



**(19) 대한민국특허청(KR)**  
**(12) 공개특허공보(A)**

(11) 공개번호 10-2023-0060211  
(43) 공개일자 2023년05월04일

(51) 국제특허분류(Int. Cl.)  
G06F 9/455 (2018.01) G06F 9/50 (2018.01)  
H04L 12/46 (2006.01) H04L 65/40 (2022.01)

(52) CPC특허분류  
G06F 9/45558 (2013.01)  
G06F 9/5072 (2013.01)

(21) 출원번호 10-2021-0144683  
(22) 출원일자 2021년10월27일  
심사청구일자 2021년10월27일

(71) 출원인  
송실대학교산학협력단  
서울특별시 동작구 상도로 369 (상도동)

(72) 발명자  
유명식  
서울특별시 용산구 백범로90길 90, 102동 3202호  
(문배동, 용산kcc웰츠타워)

레반끄영  
서울특별시 동작구 상도로 369, 송실대학교 형남  
공학관 1103호(상도동)

(74) 대리인  
송인호, 최관락

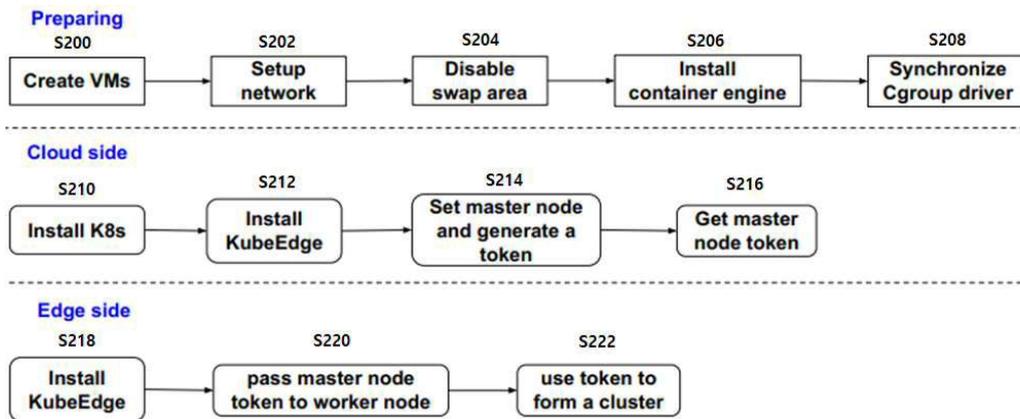
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 **에지 컴퓨팅 환경을 위한 경량 큐브에지 구성 방법 및 장치**

**(57) 요약**

본 발명은 에지 컴퓨팅 환경을 위한 경량 큐브에지 구성 방법 및 장치를 개시한다. 본 발명에 따르면, 클라우드 코어 및 에지 코어를 포함하는 큐브에지 시스템에 대해 경량 큐브에지 도구를 이용하여 에지 컴퓨팅 환경을 위한 경량 큐브에지 시스템을 구성하는 방법으로서, 상기 경량 큐브에지 도구가, 커널 기반 가상 머신(Kernal Virtual Machine, KVM)을 이용하여 가상 머신을 생성하고 네트워킹 환경을 구축하는 단계; 상기 클라우드 코어 측의 마스터 노드 및 상기 에지 코어 측의 복수의 에지 노드를 포함하는 모든 노드가, 컨테이너를 설치하고, 컨테이너 런타임을 설정하는 단계; 상기 마스터 노드가, 마스터 노드 토큰을 생성하는 단계; 상기 복수의 에지 노드가, 상기 마스터 노드 토큰을 워커 노드로 전달하는 단계; 및 상기 워커 노드가, 상기 마스터 노드 토큰을 이용하여 상기 마스터 노드와 결합하여 클러스터를 형성하는 단계를 포함하는 경량 큐브에지 시스템 구성 방법이 제공된다.

**대표도** - 도2



(52) CPC특허분류

- H04L 12/46* (2013.01)
- H04L 67/1044* (2022.05)
- H04L 67/562* (2022.05)
- G06F 2009/45562* (2013.01)
- G06F 2009/45583* (2019.08)
- G06F 2009/45595* (2019.08)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711125749
과제번호	2017-0-01633-005
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보통신방송혁신인재양성(R&D)
연구과제명	인터넷 인프라 시스템 기술 개발 및 전문 인력 양성
기여율	1/1
과제수행기관명	승실대학교 산학협력단
연구기간	2021.01.01 ~ 2021.12.31

---

## 명세서

### 청구범위

#### 청구항 1

클라우드 코어 및 에지 코어를 포함하는 큐브에지 시스템에 대해 경량 큐브에지 도구를 이용하여 에지 컴퓨팅 환경을 위한 경량 큐브에지 시스템을 구성하는 방법으로서,

상기 경량 큐브에지 도구가, 커널 기반 가상 머신(Kernal Virtual Machine, KVM)을 이용하여 가상 머신을 생성하고 네트워킹 환경을 구축하는 단계;

상기 클라우드 코어 측의 마스터 노드 및 상기 에지 코어 측의 복수의 에지 노드를 포함하는 모든 노드가, 컨테이너를 설치하고, 컨테이너 런타임을 설정하는 단계;

상기 마스터 노드가, 마스터 노드 토큰을 생성하는 단계;

상기 복수의 에지 노드가, 상기 마스터 노드 토큰을 워커 노드로 전달하는 단계; 및

상기 워커 노드가, 상기 마스터 노드 토큰을 이용하여 상기 마스터 노드와 결합하여 클러스터를 형성하는 단계를 포함하는 경량 큐브에지 시스템 구성 방법.

#### 청구항 2

제1항에 있어서,

상기 네트워킹 환경을 구축하는 단계는,

상기 경량 큐브에지 도구가, 상기 클라우드 코어 측 쿠버네티스 클러스터를 형성하기 위한 LAN과 상기 쿠버네티스 클러스터와 상기 복수의 에지 노드를 연결하기 위한 LAN을 설정하는 단계를 포함하는 경량 큐브에지 시스템 구성 방법.

#### 청구항 3

제1항에 있어서,

상기 모든 노드가 컨테이너를 설치하는 단계 이전에, 상기 경량 큐브에지 도구가, 상기 모든 노드에 대한 스왑 메모리를 비활성화하는 단계를 더 포함하는 경량 큐브에지 시스템 구성 방법.

#### 청구항 4

제1항에 있어서,

상기 모든 노드는 디바이스, CPU, RAM, 네트워크 액세스 등을 포함하는 커널의 특정 하위 시스템을 제어하기 위한 메커니즘으로 정의되는 Cgroup 드라이버를 상기 컨테이너 런타임으로 설정하는 경량 큐브에지 시스템 구성 방법.

#### 청구항 5

제1항에 있어서,

상기 마스터 노드는 쿠버네티스 API 서버인 경량 큐브에지 시스템 구성 방법.

#### 청구항 6

제1항에 있어서,

상기 마스터 노드 토큰을 생성하는 단계 이전에, 상기 마스터 노드가, 쿠버네티스 및 큐브에지를 설치하는 단계를 더 포함하는 경량 큐브에지 시스템 구성 방법.

#### 청구항 7

제1항에 있어서,

상기 마스터 노드 토큰을 상기 워커 노드로 전달하는 단계 이전에, 상기 복수의 에지 노드가 큐브에지를 설치하는 단계를 더 포함하는 경량 큐브에지 시스템 구성 방법.

**청구항 8**

제1항에 따른 방법을 수행하는 프로그램이 저장된 컴퓨터 판독 가능한 기록매체.

**청구항 9**

클라우드 코어 및 에지 코어를 포함하는 에지 컴퓨팅 환경을 위한 경량 큐브에지 시스템 구성 장치로서, 프로세서; 및

상기 프로세서에 연결되는 메모리를 포함하되,

상기 메모리는,

커널 기반 가상 머신(Kernal Virtual Machine, KVM)을 이용하여 가상 머신을 생성하고 네트워킹 환경을 구축하고,

상기 클라우드 코어 측의 마스터 노드 및 상기 에지 코어 측의 복수의 에지 노드를 포함하는 모든 노드가 컨테이너를 설치하도록 하고,

상기 모든 노드가 컨테이너 런타임을 설정하도록 하고,

상기 마스터 노드가 마스터 노드 토큰을 생성하도록 하고,

상기 복수의 에지 노드가 상기 마스터 노드 토큰을 워커 노드로 전달하도록 하고,

상기 워커 노드가 상기 마스터 노드 토큰을 이용하여 상기 마스터 노드와 결합하여 클러스터를 형성하도록,

상기 프로세서에 의해 실행되는 프로그램 명령어들을 저장하는 경량 큐브에지 시스템 구성 장치.

**청구항 10**

제9항에 있어서,

상기 프로그램 명령어들은, 상기 모든 노드가 컨테이너를 설치하기 이전에, 상기 모든 노드에 대한 스왑 메모리를 비활성화하는 경량 큐브에지 시스템 구성 장치.

**발명의 설명**

**기술 분야**

[0001] 본 발명은 에지 컴퓨팅 환경을 위한 경량 큐브에지 구성 방법 및 장치에 관한 것이다.

**배경 기술**

[0002] 모바일 장치의 수가 급격히 증가함에 따라 서비스 제공업체는 서비스를 출시할 때 짧은 대기 시간 요구 사항과 함께 대용량 데이터에 적응하기 위해 에지 컴퓨팅을 적용하는 경향이 증가하고 있다.

[0003] 애플리케이션 개발자는 대규모 클라우드 제공업체(Microsoft Azure, Amazon Web Service 등)와 함께 KubeEdge, Akri, Google Anthos와 같은 오픈 소스 프로젝트를 통해 에지에서 쿠버네티스(Kubernetes)를 구성하는 방향으로 진행하고 있다.

[0004] 쿠버네티스는 에지 컴퓨팅에 적합하지 않기 때문에 쿠버네티스를 경량화한 큐브에지(KubeEdge)가 제안되었다.

[0005] 큐브에지를 사용하면 운영자가 중앙 집중식 관리 인터페이스를 통해 에지 측에서 컨테이너화로 서비스를 배포하고 관리할 수 있다. 큐브에지 시스템에서 서비스를 배포하려면 큐브에지 시스템에서 서비스를 설계하여야 한다.

[0006] 따라서 서비스 개발을 가속화하기 위해서는 쉽고 빠른 큐브에지 시스템 구성 솔루션이 필요하다. 큐브에지 설치에는 많은 복잡한 구성과정이 요구된다.

**선행기술문헌**

**특허문헌**

[0007] (특허문헌 0001) 미국공개특허공보 2021-0232498

**발명의 내용**

**해결하려는 과제**

[0008] 상기한 종래기술의 문제점을 해결하기 위해, 본 발명은 큐브에지 시스템을 자동으로 구성할 수 있는 에지 컴퓨팅 환경을 위한 경량 큐브에지 구성 방법 및 장치를 제안하고자 한다.

**과제의 해결 수단**

[0009] 상기한 바와 같은 목적을 달성하기 위하여, 본 발명의 일 실시예에 따르면, 클라우드 코어 및 에지 코어를 포함하는 큐브에지 시스템에 대해 경량 큐브에지 도구를 이용하여 에지 컴퓨팅 환경을 위한 경량 큐브에지 시스템을 구성하는 방법으로서, 상기 경량 큐브에지 도구가, 커널 기반 가상 머신(Kernal Virtual Machine, KVM)을 이용하여 가상 머신을 생성하고 네트워킹 환경을 구축하는 단계; 상기 클라우드 코어 측의 마스터 노드 및 상기 에지 코어 측의 복수의 에지 노드를 포함하는 모든 노드가, 컨테이너를 설치하고, 컨테이너 런타임을 설정하는 단계; 상기 마스터 노드가, 마스터 노드 토큰을 생성하는 단계; 상기 복수의 에지 노드가, 상기 마스터 노드 토큰을 워커 노드로 전달하는 단계; 및 상기 워커 노드가, 상기 마스터 노드 토큰을 이용하여 상기 마스터 노드와 결합하여 클러스터를 형성하는 단계를 포함하는 경량 큐브에지 시스템 구성 방법이 제공된다.

[0010] 상기 네트워킹 환경을 구축하는 단계는, 상기 경량 큐브에지 도구가, 상기 클라우드 코어 측 쿠버네티스 클러스터를 형성하기 위한 LAN과 상기 쿠버네티스 클러스터와 상기 복수의 에지 노드를 연결하기 위한 LAN을 설정하는 단계를 포함할 수 있다.

[0011] 상기 모든 노드가 컨테이너를 설치하는 단계 이전에, 상기 경량 큐브에지 도구가, 상기 모든 노드에 대한 스왑 메모리를 비활성화하는 단계를 더 포함할 수 있다.

[0012] 상기 모든 노드는 디바이스, CPU, RAM, 네트워크 액세스 등을 포함하는 커널의 특정 하위 시스템을 제어하기 위한 메커니즘으로 정의되는 Cgroup 드라이버를 상기 컨테이너 런타임으로 설정할 수 있다.

[0013] 상기 마스터 노드는 쿠버네티스 API 서버일 수 있다.

[0014] 상기 마스터 노드 토큰을 생성하는 단계 이전에, 상기 마스터 노드가, 쿠버네티스 및 큐브에지를 설치하는 단계를 더 포함할 수 있다.

[0015] 상기 마스터 노드 토큰을 상기 워커 노드로 전달하는 단계 이전에, 상기 복수의 에지 노드가 큐브에지를 설치하는 단계를 더 포함할 수 있다.

[0016] 본 발명의 다른 측면에 따르면 상기한 방법을 수행하는 프로그램이 저장된 컴퓨터 판독 가능한 기록매체가 제공된다.

[0017] 본 발명의 또 다른 측면에 따르면, 클라우드 코어 및 에지 코어를 포함하는 에지 컴퓨팅 환경을 위한 경량 큐브에지 시스템 구성 장치로서, 프로세서; 및 상기 프로세서에 연결되는 메모리를 포함하되, 상기 메모리는, 커널 기반 가상 머신(Kernal Virtual Machine, KVM)을 이용하여 가상 머신을 생성하고 네트워킹 환경을 구축하고, 상기 클라우드 코어 측의 마스터 노드 및 상기 에지 코어 측의 복수의 에지 노드를 포함하는 모든 노드가 컨테이너를 설치하도록 하고, 상기 모든 노드가 컨테이너 런타임을 설정하도록 하고, 상기 마스터 노드가 마스터 노드 토큰을 생성하도록 하고, 상기 복수의 에지 노드가 상기 마스터 노드 토큰을 워커 노드로 전달하도록 하고, 상기 워커 노드가 상기 마스터 노드 토큰을 이용하여 상기 마스터 노드와 결합하여 클러스터를 형성하도록, 상기 프로세서에 의해 실행되는 프로그램 명령어들을 저장하는 경량 큐브에지 시스템 구성 장치가 제공된다.

**발명의 효과**

[0018] 본 발명에 따르면, 복잡한 설정 과정 없이 자동으로 경량 큐브에지 시스템을 구축할 수 있는 장점이 있다.

**도면의 간단한 설명**

- [0019] 도 1은 일반적인 큐브에지 시스템의 상세 구성을 도시한 도면이다.  
도 2는 본 실시예에 따른 경량 큐브에지 시스템을 구성하는 과정을 설명하기 위한 도면이다.
- 발명을 실시하기 위한 구체적인 내용**
- [0020] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다.
- [0021] 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.
- [0023] 큐브에지 시스템은 쿠버네티스 컨테이너형 조정 기능을 에지의 호스트로 확장하도록 설계한 오픈 소스 에지 컴퓨팅 플랫폼으로 클라우드와 에지 디바이스 간의 네트워크 애플리케이션 배포 및 메타데이터 동기화를 지원한다.
- [0024] 도 1은 일반적인 큐브에지 시스템의 상세 구성을 도시한 도면이다.
- [0025] 도 1을 참조하면, 큐브에지 시스템은 클라우드 코어(CloudCore, 100) 및 에지 코어(EdgeCore, 102)를 포함한다.
- [0026] 클라우드 코어(100)는 식별된 에지 노드로 데이터를 전송하기 위한 에지 노드 및 파드(Pod)의 메타데이터를 제어하는 확장된 쿠버네티스 컨트롤러이다.
- [0027] 클라우드 코어(100)는 에지 컨트롤러(EdgeController, 110), 디바이스 컨트롤러(DeviceController, 112) 및 클라우드 허브(Cloud Hub, 114)를 포함할 수 있다.
- [0028] 에지 컨트롤러(110)는 에지 클러스터에 존재하는 복수의 에지 노드(디바이스)를 관리한다.
- [0029] 이를 위해 에지 컨트롤러(110)는 쿠버네티스 API 서버(K8s API Server, 마스터 노드, 130)와 복수의 에지 노드를 연결하고, 이를 통해 복수의 에지 노드들을 쿠버네티스 API 서버(130)에 등록한다.
- [0030] 디바이스 컨트롤러(112)는 에지 클러스터 내의 에지 노드들의 메타데이터/상태를 관리하고, 에지 및 클라우드 간의 동기화를 통해 에지 노드를 관리한다.
- [0031] 클라우드 허브(114)는 클라우드 코어(100)와 에지 코어(102) 간, 특히 에지 컨트롤러(110)와 에지 노드 간의 통신 기능을 제공한다.
- [0032] 에지 코어(102)는 에지 허브(Edge Hub, 120), 메타관리자(MetaManager, 122), 에지D(EdgeD, 124) 및 디바이스 트윈(Device Twin, 126)을 포함할 수 있다.
- [0033] 에지 허브(120)는 에지 컴퓨팅을 위해 클라우드 코어(100)와 상호 작용하며, 클라우드 코어(100)와 에지 코어(102) 간 통신 기능을 제공하여 에지 노드의 상태를 클라우드 측으로 보고하고 이들 사이의 리소스 업데이트를 동기화한다.
- [0034] 메타관리자(122)는 에지D(124)를 통해 관리되는 에지 노드의 컨테이너의 상태를 관리한다. 이를 위해 메타관리자(122)는 디바이스 컨트롤러(112)와 메타데이터를 교환한다.
- [0035] 에지D(124)는 에지 노드의 컨테이너를 관리하며, 큐브에지 시스템에서 파드를 작동시킨다.
- [0036] 모든 에지 노드에는 컨테이너를 운영하는 에이전트인 Kubelet이 있어야 한다. 에지 허브(120) 및 클라우드 허브(114)를 통해 쿠버네티스 API 서버(130)에 에지 노드를 등록한다. Kubelet은 쿠버네티스 API 서버(130)로부터 애플리케이션 컨테이너 시작, 중지 및 종료와 같은 각 에지 노드에서 수행할 작업을 수신한다. 또한 에지 노드 및 컨테이너 상태를 쿠버네티스 API 서버(130)로 다시 보낸다.
- [0037] 디바이스 트윈(126)은 에지 노드의 상태를 저장하고, 리소스 상태를 관리한다.
- [0038] 디바이스 트윈(126)은 매퍼(Mapper, 140)를 통해 각 에지 노드의 리소스 상태를 수집하고, 이벤트 버스(Event Bus, 142)를 통해 수집된 리소스 상태를 수신하고, 자신의 DB에 저장한다.

- [0039] 이벤트 버스(142)는 MQTT 서버와 상호 작용하는 MQTT 클라이언트로, 다른 컴포넌트에 게시/구독(pub/sub) 기능을 제공한다.
- [0040] 이후 리소스 상태는 에지 허브(120) 및 클라우드 허브(114) 간 통신을 통해 디바이스 컨트롤러(112)로 전달되어 클라우드 측과 에지 측에 동일한 내용이 저장되어 관리될 수 있도록 동기화한다.
- [0041] 서비스 버스(Service Bus, 144)는 에지 노드 리소스를 활용하려는 사용자로부터 메시지를 수신하고, 이를 메타 관리자(122)를 통해 에지D(124)에 전달하여 컨테이너에서 사용자가 요청한 작업이 수행될 수 있도록 한다.
- [0042] 에지 노드는 도커, 컨테이너 런타임 인터페이스 및 컨테이너 런타임을 포함할 수 있다.
- [0043] 이하에서는 이들 구성을 상세하게 설명한다.
- [0044] CRI(Container Runtime Interface)는 플러그인 인터페이스로서, kubelet에 다양한 컨테이너 런타임을 사용할 수 있는 기능을 제공한다. 또한 쿠버네티스 플러그인은 CRI를 사용하여 컨테이너를 실행하고 관찰한다. CRI에는 gRPC API, 프로토콜 버퍼 및 라이브러리가 포함된다.
- [0045] OCI(Open Container Initiative) 런타임은 로우레벨 런타임으로서, 로우레벨 런타임은 컨테이너 수명 주기 관리를 수행하고, 컨테이너를 만들고 실행한다.
- [0046] Containerd는 Docker(도커) 엔진의 기본 CRI이며, Containerd는 컨테이너를 실행하고 노드에서 이미지와 스냅샷을 관리하기 위한 최소한의 기능을 제공한다. OCI 런타임에 실행 작업을 위임하여 컨테이너를 실행하고 종료한다. Containerd는 새 컨테이너를 실행하기 위해 다음 작업을 수행한다.
- [0047] CRI-O는 하이레벨 쿠버네티스 CRI와 OCI 런타임 사이의 링크를 제공하기 위해 구축된 컨테이너 런타임이다.
- [0048] 도커(Docker)에서 Containerd를 기본 컨테이너 런타임으로 사용한다.
- [0049] 컨테이너 런타임을 사용하여 컨테이너를 만들고 관리할 수 있다. 그러나 클러스터 시스템에서 이러한 컨테이너를 관리하고 조정하는 도구가 필요하다. 일반적으로 하나의 서비스가 다수의 컨테이너로 구성되기 때문에 컨테이너 수명 주기 관리를 위한 컨테이너 오케스트레이션 도구가 사용된다.
- [0050] Kubernetes, Apache Mesos, Docker Swarm 등과 같은 다양한 컨테이너 오케스트레이션 도구를 사용할 수 있다.
- [0051] 일반적으로 컨테이너는 파드라고 하는 복제된 그룹의 호스트에 배포된다. 컨테이너 오케스트레이션 도구는 배포를 예약하고, 유틸리티 컴퓨팅 단위, 사용 가능한 메모리, 네트워크 대역폭 등과 같은 사전 정의된 제약 조건을 기반으로 가장 적절한 호스트가 컨테이너를 배치하도록 선택된다.
- [0052] 컨테이너가 호스트에서 실행되는 동안 컨테이너 오케스트레이션 도구는 컨테이너의 수명 주기를 관리한다. 컨테이너 오케스트레이션 도구는 개인용 컴퓨터에서 Amazon Web Services(AWS), DigitalOcean 등에서 실행되는 프라이빗 클라우드 인스턴스에 이르기까지 컨테이너를 실행할 수 있는 모든 환경에서 사용할 수 있다.
- [0053] 도커 엔진은 애플리케이션을 구축하고 컨테이너화하기 위한 Linux 컨테이너를 기반으로 하는 오픈 소스 컨테이너화 플랫폼이다. 멀티 컨테이너 애플리케이션의 컨테이너 이미지를 관리하기 위한 API와 경량 컨테이너를 제공하여 현재의 컨테이너 기술을 확장한다.
- [0054] 도커 엔진은 환경 설정에 관계없이 하나의 환경에서 다른 환경으로 애플리케이션을 배포하는 데 도움이 될 수 있으며, 서비스 기능 체인의 모든 네트워크 기능을 컨테이너로 생성할 수 있다.
- [0055] 도커 컨테이너는 도커 엔진을 실행하는 호스트의 커널에 의존한다. 운영 체제 수준에서만 사용자 공간을 분리하고 응용 프로그램을 실행할 때 리소스를 소비한다. 기본적으로 컨테이너는 네임스페이스로 나머지 프로세스와 격리된 프로세스로 구성된다. 도커에서 실행되는 컨테이너는 호스트 OS 커널을 공유한다. 스토리지, 네트워킹 및 제어 그룹과 같은 Linux 커널 기능을 사용하여 운영 체제 위에 컨테이너를 구축한다. 도커는 애플리케이션의 소프트웨어를 OS, 애플리케이션 코드, 런타임, 시스템 도구, 라이브러리 등과 같이 실행하는 데 필요한 모든 것이 포함된 패키지로 묶는다. 따라서 애플리케이션은 사용자 지정 구성에 관계없이 다른 Linux 시스템에서 실행할 수 있다. 또한 도커 이미지의 읽기 전용 파일 시스템 위에 읽기-쓰기 파일 시스템을 추가하여 도커 컨테이너를 생성한다.
- [0056] 도 2는 본 실시예에 따른 경량 큐브에지 시스템을 구성하는 과정을 설명하기 위한 도면이다.
- [0057] 도 2를 참조하면, 본 실시예에 따른 경량 큐브에지 도구(tool)를 통해 클라우드 코어 및 에지 코어 모두에서 수

행되어야 하는 단계와 개별 코어에서 수행되어야 할 단계가 구분된다.

- [0058] 본 실시예에 따른 경량 큐브에지 도구는 프로세서 및 메모리를 포함하는 장치에서 실행되는 프로그램 명령어로 정의될 수 있다.
- [0059] 프로세서는 컴퓨터 프로그램을 실행할 수 있는 CPU(central processing unit)나 그 밖에 가상 머신 등을 포함할 수 있다.
- [0060] 메모리는 고정식 하드 드라이브나 착탈식 저장 장치와 같은 불휘발성 저장 장치를 포함할 수 있다. 착탈식 저장 장치는 콤팩트 플래시 유닛, USB 메모리 스틱 등을 포함할 수 있다. 메모리는 각종 랜덤 액세스 메모리와 같은 휘발성 메모리도 포함할 수 있다.
- [0061] 메모리에는 경량 큐브에지 시스템을 구성하기 위한 프로그램 명령어들이 저장된다.

[0063] **모든 노드에 적용**

- [0064] 이하의 과정은 클라우드 코어에 포함된 노드와 에지 코어에 포함된 노드 모두에서 수행되는 과정이다.
- [0065] 본 실시예에 따른 경량 큐브에지 도구는 커널 기반 가상 머신(Kernal Virtual Machine, KVM)을 이용하여 가상 머신(VM)을 생성한다(단계 200).
- [0066] 경량 큐브에지 도구는 가상 머신 생성 이후, KVM 내부에 최소 두 개의 LAN을 설정하여 네트워킹 환경을 구축한다(단계 202).
- [0067] 하나의 LAN은 큐브에지 시스템의 클라우드 측 쿠버네티스 클러스터를 형성하기 위한 것이고, 다른 하나는 클라우드 측 쿠버네티스 클러스터와 큐브에지 시스템의 에지 노드를 연결하기 위한 것이다.
- [0068] 이후, 각 가상 머신에 네트워크 인터페이스가 추가되며, 클라우드 측에서는 쿠버네티스 API 서버(130)에 에지 노드의 수에 상응하는 네트워크 인터페이스가 추가된다.
- [0069] 큐브-스케줄러가 CPU/메모리가 부족하거나 지정된 CPU/메모리 제한에 도달한 노드에 파드를 할당하지 못하도록 하려면, 클라우드 코어 측 및 에지 코어 측의 모든 노드에서 스왑 메모리(swap area)를 비활성화해야 한다.
- [0070] 큐브에지 시스템은 쿠버네티스를 기반으로 하므로 경량 큐브에지 도구는 클라우드 및 에지 코어에 포함된 모든 노드에 대해 스왑 메모리를 비활성화한다(단계 204).
- [0071] 이후, 모든 노드들은 컨테이너를 설치한다(단계 206).
- [0072] 큐브에지 시스템의 각 노드는 Cgroup 드라이버를 컨테이너의 런타임으로 설정한다(단계 208).
- [0073] 여기서, Cgroup은 디바이스, CPU, RAM, 네트워크 액세스 등을 포함하는 커널의 특정 하위 시스템을 제어하기 위한 메커니즘이다.

[0074] **클라우드 노드**

- [0075] 마스터 노드에 쿠버네티스(K8s)를 설치한다(단계 210).
- [0076] 여기서, 마스터 노드는 쿠버네티스 API 서버(130)이다.
- [0077] 이후, 마스터 노드에 큐브에지를 설치한다(단계 212).
- [0078] 마스트 노드를 설정하고, 마스터 노드가 토큰을 생성하도록 한다(단계 214).
- [0079] 마지막으로 마스터 노드 토큰을 가져온다(단계 216).

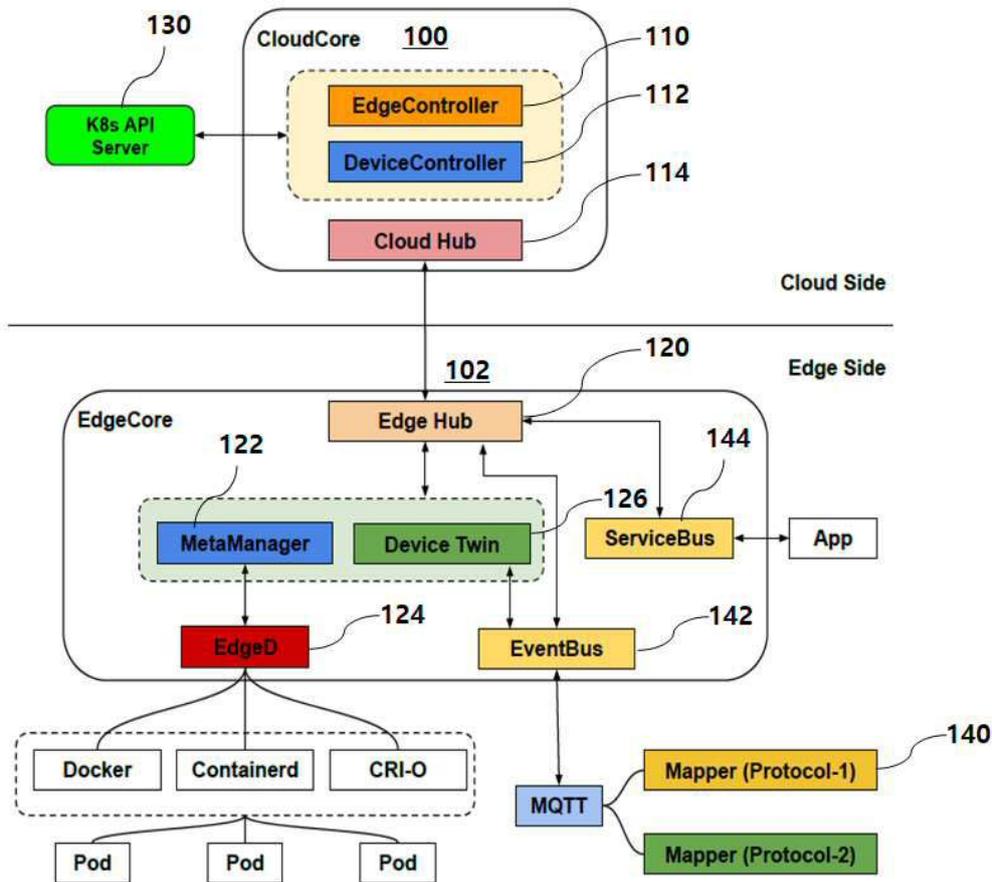
[0080] **에지 노드**

- [0081] 큐브에지를 에지 노드에 설치한다(단계 218).
- [0082] 마스터 노드 토큰을 워커 노드로 전달한다(단계 220).
- [0083] 마지막으로 워커 노드는 마스터 노드 토큰을 사용하여 마스터 노드와 결합하여 클러스터를 형성한다(단계 222).
- [0084] 상기한 본 발명의 실시예는 예시의 목적을 위해 개시된 것이고, 본 발명에 대한 통상의 지식을 가지는 당업자라

면 본 발명의 사상과 범위 안에서 다양한 수정, 변경, 부가가 가능할 것이며, 이러한 수정, 변경 및 부가는 하기의 특허청구범위에 속하는 것으로 보아야 할 것이다.

도면

도면1



도면2

