패브릭 컨트롤러(295)에 의해 서비스 애플리케이션(350)에 실행되는 변경 동작들의 불가변의 집합(immutable set)을 지칭한다. 이런 변경 동작들은 서비스 애플리케이션(350)의 가용성에 영향을 미칠 수 있다. 실시예에서, 호스팅 환경(300)에 의해 정의된 일정 개수의 테넌트-잡 유형이 현재 존재한다. 그러나, 이런 잡 유형은 장차 확대될 수 있다. 서비스 애플리케이션(350)에 중점을 둔 예시적인 테넌트 잡은 서비스-애플리케이션 업데이트 잡(예컨대, 정비 행위), 및 서비스-애플리케이션 코드 수정(예컨대, 업그레이드 행위 또는 새로운 버전 생산), 서비스-애플리케이션 토폴로지 변경, 서비스-애플리케이션 구성 변경, 재부팅을 이용하는 노드(320)의 가상 기계(VM) 내에서 서비스-애플리케이션 게스트 OS 변경(예컨대, 정지-및-시작 동작), 재부팅 및 재포장(repave)을 이용하는 서비스-애플리케이션 보수(예컨대, 게스트 OS의

변경 등록)를 포함한다. 일반적으로 다른 프로세스에 의해 VM이 복원되지 않을 때 보수가 시작된다.

- [0047] 다른 종류의 테넌트 잡들이 호스팅 환경(300)의 정비에 중점을 둘 수 있다. 이런 테넌트-잡 유형의 예시들로 인 프라 업데이트(오동작 시에 인스턴스들(321, 322 및 323)을 오프라인이 되게 함), 하나의 지리적 위치에서 다른 지리적 위치로 애플리케이션(350)의 이전(예컨대, 하드웨어 및 기타 플랫폼 문제로 인해 인스턴스들(321, 322 및 323)의 재배포), 및 리소스 업데이트(예컨대, 노드(320)의 VM에서 실행되는 인스턴스들(321, 322 및 323)을 모니터링하는 에이전트에 대한 업데이트)를 포함한다. 호스팅 환경(300)의 변경에 중점을 두었지만, 이런 테넌트 잡 유형은 인스턴스들(321, 322 및 323)의 가용성을 방해할 수도 있다.
- [0048] 테넌트 잡의 실행은 호스팅 환경(300)의 필요로 인해(예컨대, 운영 체제 소프트웨어의 버전 업데이트) 구동되거나 그 큐레이터에 의해 유도될 수 있다. 데이터 센터(200) 내에서 구현되도록 테넌트 잡이 선택될 때, 테넌트 잡의 실제 실행은 단계별로 점진적으로 수행될 수 있다. 즉, 테넌트 잡은 단계들로 나뉘어지고, 각각의 단계의 실행이 서비스 애플리케이션(350)의 부분(예컨대, 하나 이상의 롤 인스턴스)에 영향을 줄 가능성이 있다. 예를 들어, 서비스 애플리케이션(350)이 롤 A의 열 개의 인스턴스들을 갖고 있는 경우, 테넌트 잡은 열 단계로 나뉘어지고, 이는 부분 집합들로 더 나뉘어질 수 있다. 실시예에서, 테넌트-잡 유형, 테넌트 잡 특유의 제약, 또는 서비스 애플리케이션(350)의 배치 레이아웃에 따라서, 전체 업데이트 도메인 또는 업데이트 도메인의 부분 집합이 단계에 의해 영향을 받을 수 있다. 또한, 실행될 단계들의 배열은 테넌트-잡 유형에 특정되거나 및/또는 서비스 애플리케이션(350)의 서비스 모델에 의해 좌우될 수 있다.
- [0049] 동작 시에, 각각의 단계의 실행은 각각의 업데이트 도메인이 다루어질 때까지 다른 업데이트 도메인에서 테넌트 잡을 실행하는 것을 포함할 수 있다. 예를 들어, 테넌트 잡의 제 1 단계는 정비 작업을 실행하기 위해 업데이트 도메인(305)의 노드(320)를 사용 불가하게 하는 것을 포함할 수 있고, 테넌트 잡의 제 2 단계는 업데이트 도메인(310)의 노드들을 사용 불가하게 하는 것을 포함할 수 있다. 노드(320)를 사용 불가하게 하면 그에 따라 동시에 롤들(321, 322, 및 323)이 오프라인이 되게 된다. 그러나, 롤 A는 인스턴스들(311 및 321)에 위치하고, 롤 B는 인스턴스들(322 및 312)에 위치하고, 롤 C는 인스턴스들(323 및 313)에 위치하기 때문에, 롤들(321, 322, 및 323)을 오프라인이 되게 하는 것이 반드시 서비스 애플리케이션(350)을 이용 불가능하게 만들지는 않을 것이다.
- [0050] 때때로, 최소 개수의 인스턴스에만 롤이 존재할 때, 테넌트 단계 동안 단지 업데이트 도메인의 부분 집합만이 실제로 중단된다. 즉, 롤 인스턴스들(321, 322, 및 323)의 임의의 조합이 동일한 업데이트 도메인(305)으로 그룹화된 경우 이들이 동시에 중단될 수 있다. 그렇지만, 테넌트 잡의 영향은 제 1 단계 동안 업데이트 도메인의 이런 롤 인스턴스들(321, 322, 및 323)에만 국한된다.
- [0051] 서비스 애플리케이션(350)이 상태 보존형 애플리케이션인 경우, 그리고 (업데이트 도메인(305)에 포함된) 롤 인스턴스들(321, 322, 및 323)의 상태들이 (업데이트 도메인(310)에 포함된) 롤 인스턴스들(311, 312, 및 313)에 복제되지 않은 경우, 테넌트 잡의 제 1 단계의 구현 시에 정보가 이용 불가능할 수 있다. 따라서, 이하에서 논의되는 본 발명의 실시예는 테넌트 잡의 단계들을 반복적으로 실행할 때 롤 인스턴스에 저장된 상태들이 이용 불가능하게 되지 않도록 테넌트 잡을 조정하는 기술을 포함한다.
- [0052] 나아가 본 발명의 실시예는 분산 서비스 애플리케이션(350)의 토폴로지에 대한 업데이트를 포함한다. 장점으로, 토폴로지 업데이트를 통해 서비스 애플리케이션(350)의 롤 인스턴스들은 상호 통신의 목적으로 서비스 애플리케이션(350)의 다른 롤 인스턴스들을 확인할 수 있다. 예를 들어, 토폴로지 업데이트는 롤 인스턴스들을 통해 서비스-애플리케이션 토폴로지에 관한 정보를 전파하는 메커니즘을 포함한다. 예를 들어, 이렇게 전파된 정보는 재배포된 롤 인스턴스들의 변경된 주소의 설명 및/또는 전체 서비스 애플리케이션(350)에 분산된 기타 정보를 포함할 수 있다.
- [0053] 이제부터 도 3을 참조하여 토폴로지 업데이트가 설명될 것이다. 먼저, 롤 A의 인스턴스(321)가 롤 B의 인스턴스(322)와 통신하기를 원하는 경우, 인스턴스(321)는 통신을 시작하기 위해 인스턴스(322)의 위치(예컨대, IP 주소)를 알아야 한다. 통상적으로, 둘 간의 통신을 조정하기 위해서 각각의 롤 인스턴스들(321 및 322)의 이름에 대한 IP 주소의 매핑이 사용된다. 이러한 매핑은 인스턴스들(321 및 322) 각각에서 유지된다.
- [0054] 노드(320)에서의 인스턴스화(instantiation) 이래로 인스턴스들(321 및 322)의 위치가 계속 고정되어 있으므로, 인스턴스들(321 및 322)이 초기 배치되면, 매핑이 통용된다. 시간이 흘러, 서비스 애플리케이션(350)에 동적으로 업데이트가 일어나면(예컨대, 재배포, 추가, 또는 삭제), 인스턴스들(321 및 322) 중 어느 한쪽 또는 양쪽 모두에서 원래 IP 주소가 변경될 수 있다. 인스턴스들(321 및 322) 중 하나 이상에서 이렇게 변경된 IP 주소는 서비스 애플리케이션의 다른 인스턴스들이 부정확한 매핑을 갖게 만들기 때문에, 서비스-애플리케이션 토폴로지

의 모순된 뷰와 상호 통신의 두절을 초래한다. 예를 들어, 인스턴스(321)에 유지된 매핑이 인스턴스(322)의 이동으로 인해 서비스-애플리케이션 토폴로지에 관한 부정확한 정보를 포함하는 경우, 인스턴스(321)는 인스턴스(322)를 찾지 못하거나 인스턴스(322)와 교류할 수 없게 된다.

[0055] 상기에서 논의된 이런 문제를 해결하기 위해서, 업데이트 도메인에 관한 IP 주소의 변경을 기억하는 중복분 측정치를 부과하도록 토폴로지 업데이트가 구성된다. 예를 들어, 두 롤 인스턴스들(321 및 311)이 별개의 두 업데이트 도메인(305 및 310)에 각각 추가될 때, 제 1 롤 인스턴스(321)가 그 내부에 배치되었다고 제 1 업데이트 도메인(305)에서 작동하는 서비스 애플리케이션(350)에 통지되는 한편, 제 2 롤 인스턴스(311)가 그 내부에 배치되었다고 제 2 업데이트 도메인(310)에서 작동하는 서비스 애플리케이션(350)에 통지된다. 다음으로, 통지가 배포되면, 제 1 및 제 2 롤 인스턴스들(321 및 311)이 인스턴스화되고 각자의 업데이트 도메인(305 및 310)에서 시작된다. 따라서, 다른 롤 인스턴스들(322, 323, 312, 및 313)의 매핑에 존재하는 토폴로지 정보는 두 번 업데이트되어, 새로운 롤 인스턴스들(321 및 311)이 위치한 곳의 일관된 뷰를 서비스 애플리케이션(350)에 제공하게 된다.

[0056] **업데이트 도메인(UD)에의 테넌트 잠 적용**

[0057] 이제 도 4에서, 본 발명의 실시예에 따른, 상태 보존형 서비스 애플리케이션(예컨대, 롤 A)과 상태 비보존형 서비스 애플리케이션(예컨대, 롤 B) 양쪽에서 테넌트 잠의 단계를 실행하는 예시적인 프로세스 흐름을 도시하는 그래픽 표현이 도시된다. 상태 비보존형 서비스 애플리케이션에서 테넌트 잠을 실행할 때, 롤 인스턴스들 중 하나 이상에 구체적으로 저장된 내부 상태는 보통 고려하지 않는다. 따라서, 상태 비보존형 애플리케이션의 롤 인스턴스들은 예정된 대로 오프라인으로 될 수 있다. 예를 들어, 테넌트 잠의 한 단계가 상태 비보존형 애플리케이션의 롤 인스턴스 B(410)를 포함하는 UD의 업데이트를 포함할 때, 롤 인스턴스 B(410)에서 테넌트 잠이 실행되어(도 4의 블록(440) 참조) 그 결과 업데이트된 롤 인스턴스 B'(420)가 생긴다. 본 예시에서, 상태들이 원격 데이터 저장소에 안전하게 업로드되기 때문에(즉, 롤 인스턴스 B(410)에 지역적으로 저장되지 않음) 상태 비보존형 애플리케이션의 다른 롤 인스턴스들과의 조정을 할 이유가 없다.

[0058] 그러나, 상태 보존형 서비스 애플리케이션에서 테넌트 잠을 실행할 때는, 롤 인스턴스들에 저장된 내부 상태를 고려한다. 따라서, 상태 보존형 서비스 애플리케이션의 롤 인스턴스들은 조정된 방식으로 오프라인으로 될 수 있다. 예시적인 실시예에서, 이러한 조정은 UD에서 테넌트 잠을 실행할 때 적어도 다음 단계들, UD 준비 단계(블록(430) 참조), UD에서 작업 실행 단계(블록(440)) 참조), 및 UD 복구 단계(블록(450) 참조)의 이용을 포함한다. 테넌트 잠의 수신을 위해 UD의 롤 인스턴스 A(405)를 준비할 때, 상태 보존형 서비스 애플리케이션의 관리 컴포넌트(도 5의 도면 부호(597, 598, 및 599) 참조)는 롤 인스턴스 A(405)의 오프라인 여부에 관한 결정을 담당하고 롤 인스턴스 A(405)에 저장된 내부 상태(들)를 확인한다.

[0059] 실시예에서, 롤 인스턴스 A(405)의 오프라인 여부를 결정하는 단계는 적어도 다음의 단계들, 특정 테넌트 잠을 실행하기 위해 패브릭 컨트롤러의 의도를 파악하는 단계(예컨대, 롤 인스턴스 A(405)가 오프라인 상태가 될 것을 관리 롤에게 알려주는 롤 인스턴스 A(405)의 목표 상태를 패브릭 컨트롤러로부터 수신함), 테넌트-잠 유형(예컨대, 실행될 작업과 롤 인스턴스 A(405)에 대한 그 작업의 영향)을 설명하는 정보를 패브릭 컨트롤러로부터 수신하는 단계, 패브릭 컨트롤러가 테넌트 잠의 실행을 일시 정지하게 하는 단계(예컨대, 관리 롤이 진행을 승인하는 응답을 할 때까지 또는 사전 결정된 기간의 만료까지), 롤 인스턴스 A(405)가 오프라인 상태가 되면 서비스 애플리케이션이 이용 가능하게 남을 것인지를 확인하는 조치를 취하는 단계, 및 패브릭 컨트롤러에 메시지로 응답하는 단계를 포함한다. 따라서, 관리 롤은 더 이상 오프라인 상태가 된 UD 내의 롤 인스턴스 A(405)에 의해 좌우되거나 또는 의존하지 않도록 서비스 애플리케이션이 스스로 재구성되게 한다. 이와 같은 방식으로, 패브릭 컨트롤러에 대한 응답으로 관리 롤로부터 전송된 메시지는 롤 인스턴스 A(405)가 오프라인이 될 준비가 되었음을 나타낼 수 있다.

[0060] 롤 인스턴스 A(405)에서 실행될 예정인 테넌트-잠 유형을 파악하고 롤 인스턴스 A(405)가 오프라인 상태가 되면 서비스 애플리케이션이 이용 가능하게 되지 않을 것임을 확인 시에, 관리 롤이 UD의 중단을 동의하지 않을 수 있다. 이런 식으로, 관리 롤은 롤 인스턴스 A(405)에서 테넌트 잠이 계속되는 것을 거부하는 메시지로 패브릭 컨트롤러에 대응할 수 있다. 관리 롤로부터의 테넌트 잠의 거부를 전달하는 메시지를 고려하여, 패브릭 컨트롤러는 테넌트-잠 유형에, 부분적으로, 기반하여 어떻게 진행할지를 지능적으로 결정한다. 예를 들어, 테넌트-잠 유형이 호스팅 환경의 기저 플랫폼에 대한 보수를 포함하는 경우, 패브릭 컨트롤러는 관리 롤이 표명한 거부에도 불구하고 롤 인스턴스 A(405)에서 테넌트 잠의 실행을 계속할 수 있다. 테넌트-잠 유형이 단지 서비스 애플리케이션에 대한 업데이트를 포함하면, 패브릭 컨트롤러는 관리 롤의 거부에 응해서 롤 인스턴스 A(405)에서의

테넌트 잡의 실행을 취소한다. 테넌트-잡 유형이 롤 인스턴스 A(405)를 호스팅하는 노드의 OS에 대한 정비 행위를 포함하는 경우, 패브릭 컨트롤러는 롤 인스턴스 A(405)를 오프라인 상태가 되도록 관리 물과의 조정을 시도할 수 있다. 일례로, 롤 인스턴스 A(405)가 테넌트 잡을 위해 현재 목표로 한 UD 밖에 있는 다른 롤 인스턴스에 그 내부 상태를 복제할 수 있도록 하기 위해 테넌트 잡의 실행을 일시적으로 중단하는 일시 정지를 관리 롤이 요청하는 것이 조정에 포함된다. 다른 예로, UD에서 중복본을 생성하기 위해 롤 인스턴스 A(405)를 오프라인 상태가 되게 하는 것을 제어하여 지연시키고 그 내부 상태를 복제함으로써 페일오버(failover)를 중재하는 한편, 동시에, UD의 다른 롤 인스턴스들을 오프라인 상태로 되게 하는 것이 조정에 포함된다.

[0061] 롤 인스턴스 A(405)가 오프라인 상태로 될 수 있다고 관리 롤 및/또는 패브릭 컨트롤러에서 결정하면(블록(435) 참조), 롤 인스턴스 A(405)를 포함하는 UD에서 작업이 시작될 수 있다(블록(440)). 작업 단계 내에서, 패브릭 컨트롤러는 테넌트 잡의 정의에 따라 다양한 태스크를 실행한다(예컨대, 정비 행위를 실행). 통상적으로, 작업 단계 동안, 패브릭 컨트롤러는 관리 물과의 대화 없이 테넌트 잡의 실행을 수행한다. 테넌트 잡의 태스크가 완료되면, 다음 단계로 이동하기 전에는, 내부 상태(들)가 전환 중인 상태가 아니도록, 롤 인스턴스 A(405)가 안정화되도록 허용된다.

[0062] 롤 인스턴스 A(405)가 안정화되면, UD가 복원될 수 있다(블록(450))을 참조). 복원 단계 동안, 패브릭 컨트롤러는 테넌트 잡이 완료되었음을 서비스 애플리케이션의 관리 롤에 통지한다. 또한, 복원 단계 동안, 관리 롤은 UD를 온라인 상태가 되게 하기 전에 영향을 받은 모든 롤 인스턴스들이 가동되고 안정되어 있음을 보장하고 롤 인스턴스 A(405)를 업데이트된 롤 인스턴스 A' (415)로 지정하는 담당을 한다. 도시된 바와 같이, UD 준비 단계(430)와 UD 복원 단계(450) 동안 서비스 애플리케이션의 관리 롤과 패브릭 컨트롤러 간에 조정이 있다. 특히, 이런 두 단계(430 및 450) 동안 관리 롤은 테넌트 잡이 다음 단계로 진행될 수 있는지 여부를 결정하기 위해 서비스 애플리케이션의 가용성을 계산한다. 잡 유형과 서비스 애플리케이션 계정 설정에 따라, 애플리케이션이 잡 실행을 막거나 또는 막지 않을 수 있지만, 어떠한 경우에도 잡 실행을 지연시킬 수는 있다.

[0063] 때때로, 테넌트 잡이 계류 중인(예컨대, 롤 A(405)가 단계(430, 440, 또는 450) 중 어느 하나에서 처리 중인) 동안에 최우선 순위 잡이 발생할 수 있다. 최우선 순위 잡이 발생하였다고 탐지될 때, 계류 중인 테넌트 잡이 선점된다. 예시적인 실시예에서, 선점에는 테넌트 잡의 실행을 양보하고, 롤 A(405)에서 최우선 순위 잡이 실행되게 하고, 최우선 순위 잡의 실행이 완료되면 테넌트 잡을 재개하는 것이 포함된다.

[0064] 선점 동안, 예를 들어, 스케줄링 컴포넌트(도 5의 도면 부호(530) 참조)는 롤 A(405)에서 단계들(430, 440, 및 450)의 진행을 담당하는 노드-상태 드라이버(도 5의 도면 부호(460) 참조)에 양보 요청을 전송할 수 있다. 이러한 양보 요청은 계류 중인 테넌트 잡을 잠시 중단시켜 테넌트 잡의 진행을 멈추게 하는 한편, 롤 A(405)에서 최우선 순위 잡을 시작하게 한다. 즉, 통상 롤 A(405)에서는 한 번에 한 개의 잡만이 실행되므로, 스케줄링 컴포넌트는 테넌트 잡을 능동(active) 상태에서 수동(passive) 상태로 전환시키고 최우선 순위 잡을 수동 상태에서 능동 상태로 전환시키는 적절한 작업을 실행하도록 노드-상태 드라이버에게 지시한다. 이런 적절한 작업은 롤 A(405)가 실질적으로 복원되도록 롤 A(405)를 강제로 온라인 상태가 되게 하는 것과 최우선 순위 잡이 롤 A(405)에서 작업을 수행하도록 롤 A(405)를 오프라인 상태가 되게 하는 것을 포함할 수 있다.

[0065] 최우선 순위 잡이 완료되면, 테넌트 잡이 재개되기를 기다리는 동안, 스케줄러는 중단된 테넌트 잡을 다시 계속할지 여부에 대한 결정을 내릴 수 있다. 예를 들어, 롤 A(405)의 상태(예컨대, 롤 A에서 실행 중인 소프트웨어의 현재 버전)가 측정되고, 그 상태에 기반하여, 테넌트 잡이 롤 A(405)에서 다시 계속되거나 또는 취소될 수 있다. 따라서, 계류 중인 테넌트 잡의 선점 프로세스는 테넌트 잡의 통상적인 진행 내에서 단계들(430, 440, 및 450)을 바꿀 수 있는(예컨대, 추가, 삭제, 수정) 다양한 결정과 행동을 포함할 수 있다.

[0066] **테넌트 잡을 구현하기 위한 컴포넌트들**

[0067] 도 5를 참조하면, 본 발명의 실시예에 따른, 테넌트 잡을 구현하는 개별 동작들을 실행하기 위한 다양한 컴포넌트들을 포함하는 클라우드-컴퓨팅 네트워크(500)를 나타내는 블록도가 도시된다. 일반적으로, 네트워크(500)는 테넌트-변경 관리자(580)와 서비스 애플리케이션(350)의 관리 물들(597-599) 간의 인터랙션의 일례를 도시한다. 이러한 인터랙션은 네트워크(500) 및/또는 서비스 애플리케이션(350)에 걸친 테넌트 잡 구현의 조정에 영향을 줄 수 있다.

[0068] 먼저, 테넌트-변경 관리자(580)는 서비스 애플리케이션(350)의 관리 물들(597-599)과의 인터랙션과, 부분적으로, 그 인터랙션에 기반하는 테넌트 잡의 배치 감독을 담당하는 도 3의 패브릭 컨트롤러(295)의 컴포넌트를 나타낸다. 배치 감독 프로세스는 테넌트 잡의 배치 표시를 관리 물들(597-599)에게 전송, 관리 물들(597-

599)의 오프라인 상태로의 사용 불가 시작, 롤 인스턴스(521-523)에서 테넌트 잡의 실행 허용, 및 롤 인스턴스들(521-523)의 온라인 상태로의 복원 시작을 일반적으로 포함한다. 서비스 애플리케이션(350)의 컴포넌트들인 관리 롤들(597-599)은 각각 업데이트 도메인(305) 내에서 그룹화된 롤 인스턴스들(521-523)의 내부 상태를 모니터링하도록 일반적으로 구성된다. 실시예에서, 테넌트 잡의 배치 표시를 수신하면, 관리 롤들(597-599)은 롤 인스턴스들(521-523)의 모니터링된 내부 상태가, 업데이트 도메인(310)의 노드들 A, B, 및 C와 같이, 테넌트 잡의 영향을 받지 않은 서비스 애플리케이션(350)의 부분에 각각 복제되었는지를 결정하는 담당을 한다. 본원에서 설명된 바와 같이, 테넌트 잡은 처음에는 노드들 X, Y 및 Z에서 각각 호스팅되는 컴포넌트들(501, 502, 및 503)의 가용성에 영향을 미치는 업데이트 도메인(305)을 목표로 한다. 또한, 관리 롤들(597-599)은 내부 상태의 가용성의 표시를 테넌트-변경 관리자(580)에게 전달하도록 구성될 수 있다.

[0069] 이제부터 테넌트-변경 관리자(580)가 도 5를 참조하여 상세하게 설명될 것이다. 앞서 언급한 바와 같이, 테넌트-변경 관리자(580)는 일반적으로 클라우드-컴퓨팅 네트워크(500)의 기저 플랫폼에 대한 업데이트 및/또는 보수를 제어하는 작업을 한다. 제어 프로세스는 관리 롤들(597-599)과 목표-상태 메커니즘(510) 간의 인터랙션을 관리하는 테넌트-변경 관리자(580)를 포함할 수 있다. 통상적으로, 각각의 테넌트 잡은 테넌트 잡 실행 시에 고유한 작업을 구동하는 상태 기계로써 동작하는 특정 목표-상태 메커니즘에 관련된다. 따라서, 테넌트-변경 관리자(580)가 관리하는 인터랙션은 다음의 단계들, 업데이트 도메인(305)의 노드들 X, Y, 및 Z에 대한 테넌트 잡을 스케줄링하는 단계, 노드들 X, Y, 및 Z에서 호스팅되는 컴포넌트들(501, 502, 및 503)의 신원을 결정하는 단계, 컴포넌트들(501, 502, 및 503)이 각각 서비스 애플리케이션(350)의 일부인 롤 인스턴스들(521, 522, 및 523)을 포함하고 있음을 인식하면, 스케줄링된 테넌트 잡을 실행하려는 의도를 관리 롤들(597-599)에게 전달하는 단계, 롤 인스턴스들(521, 522, 및 523)의 내부 상태에 관한 정보를 전달하는 메시지를 관리 롤들(597-599)로부터 수신하는 단계, 및 롤 인스턴스들(521, 522, 및 523)의 내부 상태를 고려하여 테넌트 잡의 실행을 처리하는 단계를 포함할 수 있다. 동작 시에, 롤 인스턴스들(521, 522, 및 523)의 내부 상태는 노드들 X, Y, 및 Z이 온라인 상태에서 오프라인 상태로 전환할 때 어떻게 진행하여야 하는지를 목표-상태 메커니즘(510)에게 설명한다.

[0070] 롤 인스턴스들(521-523)의 내부 상태를 모니터링하는 책임을 관리 롤들(597-599)에 각각 부여함으로써, 테넌트-변경 관리자(580)는 효율적으로 동작하기 위해 서비스 애플리케이션(350)의 구현의 구체적인 세부 사항에 대해 모르는 채로 남게 된다. 따라서, 테넌트-변경 관리자(580)와 협력하여 관리 롤들(597-599)은 임의의 애플리케이션-관리 목표를 실현하도록 서비스 애플리케이션(350)의 상태에 관한 충분한 정보를 기저 플랫폼에 제공하기, 기저 플랫폼에 의해 시작된 지속적인 정비 행위의 가시성을 서비스 애플리케이션(350)에 제공하기, 및 플랫폼으로부터 시작된 활동들을 적절하게 타이밍할 기회를 서비스 애플리케이션(350)에 제공하기 모두를 할 수 있는 패브릭 컨트롤러와 애플리케이션(350) 간의 일반적인 프로토콜을 효과적으로 도입한다.

[0071] 이제부터 예시적인 관리 롤의 성능이 도 5의 관리 롤들(597-599)에 관해서 상세하게 설명될 것이다. 일반적으로, 서비스 애플리케이션(350)의 하나 이상의 롤 인스턴스에 지역적으로 저장된 내부 상태(들)를 이해하고 네트워크(500)의 기저 플랫폼에서의 테넌트 잡들(예컨대, 정비, 업데이트, 및 복구 작업들)의 배치 중에 이러한 이해를 전달하도록 예시적인 관리 롤이 프로그래밍된다. 따라서, 본원에서 사용된 바와 같이, “관리 롤”이라는 용어는 대체로 서비스 애플리케이션의 내부 상태를 모니터링하고 평가하는 담당을 하는 서비스 애플리케이션의 컴포넌트(들)를 지칭한다. 또한, 예시적인 관리 롤은 테넌트 잡의 실행에 관련된 패브릭 컨트롤러와의 작업을 조정하는 담당을 할 수 있다.

[0072] 이러한 책임을 관리 롤에 위임하면, 서비스 애플리케이션의 롤 인스턴스들에 지역적인 내부 상태(들)에 대해 액세스 가능한 뷰가 패브릭 컨트롤러에 제공된다. 따라서, 패브릭 컨트롤러는 롤-인스턴스 내부 상태들의 복제를 파악하고 이를 고려할 수 있다. 사실상, 롤-인스턴스 내부 상태를 설명하는 정보를 수집하고 배포함으로써, 관리 롤은, 테넌트 잡 도중에 노드(들)를 오프라인 상태가 되도록 할 때, 패브릭 컨트롤러가 특정 내부 상태의 모든 복제를 중단하지는 않도록 보장하는 데 도움을 주려고 시도한다. 따라서, 패브릭 컨트롤러가 데이터 센터의 모든 상태 보존형 애플리케이션들의 모든 롤 인스턴스들의 내부 상태를 직접적으로 모니터링을 함에 따라 초래될 수 있는 비효율이 개선되는 한편, 패브릭 컨트롤러는 여전히 관리 롤을 통해 내부 상태를 볼 수 있게 된다.

[0073] 일례로, 서비스 애플리케이션(350)이 롤 A(예컨대, 고객들로부터 일련의 디지털 이미지를 수신), 롤 B(예컨대, 수신하는 대로 디지털 이미지들을 포맷), 및 롤 C(예컨대, 포맷된 디지털 이미지들을 UI 디스플레이에 제공)를 갖고 있는 경우, 롤 A, B, 및 C의 인스턴스들의 내부 상태(예컨대, 특정 고객 데이터)를 추적하는 네 번째 집합의 롤 D가 생성될 수 있다. 따라서, 관리 롤 D(예컨대, 관리 롤들(597-599))는 서비스 애플리케이션(350)의 내부 세부 사항의 정통한 이해를 갖고 있다. 예를 들어, 관리 롤 D는 디지털 이미지의 각각의 복제 이미지가 어디에 저장되어 있는지와 디지털 이미지를 저장하는 롤 인스턴스들(예컨대, 롤 인스턴스들(521-523))이 현재 이용

가능한지 여부를 파악하고 있을 수 있다. 또한, 관리 롤 D는 디지털 이미지를 지역적으로 각각 따로 저장하고 있는 많은 롤 인스턴스들을 파악하고 있을 수 있다.

[0074] 실시예에서, 관리 롤들(597-599)은 다음의 단계들, 패브릭 컨트롤러가 고가시도(high visibility)를 설정하려고 선택했었던 상태 보존형 애플리케이션으로서 서비스 애플리케이션(350)을 식별하는 단계, 및 서비스 애플리케이션(350)을 포함하는 다른 롤 인스턴스들(예컨대, 롤 인스턴스들(521-523))의 내부 상태를 이해하는 책임을 부여하기 위해 애플리케이션(350)의 하나 이상의 롤 인스턴스(예컨대, 관리롤들(597-599))를 지정하는 단계에 의해 구축될 수 있다. 일반적으로, 서비스 모델의 관점에서, 관리 롤들(597-599)은 서비스-모델 정의에서 플러그에 의해 지정되는 보통의 서비스-애플리케이션 롤들을 나타낸다. 또한, 관리 롤들(597-599)의 구축 시에, 서비스 애플리케이션(350)은 서비스 애플리케이션(350)의 다양한 독립적인 부분들을 모니터링하고 보고하도록 관리 롤들(597-599) 중 하나 이상을 구성할 수 있다.

[0075] 통상적으로, 패브릭 컨트롤러가 강제하는 일련의 조건 내에서 관리 롤들(597-599)을 배치할 장소를 결정하도록 서비스 애플리케이션(350)이 프로그래밍된다. 예를 들어, 패브릭 컨트롤러는 두 개의 관리 롤들이 동일한 노드를 공유할 수 없다는 조건을 부여할 수 있고, 따라서 다양한 롤 인스턴스들의 가용성을 높게 유지할 수 있다. 본 예시에서, 도 5를 참조하면, 관리 롤들(597-599)은 모니터링되고 있는 다른 롤 인스턴스들(521-523)(즉, 서비스-롤 인스턴스들 대 관리-롤 인스턴스들)과 동일한 노드들 X, Y, 및 Z(예컨대, 물리적 기계 또는 가상 기계)에 구축되지는 않는다.

[0076] 관리 롤들(597-599)을 구축 및 배치하는 대로, 관리 롤들(597-599)은 앞서 논의한 바와 같이 테넌트 잡들의 구현을 관리하는 테넌트-변경 관리자(580)에게 자신의 내부 상태에 대한 관리 롤들(597-599)의 이해를 노출시킴으로써, 모니터링되는 롤 인스턴스들(521-523)에서의 테넌트 잡들의 조정에 일반적으로 영향을 미칠 수 있게 된다. 롤 인스턴스들(521-523)을 크롤링(crawling)하고 서비스 애플리케이션(350) 자체의 가용성이 내부 상태들의 복제 상태의 개수와 위치에 기반을 두는 복합 논리를 임의적으로 구현함으로써 롤-인스턴스 내부 상태에 관한 이러한 이해를 얻을 수 있다. 내부 상태에 대한 이해가 알려지면, 테넌트-변경 관리자(580)에게 이러한 이해가 전달된다. 롤 인스턴스들(521-523)의 내부 상태들에 대한 이해를 전달하기 위해 관리 롤들(597-599)에 의해 채택된 계획의 예시에서, 관리 롤들(597-599)은 롤 인스턴스들(521-523)을 평가하고 테넌트-변경 관리자(580)로부터의 요구에 대한 평가를 요약한 메시지를 전달할 수 있다. 또는, 평가 및 전달 프로세스는 (예컨대, 사전 정의된 시기에 또는 트리거링 이벤트를 탐지하면) 관리 롤들(597-599)에 의해 자동으로 실행될 수도 있다.

[0077] 테넌트-변경 관리자(580)는, 실시예에서, 테넌트 잡의 조정에 관련된 특정 태스크들을 실행하는 다양한 컴포넌트들을 포함할 수 있다. 도 5에 도시된 바와 같이, 예시적인 컴포넌트들로 실행 엔진(520), 스케줄링 컴포넌트(530), 테넌트-잡 관리 컴포넌트(540), 및 잡-호환성(job-compatibility) 컴포넌트(550)를 포함한다. 잡-호환성 컴포넌트(550)는 테넌트 잡들의 선점 프로세스를 처리하도록 구성되고, 여기서 어느 하나의 잡(두 번째로 배치)이 계류 중인 다른 테넌트 잡(첫 번째로 배치)보다 높은 우선 순위를 갖고 있다. 테넌트 잡 관리 컴포넌트(540)는 테넌트 잡들의 배치를 조정하고, 스케줄링 컴포넌트(530)에 명령을 전송하고, 테넌트 잡들의 진행을 추적한다. 일반적으로 스케줄링 컴포넌트(530)는 다른 서비스 애플리케이션에서 현재 실행되고 있는 계류 중인 테넌트 잡들의 유형을 고려하면서 테넌트 잡들의 실행을 스케줄링하는 담당을 한다. 나아가, 스케줄링 컴포넌트는 (도 6과 관련하여 아래에서 논의되는) 우선 순위 계획에 따라 최우선 순위 잡에 의해 하나 이상의 테넌트 잡이 중단되게 하도록 구성될 수 있다.

[0078] 실행 엔진(520)은, 실시예에서, 테넌트 잡을 시작하고 실행시키도록 구성된 범용 유한 상태 기계(FSM)를 나타낸다. 통상적으로, 실행 엔진(520)은 특정 테넌트 잡에 관한 작업을 수행하기 위해 실행되는 일련의 단계들을 서술하는 프레임워크를 포함한다. 프레임워크는 어느 하나의 단계에서 다른 단계로의 진행에 관련된 일련의 단계들과 링크들 간의 배열을 제공할 수 있다. 이런 링크는 다른 단계로 이동하기 전에 첫 번째 단계를 실행할 때 충족해야만 하는 조건과 관련된다. 예를 들어, 실행 엔진(520)은 업데이트 도메인(305)의 상태 전환을 하기 위해서 스케줄링 컴포넌트(530) 및/또는 목표-상태 메커니즘(510)으로부터의 입력과 출력을 관독할 수 있고, 업데이트 도메인(305)에서의 상태 전환 완료로 인해 테넌트 잡의 적절한 실행과 업데이트 도메인(310)으로의 진행이 가능하게 된다.

[0079] 특정 실시예에서, FSM으로서 작동할 때, 실행 엔진(520)은 테넌트 잡들의 전환(예컨대, 수동 상태 및 능동 상태) 및 롤 인스턴스들의 전환(예컨대, 온라인 상태 및 오프라인 상태)에 관련된 한정된 수의 상태로 이루어지며, 여기서 각각의 전환은 어느 하나의 상태에서 시작해서 다른 상태에서 종료되는 일련의 행위를 포함한다. 탐지된 이벤트(예컨대, 기록/저장 작업)와 같은 트리거에 의해 전환이 보통 시작된다. 일례로, 트리거는 실행 엔

진(520)이 업데이트 도메인들(305 및 310)에서 테넌트 잡을 실행하게 유도하는 스케줄링 컴포넌트(530)로부터의 명령어 수신을 포함할 수 있다. 일반적으로, FSM은 한 번에 하나의 상태에만 있도록 허용하며, 본원에서는 이를 계류 중인 테넌트 잡의 현재 상태라고 한다.

[0080] 네트워크(500)의 기저 플랫폼은 테넌트 잡을 지능적으로 배치하고 실행할 때 테넌트-변경 관리자(580)를 돕는 추가 컴포넌트들을 포함한다. 이런 추가 컴포넌트는 패브릭 컨트롤러의 범위에 포함되거나, 혹은 이런 컴포넌트는 패브릭 컨트롤러를 지원하지만 패브릭 컨트롤러와는 별개로 존재할 수 있다. 실시예에서, 기저 플랫폼의 컴포넌트들은 루트(root) 업그레이드 엔진(590), 목표-상태 메커니즘(510), 및 노드-상태 드라이버(460)를 포함하지만, 그러나 이에 제한되지는 않는다.

[0081] 루트 업그레이드 엔진(590)은 일반적으로, 기저 플랫폼 또는 서비스 애플리케이션(350)의 서비스 모델의 사전 정의된 규칙을 위반하지 않고 업데이트될 수 있는 일련의 노드들(예컨대, 업데이트 도메인(310)의 노드들 A, B, 및 C 또는 업데이트 도메인(305)의 노드들 X, Y, 및 Z)을 선택하도록 설계된다. 예를 들어, 규칙은 테넌트 잡을 수신하도록 선택된 일련의 노드들이 단지 하나의 업데이트 도메인에 속하는 롤 인스턴스들을 호스팅하도록 보장할 수 있다. 테넌트 잡을 수신할 일련의 노드들이 선택되면, 선택된 일련의 노드들이 테넌트 잡의 배치를 안내하기 위해 테넌트-변경 관리자(580)에게 제시된다.

[0082] 목표-상태 메커니즘(510)은 일반적으로 관리 물들(597-599)과 통신하며, 실행 엔진(520)이 명령하는 대로 롤 인스턴스들(521-523)을 예상 상태로 전환 및 지속하도록 구성된다. 예를 들어, 업데이트 도메인(305)의 준비 단계(도 4의 도면 부호(430) 참조) 동안, 롤 인스턴스들(521-523)의 예상 상태는 오프라인 상태(435)일 수 있다. 반면, 업데이트 도메인(305)의 복원 단계(도 4의 도면 부호(450) 참조) 동안, 롤 인스턴스들(521-523)의 예상 상태는 온라인 상태(455)일 수 있다.

[0083] 목표-상태 메커니즘(510)의 작업의 일례로, 실행 엔진(520) 및/또는 스케줄링 엔진(530)은 목표-상태 메커니즘(510)이 업데이트 도메인(305)을 예상 상태로 전환시키도록 요청하는 명령(560)을 목표-상태 메커니즘(510)에 전송할 수 있다. 본원에서 사용된 바와 같이, “예상 상태”라는 용어는 일반적으로 롤 인스턴스들(521-523)을 오프라인 상태 또는 온라인 상태로 추정된 목표를 나타낸다. 롤 인스턴스들(521-523)을 예상 상태로 전환하기 위해 적절한 행동이 취해지면, 테넌트-변경 관리자(580)에게 실제 상태를 전달하는 (롤 인스턴스들(521-523)와 공통 노드들에 있는) 각각의 호스트 에이전트에게 유선 프로토콜(화살표 없는 점선으로 도시됨)을 통해 롤 인스턴스들(521-523)의 실제 상태가 보고될 수 있다. 진행 중인 테넌트 잡의 구현을 어떻게 계속할지에 관한 목표-상태 메커니즘(510)의 결정을 돕기 위해서, 연이어, 테넌트-변경 관리자(580)는 목표-상태 메커니즘(510)에게 테넌트 잡의 실제 상태를 전달할 수 있다.

[0084] 다른 실시예에서, 업데이트 도메인(305)의 노드들 X, Y, 및 Z가 예상 상태에 도달하면, 목표-상태 메커니즘(510)은 롤 인스턴스들(521-523)의 상태를 관리 물들(597-599)에 전달할 수 있다. 오프라인 상태가 예상 상태인 실시예에서, 서비스 애플리케이션(350)이 롤 인스턴스들(521-523)을 사용에 의존하지 않게 준비되도록 롤 인스턴스들(521-523)은 사용 불능으로 간주되어야 함을 관리 물들(597-599)에 통지하는 제 1 메시지의 전송이 통신에 포함된다. 온라인 상태가 예상 상태인 실시예에서, 서비스 애플리케이션(350)이 롤 인스턴스들(521-523)에 의존할 수 있도록 롤 인스턴스들(521-523)은 사용 가능으로 간주되어야 함을 관리 물들(597-599)에 통지하는 제 2 메시지의 전송이 통신에 포함된다.

[0085] 경우에 따라서, 예상 상태를 전달하는 메시지들은 테넌트 잡들에 의한 영향을 받고 있는 노드들 X, Y, 및 Z의 각각의 루트 파티션에서 실행되는 호스트 에이전트에게 직접적으로 보내진다. 예를 들어, 메시지들을 적절한 관리 물들(597-599)에게 배포하기 이전에 도 6의 호스트 에이전트(615)에게 메시지들이 전송될 수 있다. 따라서, 호스트 에이전트는 롤 인스턴스들(521-523)의 상태를 서비스 애플리케이션(350)에게 밝히는 기능을 한다. 또한, 메시지들을 노드들 X, Y, 및 Z에 지역적으로 저장함으로써, 패브릭-컨트롤러 페일오버에도 및/또는 관리 물들(597-599)을 호스팅하는 노드들이 각각 오프라인으로 되어도(예컨대, 재부팅 또는 재포장 작업 동안) 메시지들이 살아남을 수 있다. 따라서, 네트워크(500)의 기저 플랫폼은 현재 진행 중인 테넌트 잡들에 상관 없이 메시지들에 대한 일관된 가시성을 유지하도록 설계된다.

[0086] 실시예에서, 메시지들을 통해 지속되는 예상 상태는 서비스 애플리케이션(350)의 관리 물들의 인스턴스들을 호스팅하는 노드들 각각에 전송될 수 있다. 그리고 나서, 관리 물들(597-599)은 메시지들을 정기적으로 검사하여 업데이트 도메인들(305 및 310)의 예상 상태를 주기적으로 폴링하는(polling) 메커니즘을 구현할 수 있다. 이러한 폴링은 예상 상태를 및 기타 정보를 검색하고 서비스 애플리케이션(350)에서 이를 이용할 수 있게 한다.

- [0087] 목표-상태 메커니즘(510)이 예상 상태를 설정하고 전달하면, 예상 상태에 따라, 일반적으로 노드-상태 드라이버(460)는 롤 인스턴스들(521-523)을 오프라인 상태로 이용 불가능하게 하고, 테넌트 잠을 실행하고, 테넌트-변경 관리자(580)의 지시 하에 롤 인스턴스들(521-523)을 온라인으로 복원하도록 구성된다.
- [0088] 본 발명의 각각의 실시예에서, 노드들은 예를 들어, 개인용 컴퓨터, 데스크탑 컴퓨터, 랩탑 컴퓨터, 모바일 장치, 소비자 가전, 서버(들), 도 1의 컴퓨팅 장치(100) 등과 같은 임의의 형태의 컴퓨팅 장치를 나타낼 수 있다. 통상적으로, 노드는 거기서 실행되는 롤 인스턴스들 및 다른 컴포넌트들의 작업을 지원하기 위해 컴퓨팅 유닛(예컨대, 중앙 처리 장치, 마이크로프로세서 등)을 포함하거나 컴퓨팅 유닛에 연결된다. 본원에서 이용된 바와 같이, “컴퓨팅 유닛”이라는 용어는 일반적으로 하나 이상의 운영 체제 또는 기타 기저 소프트웨어를 지원하는 처리 능력과 저장 메모리를 갖는 전용 컴퓨팅 장치를 지칭한다. 일례로, 컴퓨팅 유닛은, 각각의 종점이 다양한 프로세스와 작업을 실행할 수 있도록 노드들에 통합되거나, 혹은 동작 가능하게 결합된 유형 하드웨어 요소, 또는 기계들로 구성된다. 다른 예로, 컴퓨팅 유닛은 노드들이 수용하는 컴퓨터-판독 가능 매체에 결합된 프로세서(도시되지 않음)를 포함할 수 있다. 일반적으로, 컴퓨터-판독 가능 매체는 프로세서에 의해 실행될 수 있는 복수의 컴퓨터 소프트웨어 컴포넌트들(예컨대, 롤 인스턴스들(521-523))을, 적어도 일시적으로, 저장한다. 본원에서 이용된 바와 같이, “프로세서”라는 용어는 제한하기 위한 것이 아니며 계산 능력 내에서 동작하는 컴퓨팅 유닛의 임의의 요소를 포함할 수 있다. 이러한 능력으로, 프로세서는 명령어를 처리하는 유형 제품으로써 구성될 수 있다. 예시적인 실시예에서, 명령어의 패칭(patching), 디코딩/해석, 실행, 및 재기록이 처리에 포함될 수 있다.
- [0089] 도 5에 도시된 바와 같이, 업데이트 도메인(305)의 노드들 X, Y, 및 Z 및 업데이트 도메인(310)의 노드들 A, B, 및 C는 각각 가상 기계 혹은 물리적 기계를 나타낸다. 가상 기계로 표현될 때, 노드들은 도 1의 메모리(112)의 일부 및/또는 도 1의 프로세서들(114)의 일부를 포함할 수 있다. 본원에서 사용된 바와 같이, “가상 기계”라는 용어는 제한하기 위한 것이 아니며, 서비스 애플리케이션의 기능의 기저를 이루도록 처리 장치에 의해 실행되는 임의의 소프트웨어, 애플리케이션, 운영 체제, 또는 프로그램을 지칭할 수 있다. 실시예에서, 가상 기계는 노드의 각각의 게스트 파티션을 나타내며, 게스트 파티션은 서비스 애플리케이션(350), 또는 적어도 그 일부를 호스팅할 수 있다.
- [0090] 일반적으로, 분산적 방식으로 서비스 애플리케이션(350)을 호스팅하는 클라우드-컴퓨팅 네트워크(500)의 테넌트(예컨대, 고객)가 서비스 애플리케이션(350)을 소유한다. 또한 노드는 거기서 실행되는 호스트 에이전트(도 6의 도면 부호(615) 참조)를 지원할 수 있다. 실시예에서, 호스트 에이전트(615)는 노드의 루트 파티션에 속해 있고, 루트 파티션은 디스크 I/O 작업을 구현하는 요청과 같이 가상 기계로부터의 요청을 일반적으로 관리하는 작업을 한다.
- [0091] 도 5에 도시된 클라우드-컴퓨팅 네트워크(500)는 단지 적절한 컴퓨팅 시스템 환경의 일례에 불과하며 본 발명의 실시예의 사용 또는 기능의 범위에 관한 임의의 제한을 제시하기 위한 것이 아님을 이해하고 인식할 필요가 있다. 예를 들어, 클라우드-컴퓨팅 네트워크(500)는 공용 클라우드, 개인용 클라우드, 또는 전용 클라우드일 수 있다. 클라우드-컴퓨팅 네트워크(500)는 그 안에 도시된 임의의 하나의 컴포넌트 또는 컴포넌트들의 임의의 조합과 관련하여 어떠한 의존성 또는 요건을 갖는 것으로서 해석되어서는 안 된다. 또한, 도 5의 각종 블록은 명확성을 위해 선으로 도시되어 있지만, 실제로는, 다양한 컴포넌트의 윤곽을 그리는 것이 그렇게 명확하지는 않으므로, 비유적으로, 더 정확하게는 선은 회색이고 흐릴 것이다. 게다가, 본 발명의 실시예의 범위 내에서 원하는 기능을 얻기 위해 많은 물리적 기계, 가상 기계, 데이터 센터, 종점, 또는 이들의 조합을 이용할 수 있다.
- [0092] 클라우드-컴퓨팅 네트워크(500)는 클라우드-컴퓨팅 네트워크(350)의 테넌트들/고객들이 소유한 분산 서비스 애플리케이션의 롤 인스턴스들의 작업을 호스팅하고 지원하도록 구성된 데이터 센터들을 일반적으로 포함한다. 본원에서 사용된 바와 같이, “서비스 애플리케이션”이라는 용어는 대체적으로 클라우드-컴퓨팅 네트워크(500) 상에서 실행되는, 또는 클라우드-컴퓨팅 네트워크(500) 내의 저장 장소에 액세스하는 임의의 소프트웨어, 또는 소프트웨어의 일부를 지칭한다. 일 실시예에서, 롤 인스턴스들(예컨대, 롤 인스턴스들(521-523))은 서비스 애플리케이션(예컨대, 서비스 애플리케이션(350))의 기능 지원에 참여하는 소프트웨어 또는 컴포넌트 프로그램의 일부를 지칭할 수 있다. 도 5에 도시된 각각의 롤 인스턴스는 단지 서비스 애플리케이션을 지원하기 위한 적절한 일부분의 예시에 불과하며, 본 발명의 실시예의 사용 또는 기능의 범위에 관한 임의의 제한을 제시하기 위한 것이 아님을 이해하고 인식할 필요가 있다.
- [0093] 클라우드-컴퓨팅 네트워크(500)의 컨텍스트 안에서, 도시된 다양한 컴포넌트들이 노드들 내에서 내부적으로 통신하고, 데이터 센터를 가로질러 동적으로 만들어진 연결을 통해 물리적 노드들 사이에서 통신하고, 물리적 네

트위크 토폴로지를 통해 원격 네트워크(예컨대, 기업 사설망)의 리소스에 외부적으로 통신할 수 있다. 이런 연결은 네트워크 클라우드(도시되지 않음)를 통해 데이터 센터의 물리적 리소스에 분산된 컴포넌트들의 상호 연결을 포함할 수 있다. 네트워크 클라우드는 컴포넌트들 간의 통신을 구축하기 위해 하나의 컴포넌트가 다른 컴포넌트의 위치를 인식할 수 있도록 이런 리소스를 상호 연결한다. 예를 들어, 네트워크 클라우드는 공통 서비스 애플리케이션의 물 인스턴스들을 연결하는 채널을 통해 이런 통신을 구축할 수 있다. 예를 들어, 채널은, 제한 없이, 하나 이상의 LAN(local area network) 및/또는 WAN(wide area network)을 포함할 수 있다. 이런 네트워킹 환경은 사무실, 전사적 컴퓨터 네트워크, 인트라넷, 및 인터넷에서 흔하다. 따라서, 네트워크는 본원에서 더 이상 설명되지 않는다.

[0094] 또한, 본 발명의 실시예의 범위 내에서 원하는 기능을 얻기 위해 많은 컴포넌트들을 이용할 수 있다. 도 5의 다양한 컴포넌트들은 명확성을 위해 선으로 도시되어 있지만, 실제로는, 다양한 컴포넌트의 윤곽을 그리는 것이 그렇게 명확하지는 않으므로, 비유적으로, 더 정확하게는 선은 회색이고 흐릿 것이다. 또한, 도 5의 일부 컴포넌트들이 단일 블록들로 도시되어 있지만, 이런 도시는 성질과 개수에서 예시적이며, 제한하는 것으로 해석되어서는 안 된다(예컨대, 한 개의 실행 엔진(520)만이 도시되었지만, 각각의 테넌트 잡들에 관련된 전환을 관리하기 위해서 더 많이 구축될 수 있다).

[0095] **테넌트 잡의 우선 순위 결정 및 선점**

[0096] 이제 도 6으로 가서, 본 발명의 실시예에 따른, 패브릭 컨트롤러(295)와 서비스 애플리케이션(350) 간의 인터랙션을 묘사하는 데이터 센터(600)의 예시적인 토폴로지의 그래픽 표현이 도시된다. 먼저, 패브릭 컨트롤러(295)는 현재 계류 중인 테넌트 잡들(예컨대, 테넌트 잡들(611-613))의 잡 목록(610)을 유지할 수 있다. 스케줄링 컴포넌트(530)는 서비스 애플리케이션(350)에 영향을 미치는 현재 능동 상태이고 계류 중인 테넌트 잡들은 물론 잡 목록(610) 내에서 발생하는 임의의 적합한 수정(예컨대, 테넌트 잡들(611-613)의 재배열 또는 계류 중인 테넌트 잡의 선점)까지도 서비스 애플리케이션(350)의 적어도 하나의 관리 물(620)에게 알리도록 구성된다.

[0097] 때때로, 상기에서 언급한 바와 같이, 테넌트 잡이 현재 계류 중으로 능동 상태로 배치되어 있는 동안 최우선 순위 잡이 발생할 수 있다. 스케줄링 컴포넌트(530)는 최우선 순위 잡을 처리하는 방법을 다루는 정책(605)을 이용할 수 있고, 상이한 정책(605)의 결과 테넌트 잡들을 실행하는 방식이 달라질 수 있다. 동작 시에, 정책(605)은 테넌트 잡들(611-613)을 둘러싼 충돌에 대한 스케줄링 컴포넌트(530)의 반응(예컨대, 타임아웃 및 취소 메시지)을 다룬다. 실시예에서, 정책(605)이 나타내는 테넌트-잡 팩터들에 의해 반응이 일어난다.

[0098] 이런 팩터들 중 하나는 테넌트 잡을 개시한 엔터티의 종류와 관련이 있다. 테넌트 잡들은 애플리케이션 또는 고객으로부터 시작 및 인프라로부터 시작의 두 개의 카테고리 나뉘어질 수 있다. 고객이 서비스 애플리케이션(350)에 대한 테넌트 잡(예컨대, 버전 업데이트)을 개시할 때, 이 테넌트 잡이 잡 목록(610)에 들어가고 관리 물(620)에게 노출된다. 고객으로부터 시작된 테넌트 잡이 시작되면(예컨대, 잡 목록(610)의 순서의 맨 위에 도달하면), 패브릭 컨트롤러(295)는 고객으로부터 시작된 테넌트 잡이 목표로 하는 물 인스턴스들에 다른 잡이 현재 계류 중인지 여부를 결정하기 위해 관리 물(620)에게 질의할 수 있다. 인프라로부터 시작된 테넌트 잡(예컨대, 기저 플랫폼에 의해 내부적으로 시작되는 정비 작업 또는 영향을 받는 노드들의 재부팅을 포함하는 새로운 루트 운영-체제 버전의 설치)이 목표 물 인스턴스들에 현재 배치되어 있으면, 고객으로부터 시작된 테넌트 잡이 보류되거나(중단 상태로 전환) 전부 종료된다(취소 상태로 전환).

[0099] 인프라가 서비스 애플리케이션(350)에 대한 테넌트 잡을 개시할 때, 이러한 잡은 잡 목록(610)에 들어가거나 또는 최우선 순위가 부여되어 즉시 배치될 수 있다. 인프라로부터 시작된 테넌트 잡의 배치를 탐지하면, 패브릭 컨트롤러(295)는 인프라로부터 시작된 테넌트 잡이 목표로 하는 물 인스턴스들에 다른 잡이 현재 계류 중인지 여부를 결정하기 위해 관리 물(620)에게 질의할 수 있다. 고객으로부터 시작된 테넌트 잡이 목표 물 인스턴스들에 현재 배치되어 있는 경우, 여러 단계가 생길 수 있다. 어느 한 단계는 고객으로부터 시작된 테넌트 잡에 인프라로부터 시작된 테넌트 잡보다 높은 우선 순위를 차지할 수 있는 특권이 주어지는지 여부를 결정하는 단계를 포함한다. 특권이 주어지지 않는 경우, 다른 단계는 고객으로부터 시작된 테넌트 잡을 선점하는 단계를 포함한다. 실시예에서, 선점은 인프라로부터 시작된 테넌트 잡을 시작하면서(능동 상태로의 전환) 고객으로부터 시작된 테넌트 잡을 일시적으로 작업 중단시키는(수동 상태로의 전환) 단계를 포함한다. 따라서, 계류 중인 고객으로부터 시작된 테넌트 잡은 인프라로부터 시작된 테넌트 잡을 방해하지 않을 것이다.

[0100] 다른 실시예에서, 스케줄링 컴포넌트(530)는 잡 목록(610)을 생성하고 잡 목록(610)의 맨 위에 도달할 때 특정 테넌트 잡을 실행시키도록 구성된다. 잡 목록(610)을 생성할 때, 스케줄링 컴포넌트(530)는 잡 목록(610)의 테넌트 잡들의 순서를 설정하기 위해 우선화 계획에 따를 수 있다. 또한, 스케줄링 컴포넌트(530)는 공통 노드를

목표로 하는 둘 이상의 테넌트 잡들 간에 충돌이 있을 때 즉각적인 실행을 위해 적절한 테넌트 잡을 선택하도록 구성된다. 이런 선택은 우선화 계획을 참조하여 우선화 계획에 따라 먼저 개시되어야 하는 테넌트 잡을 확인하는 것에 기반할 수 있다. 통상적으로, 우선화 계획은 기저 플랫폼에 의해 수립된 정책(605)을 포함한다.

[0101] 이제부터 우선화 계획을 이용하는 예시적인 프로세스가 논의될 것이다. 먼저, 계류 중인 테넌트 잡이 대상 롤 인스턴스에서 실행되고 있는 동안 새로운 테넌트 잡이 스케줄링 컴포넌트(530)에 도착한다. 이 때, 새로운 테넌트 잡은 수동 상태이고, 계류 중인 테넌트 잡은 능동 상태이며, 대상 롤 인스턴스는 오프라인 상태이다. 새로운 테넌트 잡의 우선 순위가 우선화 계획으로부터 결정될 수 있다. 결정된 우선 순위가 계류 중인 테넌트 잡보다 낮은 경우, 결정된 우선 순위에 따라 잡 목록(610)의 테넌트 잡들(611-613)의 순서에 새로운 테넌트 잡(여전히 수동 상태)을 삽입하면서 스케줄링 컴포넌트(530)는 계류 중인 테넌트 잡의 실행을 계속한다(여전히 능동 상태).

[0102] 그러나, 결정된 우선 순위가 계류 중인 테넌트 잡보다 높은 경우, 스케줄링 컴포넌트(530)는 잡-선점 시퀀스의 중재 동작을 실행함으로써 계류 중인 테넌트 잡의 실행을 선점할 수 있다. 이러한 중재 동작은 계류 중인 테넌트 잡을 수동 상태로 전환하고 새로운 테넌트 잡을 능동 상태로 전환하는 것을 포함할 수 있다. 계류 중인 테넌트 잡을 수동 상태로 전환할 때, 스케줄링 컴포넌트(530)의 목표는 대상 롤 인스턴스가 현재 오프라인 상태에서 최대한 빨리 온라인 상태가 되도록 하는 것이다. 이를 위해서, 계류 중인 테넌트 잡의 실행은 대상 롤 인스턴스에서 보류되거나, 또는 실행을 마치고(예컨대, 도 4의 단계(440)에서의 작업 실행) 명확한 상태-전환 경로를 통해 종료하게 된다. 계류 중인 테넌트 잡이 보류되면, 대상 롤 인스턴스에서 새로운 테넌트 잡이 실행되고, 새로운 테넌트 잡의 실행을 마치면, 이전에 계류 중이었던 테넌트 잡이 대상 롤 인스턴스에서 중단 시점에서 재개된다.

[0103] 스케줄링 컴포넌트(530)가 계류 중인 테넌트 잡의 대상 롤 인스턴스에서의 실행을 마치게 할 때, 계류 중인 테넌트 잡이 사전 정의된 시간 내에 마치도록 하기 위해서 타임아웃 논리를 강제할 수 있다. 동작 시에, 대상 롤 인스턴스가 사전 정의된 시간의 만료 전에 온라인 상태를 추정하지 못하면, 타임아웃 논리는 새로운 테넌트 잡의 배치의 방해로 중단시키기 위해 계류 중인 테넌트 잡의 실행을 일찍 중단되게 할 수 있다. 그러나, 타임아웃 논리의 이점은 계류 중인 테넌트 잡의 현재 단계가 방해 없이 대상 롤 인스턴스 상에서 완료되도록 하여, 선점 중에도 서비스 애플리케이션(350)이 일관된 상태가 되게 하는 것이다.

[0104] 예시적인 실시예에서, 계류 중인 테넌트 잡을 마무리 짓기 위해 타임아웃 논리에 의해 할당된 사전 정의된 시간은, 부분적으로는, 높은 우선 순위의 새로운 테넌트 잡이 목표로 하는 서비스 애플리케이션(들)에 부여된 특권 레벨에 따라 달라질 수 있다. 즉, 일반적으로 특권 레벨은 계류 중인 테넌트 잡을 중단하기 전에 패브릭 컨트롤러(295)에 의해 대기해야 하는 시간에 서비스-애플리케이션 행동이 영향을 미치도록 허용된 레벨을 나타낸다. 어떤 경우에, 높은 특권 레벨이 부여된 서비스 애플리케이션(들)은 “준비” 및 “복원” 단계까지 마치도록 더 오랜 시간이 할당되는 반면, 낮은 특권 레벨이 부여된 서비스 애플리케이션(들)은 짧은 시간이 할당된다. 예를 들어, 핵심 내부 애플리케이션에는 제삼자 호스팅 애플리케이션보다 더 높은 특권 레벨이 수여될 수 있다. 이런 특권 레벨은 테넌트 잡들을 실행하기 위한 정책(605)을 수립하는 기저 플랫폼의 관리자에 따라 수정될 수 있다. 다른 실시예에서, 서비스 애플리케이션(들)에 부여된 특권 레벨은 테넌트 잡이 동작하지 않는 경우에 언제 동작 경고(operational alarms)를 발생시킬지와 같은 다른 작업들을 제어하는 데 사용될 수도 있다.

[0105] 선점의 경우에 실행되는 동작들의 다양한 실시예가 앞서 논의되었지만, 다른 우선 순위의 테넌트 잡들이 다르게 처리되어 스케줄링 컴포넌트(530)에서 다른 동작이 일어나게 할 수 있음을 인식하고 이해할 필요가 있다.

[0106] 서비스 애플리케이션(350)에서 테넌트 잡들을 구현할 때, 패브릭 컨트롤러(295)는 서비스 애플리케이션(350)의 롤 인스턴스들의 실행 시간(runtime) 건강 상태 정보를 관리 롤(620)에게 노출시킬 수 있다. 관리 롤(620)은 패브릭 컨트롤러(295)가 호스트 에이전트(615)의 인스턴스들에게 공개한 타임 스탬프를 갖고 있는 메시지(예컨대, XML 문서)를 통해 이런 건강 상태 정보를 이용할 수 있고, 이 건강 상태 정보는 유선 프로토콜(616)을 통해 서비스 애플리케이션(350)에 제공된다. 일반적으로, 유선 프로토콜(616)은 호스트 에이전트(615)와 서비스 애플리케이션(350)의 롤들 간의 통신을 수송한다. 도 6에 도시된 바와 같이, 통상적으로 유선 프로토콜(616)과 호스트 에이전트(615)는 일대일 대응이 된다.

[0107] 실시예에서, 건강 상태 정보를 이용 가능하게 만드는 프로세스는 서비스 애플리케이션(350)의 롤 인스턴스들에 대해 질의하여 그들의 건강 상태 정보를 패브릭 컨트롤러(350)에 보고하는 단계와 관리 롤(620)이 메시지를 검토하게 하기 전에 건강 상태 정보를 종합하는 단계를 포함한다. 메시지를 검토하면, 관리 롤(620)은 문제(예컨대, 온라인 상태로 복귀 실패)를 겪은 하나 이상의 롤 인스턴스를 탐지하고 시정 조치(예컨대, 동작하지 않는

롤 인스턴스들을 재부팅하도록 패브릭 컨트롤러(295)에 요청)를 취할 수 있다. 이처럼, 롤 인스턴스들의 건강 상태 정보를 이해함으로써 관리 롤(620)은 테넌트 잡의 결과를 평가하고 및/또는 테넌트 잡의 다음 단계로의 진행 여부를 계산할 수 있다.

[0108] 건강 상태 정보는, 실시예에서, 서비스 애플리케이션(350)의 롤 인스턴스(들)에서의 특정 내부 상태의 존재(들)를 설명하는 내부-상태 정보 또한 포함할 수 있다. 일반적으로, 패브릭 컨트롤러(295)가 서비스 애플리케이션(350)의 일부분에서 테넌트 잡을 시작하는 타이밍에 영향을 주기 위해 내부-상태 정보를 사용한다. 도 6에 도시된 바와 같이, 설명 목적으로, 서비스 애플리케이션(350)이 세 개의 롤 A 인스턴스들(A_1 , A_2 , 및 A_3)과 세 개의 롤 B 인스턴스들(B_1 , B_2 , 및 B_3)을 포함하는 경우, 내부-상태 정보는 롤 A의 어떤 인스턴스들이 제 1 내부 상태를 지역적으로 저장하고 롤 B의 어떤 인스턴스들이 제 2 내부 상태를 지역적으로 저장하는지를 설명할 수 있다. 테넌트-변경 관리자(580)가 오프라인 상태를 추정하도록 목표화된 이런 롤 인스턴스들의 표시를 관리 롤(620)에게 전송할 때, 특정 단계의 계류 중인 테넌트 잡이 배치 되면, 관리 롤(620)은 목표 롤 인스턴스들을 오프라인 상태가 되게 하는 것이 안전한지 여부를 결정하기 위해 건강 상태 정보를 참조할 수 있다.

[0109] 예를 들어, 관리 롤(620)은 롤 인스턴스 A_1 (제 1 내부 상태 유지)와 B_1 (제 2 내부 상태 유지)을 오프라인 상태로 가져가는 것에 관한 피드백을 요청 받았을 때 제 1 내부 상태가 롤 A의 둘 이상의 인스턴스에서 복제되었는지 및 제 2 내부 상태가 롤 B의 둘 이상의 인스턴스에서 복제되었는지 여부를 결정하기 위해 건강 상태 정보를 참조할 수 있다. 내부-상태 정보가 제 1 및 제 2 내부 상태가 목표 롤 인스턴스들(A_1 및 B_1)의 외부에서 복제되었음을 나타내면, 관리 롤(620)은 계류 중인 테넌트 잡이 예정대로 배치될 수 있다는 신호를 보낼 수 있다. 그러나, 내부-상태 정보가 제 1 및 제 2 내부 상태가 목표 롤 인스턴스들(A_1 및 B_1)의 외부에서 복제되지 않았음을 나타내면, 관리 롤(620)은 롤 인스턴스들 A_2 , A_3 , B_2 및 B_3 중 한 개 이상에서 각각 제 1 및 제 2 내부 상태를 복제하기 위해 계류 중인 테넌트 잡을 지연시키도록 요청할 수 있다. 이러한 지연으로 인해 관리 롤(620)은 제 1 및 제 2 내부 상태의 가용성을 보존할 시간을 얻게 된다. 상기에서 언급한 바와 같이, 요청된 지연은 패브릭 컨트롤러(295)에서 고려되지만, (예컨대, 계류 중인 테넌트 잡의 우선 순위에 기반하여) 항상 승인되는 것은 아니다.

[0110] **프로세스 순서**

[0111] 이제 도 7을 참조하면, 도 7은 본 발명의 실시예에 따른, 클라우드-컴퓨팅 네트워크의 패브릭 컨트롤러와 서비스 애플리케이션 간의 인터랙션을 용이하게 하기 위한 전반적인 방법을 보여주는 순서도(700)가 도시된다. 사용된 방법의 상이한 요소를 암시적으로 나타내기 위해 "단계" 및/또는 "블록"이라는 용어를 본원에서 사용할 수 있지만, 이러한 용어는 개별 단계들의 순서가 명시적으로 설명되지 않는 한, 그리고 명시적으로 설명될 때를 제외하고는, 본원에 개시된 다양한 단계들 사이의 임의의 특정 순서를 시사하는 것으로 해석되어서는 안 된다. 방법(700)은 먼저, 블록(710)에 표시된 대로, 서비스 애플리케이션의 하나 이상의 롤 인스턴스를 포함하는 제 1 업데이트 도메인(UD)을 선택하는 단계를 포함할 수 있다. 통상적으로, 롤 인스턴스들은 온라인 상태에서 동작하고 있고 서비스 애플리케이션의 기능을 지원하는 각각의 컴포넌트 프로그램들(예컨대, 서비스 애플리케이션의 롤의 단일 복제)을 나타낸다.

[0112] 또한 상기 방법(700)은, 블록(712)에 표시된 대로, 테넌트 잡(예컨대, 플랫폼으로부터 시작된 업데이트, 고객으로부터 시작된 업데이트, 플랫폼으로부터 시작된 보수, 또는 고객으로부터 시작된 보수)의 실행을 위해 제 1 UD를 준비하는 단계를 포함할 수 있다. 예시적인 실시예에서, 제 1 UD의 준비 단계는 적어도 다음의 단계들, 서비스 애플리케이션 내의 관리 롤에게 테넌트 잡을 실행하려는 패브릭 컨트롤러의 의도를 통지하는 단계(블록(714) 참조), 및 테넌트 잡에 의해 영향을 받는 롤 인스턴스(들)의 내부 상태가 테넌트 잡에 의해 영향을 받지 않는 서비스 애플리케이션의 일부분에 복제되는지 여부를 결정에 대해 관리 롤로부터 응답을 수신하는 단계(블록(716) 참조)를 포함한다. 제 1 UD의 롤 인스턴스(들)가 준비되면, 블록(718)에 표시된 대로, 이런 롤 인스턴스(들)의 오프라인 상태로의 사용 불가가 시작된다. 일반적으로 롤 인스턴스(들)의 사용 불가 단계는 제 1 UD 내에서 하나 이상의 롤 인스턴스를 호스팅하는 일련의 노드들이 작동되지 않게 하는 단계를 포함한다. 블록(720)에 표시된 대로, 롤 인스턴스(들)가 오프라인 상태를 취하면 테넌트 잡이 제 1 UD에서 실행되게 된다.

[0113] 한편, 상기 방법(700)은, 블록(722)에 표시된 대로, 테넌트 잡의 실행이 완료되면 제 1 UD의 롤 인스턴스(들)를 온라인 상태로 복원하는 단계를 포함할 수 있다. 예시적인 실시예에서, 제 1 UD의 인스턴스(들)를 온라인 상태로 복원하는 단계는 적어도 다음의 단계들, 테넌트 잡에 의한 영향을 받는 롤 인스턴스(들)가 기능하는지를 확인하는 단계(블록(724) 참조), 및 롤 인스턴스(들)에서 테넌트 잡의 실행이 완료되었음을 관리 롤에게 통지하여

(블록(726) 참조), 서비스 애플리케이션이 태스크들을 실행하기 위해 롤 인스턴스(들)의 이용을 재개하게 하는 단계를 포함한다. 일반적으로, 제 1 UD 내의 롤 인스턴스(들)를 온라인 상태로 복원하는 단계는 롤 인스턴스(들)를 호스팅하는 일련의 노드들이 작동되게 하는 단계를 포함한다. 제 1 UD 내의 롤 인스턴스(들)를 온라인 상태로 복원하면, 상기 방법(700)은 테넌트 잡을 실행하는 롤 인스턴스들의 제 2 UD를 선택함으로써 계속될 수 있다. 통상적으로, 제 1 UD 및 제 2 UD는 상호 배타적인 관계이며, 그 각각은 클라우드-컴퓨팅 네트워크를 통한 테넌트 잡의 전과 시에 개별 단계를 나타낼 수 있다.

[0114] 이제 도 8로 가면, 본 발명의 실시예에 따른, 테넌트 잡을 실행할 때 최우선 순위 잡을 스케줄링하는 전반적인 방법(800)을 보여주는 순서도가 도시된다. 상기 방법(800)은 서비스 애플리케이션에서 테넌트 잡을 실행하라는 표시를 수신하는 단계(블록(810) 참조) 및 서비스 애플리케이션의 하나 이상의 롤 인스턴스를 포함하는 업데이트 도메인(UD)을 식별하는 단계(블록(812) 참조)를 포함한다. 이 때, 롤 인스턴스(들)는 온라인 상태에서 동작한다. 테넌트 잡의 실행을 위해 UD가 준비되고(블록(814) 참조), 이어서 UD의 인스턴스(들)가 오프라인 상태로 사용 불가해진다(블록(816) 참조). 예시적인 실시예에서, 테넌트 잡의 실행을 위해 UD를 준비하는 프로세스는 서비스 애플리케이션 내의 관리 롤에게 테넌트 잡을 실행시키려는 의도를 통지하는 단계, 및 관리 롤로부터 긍정적인 응답을 수신하거나 부정적인 응답을 수신하는 단계를 포함하는 다양한 논리 단계를 포함한다.

[0115] 추후에, 블록(818)에 표시된 대로, 최우선 순위 잡을 구현하라는 표시를 수신한다. 본원에서 사용되는 바와 같이, “최우선 순위”라는 용어는 제한하는 것으로 여겨지지는 않으며, 우선 순위 계획 안에서 테넌트 잡을 선점하기 위해 사전 결정된 임의의 잡을 나타낼 수 있다. 최우선 순위 잡을 구현하라는 표시를 수신하면, 블록(820)에 표시된 바와 같이, 최우선 순위 잡에 테넌트 잡의 배치를 양보한다. 예시적인 실시예에서, 양보 프로세스는 롤 인스턴스(들)를 온라인 상태로 복원하는 일련의 축소된 작업들을 실행하도록 서비스 애플리케이션에 지시하는 단계(블록(822) 참조), 테넌트 잡을 보유하는 단계(블록(824) 참조), 및 롤 인스턴스(들)에서 최우선 순위 잡의 실행을 시작하는 단계(블록(826) 참조)를 포함하는 다양한 논리 단계를 포함한다.

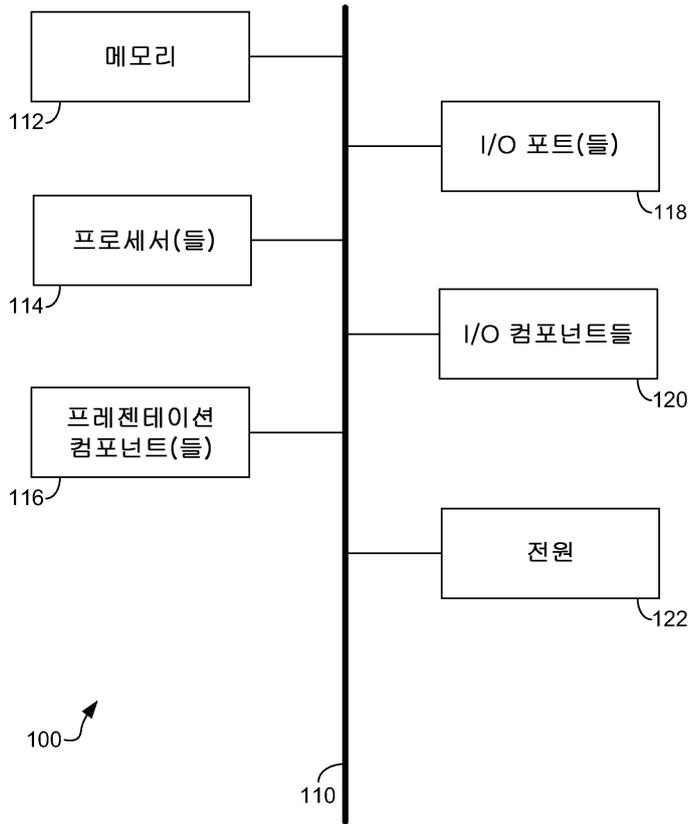
[0116] 최우선 순위 잡의 실행을 완료하면, 블록(828)에 표시된 바와 같이, 테넌트 잡의 배치가 재개된다. 예시적인 실시예에서, 재개 프로세스는 UD의 롤 인스턴스(들)의 오프라인 상태로의 사용 불가를 재시작하는 단계(블록(830) 참조), 및 롤 인스턴스(들)에서 테넌트 잡의 실행을 허용하는 단계(블록(832) 참조)를 포함하는 다양한 논리 단계를 포함한다. 테넌트 잡의 실행이 완료되면, 블록(834)에 표시된 대로, UD의 롤 인스턴스(들)는 온라인 상태로 복원될 수 있다. 예시적인 실시예에서, 롤 인스턴스(들)를 온라인 상태로 복원하는 프로세스는 테넌트 잡에 의해 영향을 받는 롤 인스턴스(들)가 기능하는지를 확인하는 단계, 및 롤 인스턴스(들)에서 테넌트 잡의 실행이 완료되었음을 관리 롤에게 통지하여, 서비스 애플리케이션이 다양한 태스크들을 실행하기 위해 롤 인스턴스(들)의 이용을 재개하게 하는 단계를 포함하는 다양한 논리 단계들을 포함한다.

[0117] 본 발명의 실시예는 특정 실시예들에 관해 설명되었으며, 이 실시예들은 제한하기보다는 예시를 들기 위한 것이다. 본 발명의 범위를 벗어나지 않는 한, 본 발명에 관련된 다른 실시예들도 당업자들에게 자명할 것이다.

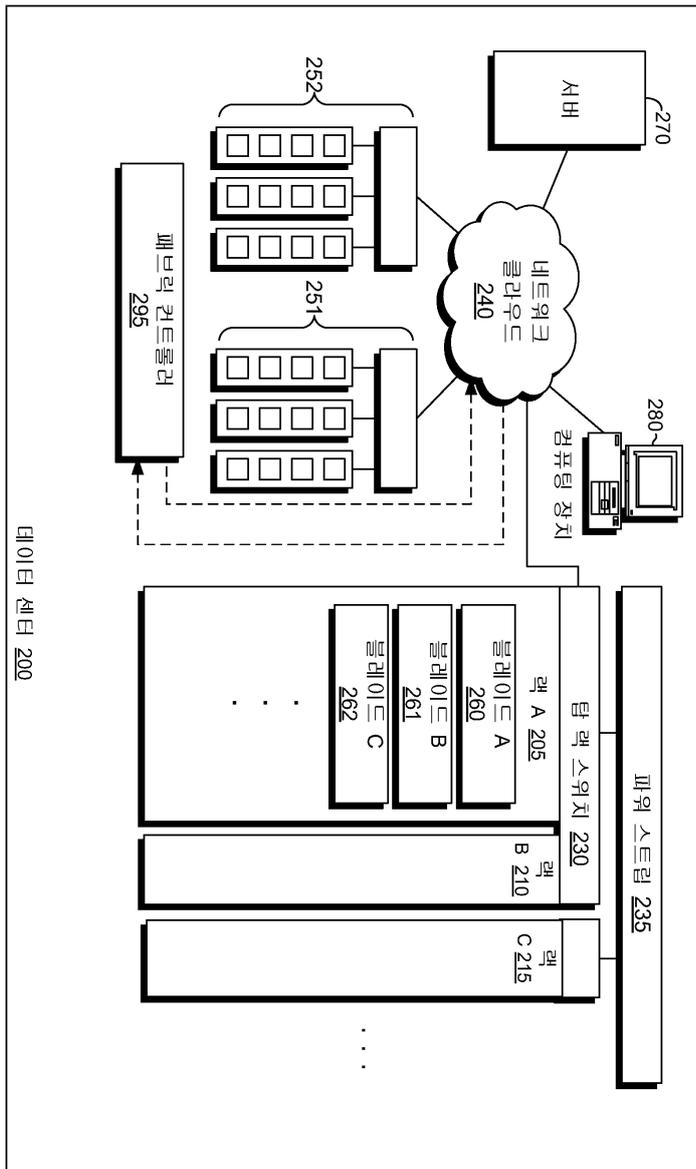
[0118] 전술한 바로부터, 본 발명은 상기 시스템 및 방법에 자명하고 고유한 다른 효과와 함께, 앞서 설명한 모든 목적 및 목표를 이루도록 조정된 것임을 알 수 있다. 특정 특징들 및 서브컴비네이션들이 유용하고, 이들은 다른 특징들 및 서브컴비네이션들과 관계없이 이용될 수 있음을 이해할 것이다. 이는 특허청구범위에 의해 그리고 그 범위 내에서 고려된다.

도면

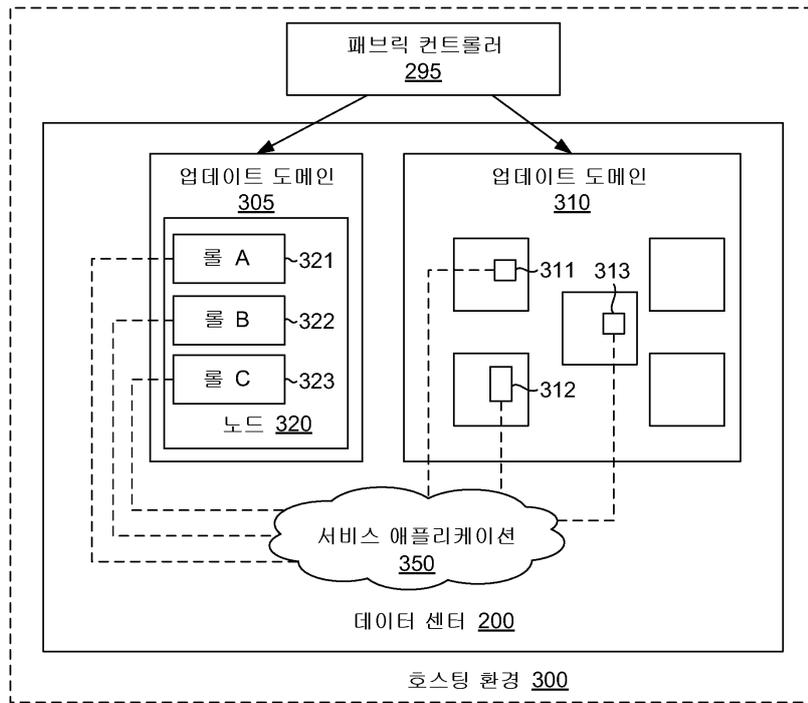
도면1



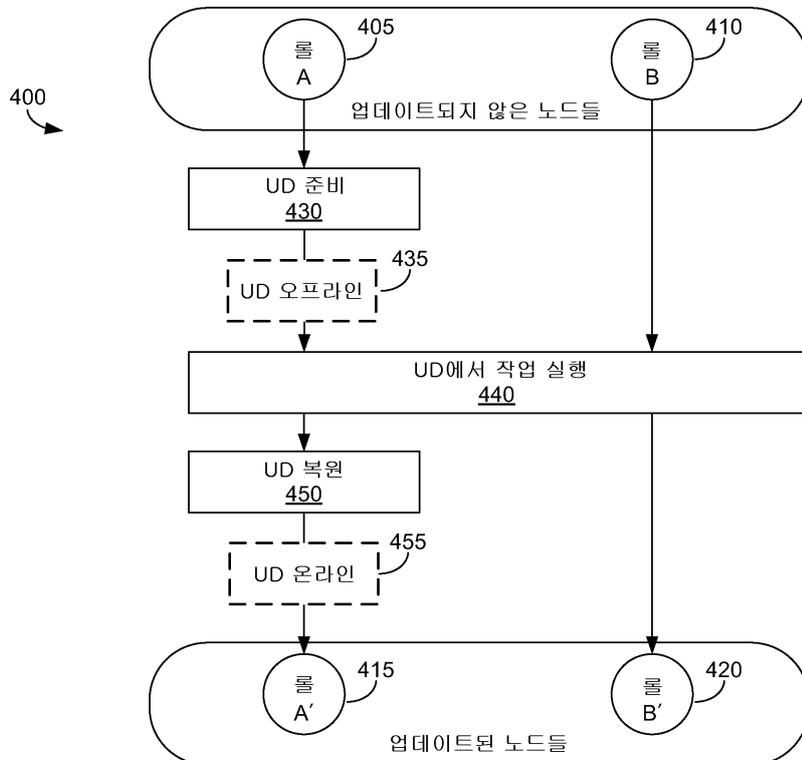
도면2



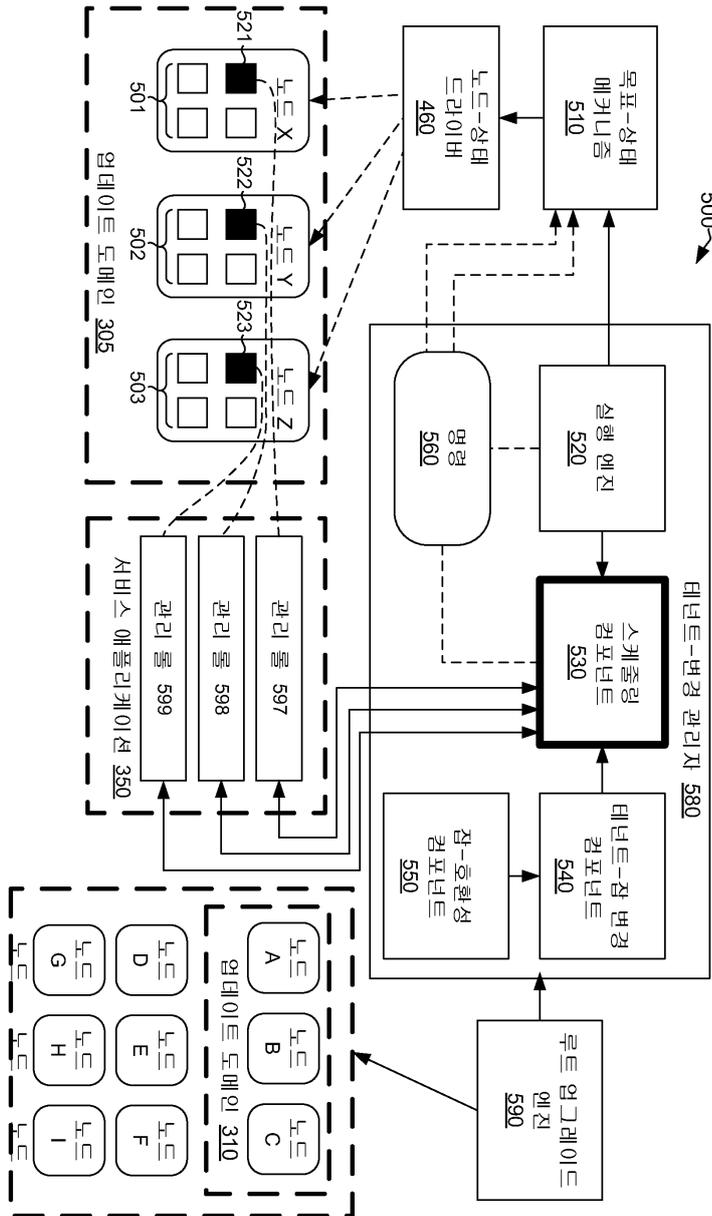
도면3



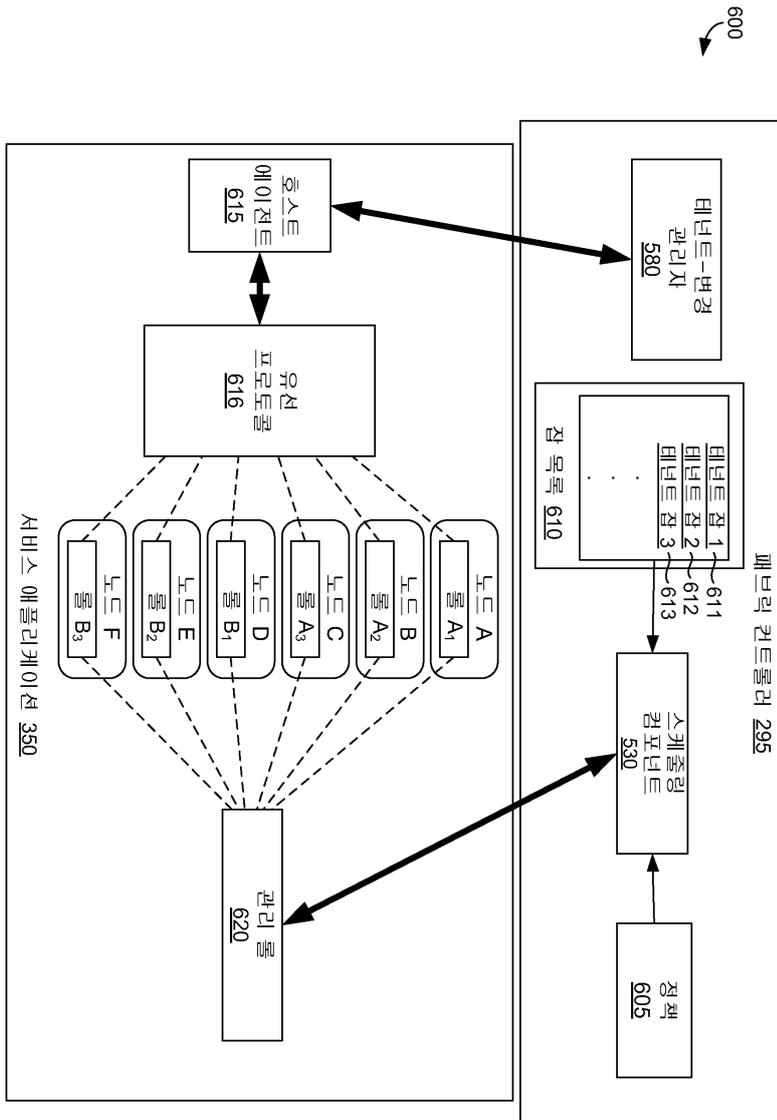
도면4



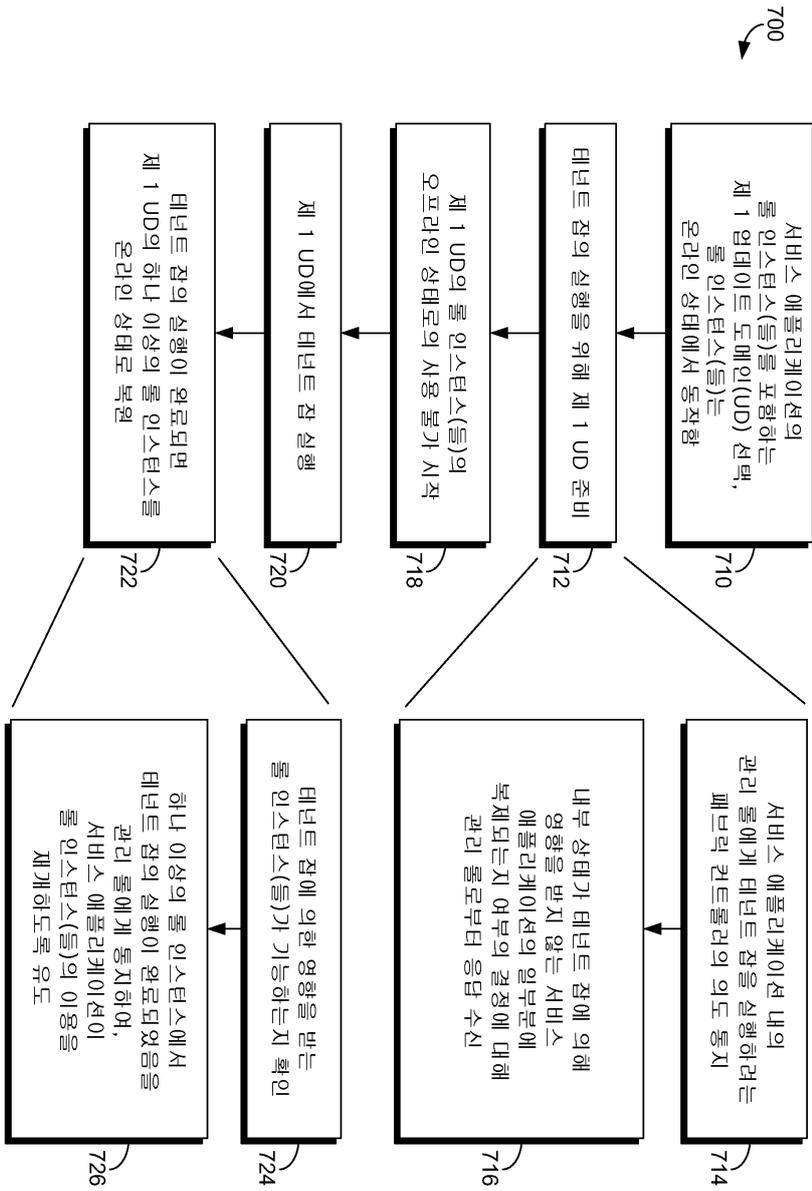
도면5



도면6



도면7



도면8

