(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년02월25일
(11) 등록번호 10-2368561
(24) 등록일자 2022년02월23일

(51) 국제특허분류(Int. Cl.)
G06F 9/455 (2018.01) *G06F 9/50* (2018.01)
(52) CPC특허분류
G06F 9/45558 (2013.01)
G06F 9/505 (2013.01)
(21) 출원번호 10-2020-0081477
(22) 출원일자 2020년07월02일
심사청구일자 2020년07월02일
(65) 공개번호 10-2022-0003803
(43) 공개일자 2022년01월11일
(56) 선행기술조사문헌
KR1020160139082 A*
(뒷면에 계속)

(73) 특허권자
연세대학교 산학협력단
서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)
(72) 발명자
정종문
서울특별시 용산구 이촌로 181, 104동 101(이촌동, 한강대우아파트)
신영환
서울특별시 서대문구 연희로10길 24-10, 202호(연희동)
(뒷면에 계속)
(74) 대리인
특허법인우인

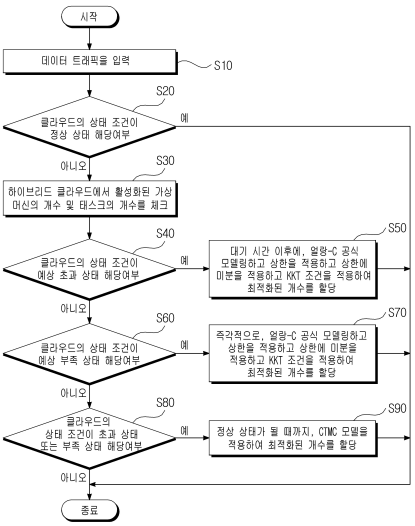
전체 청구항 수 : 총 3 항 심사관 : 유진태

(54) 발명의 명칭 **하이브리드 클라우드 기반의 IoT 환경에서 실시간 동적 자원 할당 방법**

(57) 요 약

본 실시예들은 데이터 트래픽을 입력받고, 활성화된 가상 머신의 개수와 태스크의 개수에 따른 하이브리드 클라우드의 상태 조건을 체크하여 데이터 트래픽에 따른 최소화된 가상 머신의 개수를 하이브리드 클라우드에 할당하는 방식을 통해서, 정확성과 탄력성을 모두 확보할 수 있는 하이브리드 클라우드의 동적 자원 할당 방법을 제공한다.

대 표 도 - 도3



(52) CPC특허분류
G06F 9/5066 (2013.01)
G06F 9/5077 (2013.01)
H04L 47/80 (2013.01)
G06F 2009/45562 (2013.01)
G06F 2009/4557 (2019.08)

(72) 발명자
양원식
서울특별시 서대문구 신촌로11길 11-46, 303호(창천동)
김상도
서울특별시 양천구 목동중앙북로7나길 54, 304호(목동, 석경트라움)

(56) 선행기술조사문헌
KR1020160073904 A*
KR102090911 B1
KR1020160045388 A
KR1020180014271 A
US20150039764 A1
*는 심사관에 의하여 인용된 문헌

이 발명을 지원한 국가연구개발사업

과제고유번호	1711116181
과제번호	IITP-2020-2018-0-01799
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	대학ICT연구센터지원사업
연구과제명	블록체인 비즈니스 서비스 기술 개발 및 인력양성
기 여 율	1/1
과제수행기관명	중앙대학교 산학협력단
연구기간	2018.07.01 ~ 2021.12.31

명 세 서

청구범위

청구항 1

개인 클라우드와 공공 클라우드가 결합된 하이브리드 클라우드의 동적 자원 할당 방법에 있어서,

데이터 트래픽을 입력받는 단계; 및

활성화된 가상 머신의 개수와 태스크의 개수에 따른 상기 하이브리드 클라우드의 상태 조건을 체크하여 상기 데이터 트래픽에 따른 최소화된 가상 머신의 개수를 상기 하이브리드 클라우드에 할당하는 단계를 포함한 동작들을 수행하며,

상기 상태 조건은 (i) 예상 초과 상태, (ii) 초과 상태, (iii) 정상 상태, (iv) 부족 상태, (v) 예상 부족 상태로 구분되며,

상기 데이터 트래픽에 따른 최소화된 가상 머신의 개수를 상기 하이브리드 클라우드에 할당하는 단계는,

상기 상태 조건이 상기 예상 초과 상태이면, 기설정된 대기 시간 이후에 상기 최소화된 가상 머신의 개수를 상기 하이브리드 클라우드에 할당하고,

상기 상태 조건이 상기 예상 부족 상태이면, 즉각적으로 상기 최소화된 가상 머신의 개수를 상기 하이브리드 클라우드에 할당하는 것을 특징으로 하는 동적 자원 할당 방법.

청구항 2

삭제

청구항 3

제1항에 있어서,

상기 데이터 트래픽에 따른 최소화된 가상 머신의 개수를 상기 하이브리드 클라우드에 할당하는 단계는,

상기 상태 조건이 상기 예상 초과 상태 또는 상기 예상 부족 상태이면, 상기 하이브리드 클라우드를 얼랑-C 공식(Erlang-C formula)으로 모델링하고, 상기 얼랑-C 공식에 상한을 적용하고 상기 상한에 미분을 적용하고 KKT(Karush-Kuhn-Tucker) 조건을 적용하여, 상기 데이터 트래픽에 따른 최소화된 가상 머신의 개수를 예측하는 것을 특징으로 하는 하이브리드 클라우드의 동적 자원 할당 방법.

청구항 4

삭제

청구항 5

제1항에 있어서,

상기 데이터 트래픽에 따른 최소화된 가상 머신의 개수를 상기 하이브리드 클라우드에 할당하는 단계는,

상기 상태 조건이 상기 초과 상태 또는 상기 부족 상태이면, 상기 상태 조건이 상기 정상 상태가 될 때까지, CTMC(Continuous-Time Markov Chains) 모델을 적용하여 상기 데이터 트래픽에 따른 최소화된 가상 머신의 개수를 산출하는 것을 특징으로 하는 하이브리드 클라우드의 동적 자원 할당 방법.

발명의 설명

기 술 분 야

[0001] 본 발명이 속하는 기술 분야는 하이브리드 클라우드 기반의 IoT 환경에서 실시간 동적 자원 할당 방법에 관한 것이다.

배 경 기 술

[0002] 이 부분에 기술된 내용은 단순히 본 실시예에 대한 배경 정보를 제공할 뿐 종래기술을 구성하는 것은 아니다.

[0003] 하이브리드 클라우드(Hybrid Cloud)는 공공 클라우드(Public Cloud) 및 개인 클라우드(Private Cloud) 환경이 조합된 것이다. 하이브리드 클라우드는 개인 클라우드를 통해 보안성, 데이터 프라이버시, 컴플라이언스(compliance)에 대하여 많은 통제 권한을 부여할 수 있다.

[0004] 클라우드 컴퓨팅에서 가상 머신(Virtual Machine, VM) 등의 컴퓨팅 자원을 할당하는 방식은 정확성(Accuracy)과 탄력성(Elasticity) 측면을 모두 고려해야 한다. 태스크가 발생했을 때 정확하게 가상 머신을 할당할 수 있어야 하고, 동적 환경에서 할당하는 시간을 최소화할 필요가 있으나, 하이브리드 클라우드 환경에서 정확성과 탄력성을 동시에 만족시키는 자원 할당 방식이 없는 실정이다.

선행기술문헌

특허문헌

[0005] (특허문헌 0001) 한국등록특허 제10-1720292호 (2017.03.21)

발명의 내용

해결하려는 과제

[0006] 본 발명의 실시예들은 데이터 트래픽을 입력받고, 활성화된 가상 머신의 개수와 태스크의 개수에 따른 하이브리드 클라우드의 상태 조건을 체크하여 데이터 트래픽에 따른 최소화된 가상 머신의 개수를 하이브리드 클라우드에 할당하는 방식을 통해서, 정확성과 탄력성을 모두 확보하는 데 주된 목적이 있다.

[0007] 본 발명의 명시되지 않은 또 다른 목적들은 하기의 상세한 설명 및 그 효과로부터 용이하게 추론할 수 있는 범위 내에서 추가적으로 고려될 수 있다.

과제의 해결 수단

[0008] 본 실시예의 일 측면에 의하면, 개인 클라우드와 공공 클라우드가 결합된 하이브리드 클라우드의 동적 자원 할당 방법에 있어서, 데이터 트래픽을 입력받는 단계, 및 활성화된 가상 머신의 개수와 태스크의 개수에 따른 상기 하이브리드 클라우드의 상태 조건을 체크하여 상기 데이터 트래픽에 따른 최소화된 가상 머신의 개수를 상기 하이브리드 클라우드에 할당하는 단계를 포함한 동작들을 수행하는 동적 자원 할당 방법을 제공한다.

[0009] 상기 상태 조건은 (i) 예상 초과 상태, (ii) 초과 상태, (iii) 정상 상태, (iv) 부족 상태, (v) 예상 부족 상태로 구분될 수 있다.

[0010] 상기 데이터 트래픽에 따른 최소화된 가상 머신의 개수를 상기 하이브리드 클라우드에 할당하는 단계는, 상기 상태 조건이 상기 예상 초과 상태 또는 상기 예상 부족 상태이면, 상기 하이브리드 클라우드를 얼랑-C 공식(Erlang-C formula)으로 모델링하고, 상기 얼랑-C 공식에 상한을 적용하고 상기 상한에 미분을 적용하고 KKT(Karush-Kuhn-Tucker) 조건을 적용하여, 상기 데이터 트래픽에 따른 최소화된 가상 머신의 개수를 예측할 수 있다.

[0011] 상기 데이터 트래픽에 따른 최소화된 가상 머신의 개수를 상기 하이브리드 클라우드에 할당하는 단계는, 상기 상태 조건이 상기 예상 초과 상태이면, 기설정된 대기 시간 이후에 상기 최소화된 가상 머신의 개수를 상기 하이브리드 클라우드에 할당하고, 상기 상태 조건이 상기 예상 부족 상태이면, 즉각적으로 상기 최소화된 가상 머신의 개수를 상기 하이브리드 클라우드에 할당할 수 있다.

[0012] 상기 데이터 트래픽에 따른 최소화된 가상 머신의 개수를 상기 하이브리드 클라우드에 할당하는 단계는, 상기 상태 조건이 상기 초과 상태 또는 상기 부족 상태이면, 상기 상태 조건이 상기 정상 상태가 될 때까지, CTMC(Continuous-Time Markov Chains) 모델을 적용하여 상기 데이터 트래픽에 따른 최소화된 가상 머신의 개수를 산출할 수 있다.

발명의 효과

[0013] 이상에서 설명한 바와 같이 본 발명의 실시예들에 의하면, 데이터 트래픽을 입력받고, 활성화된 가상 머신의 개수와 태스크의 개수에 따른 하이브리드 클라우드의 상태 조건을 체크하여 데이터 트래픽에 따른 최소화된 가상 머신의 개수를 하이브리드 클라우드에 할당하는 방식을 통해서, 정확성과 탄력성을 모두 확보할 수 있는 효과가 있다.

[0014] 여기에서 명시적으로 언급되지 않은 효과라 하더라도, 본 발명의 기술적 특징에 의해 기대되는 이하의 명세서에서 기재된 효과 및 그 잠정적인 효과는 본 발명의 명세서에 기재된 것과 같이 취급된다.

도면의 간단한 설명

[0015] 도 1은 하이브리드 클라우드를 예시한 도면이다.

도 2는 본 발명의 일 실시예에 따른 하이브리드 클라우드의 동적 자원 할당 장치를 예시한 블록도이다.

도 3은 본 발명의 다른 실시예에 따른 하이브리드 클라우드의 동적 자원 할당 방법을 예시한 블록도이다.

도 4 내지 도 7은 본 발명의 실시예들에 따라 수행된 모의실험 결과를 도시한 것이다.

발명을 실시하기 위한 구체적인 내용

[0016] 이하, 본 발명을 설명함에 있어서 관련된 공지기능에 대하여 이 분야의 기술자에게 자명한 사항으로서 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략하고, 본 발명의 일부 실시예들을 예시적인 도면을 통해 상세하게 설명한다.

[0017] 도 1은 하이브리드 클라우드를 예시한 도면이다.

[0018] 하이브리드 클라우드는 공공 클라우드 및 개인 클라우드 환경이 조합된 것이다. 데이터 트래픽은 IoT(Internet of Things) 장치로부터 개인 클라우드를 통해 공공 클라우드로 전달된다. 하이브리드 클라우드 환경에서 개인 클라우드는 사전에 데이터를 수신하고 복수의 클라우드에 태스크를 할당하는 브로커 역할을 한다.

[0019] 클라우드 컴퓨팅에서 가상 머신 등의 컴퓨팅 자원을 할당하는 방식은 정확성을 고려해야 한다. 클라우드 컴퓨팅에서 CTMC(Continuous-Time Markov Chains) 모델은 한 번에 하나의 가상 머신을 활성화 또는 비활성화시킨다.

[0020] CTMC 모델은 확률을 이용하여 객체 상태를 시간에 따라 어떻게 변화할지를 모델링한다. 객체의 시간에 따른 서로 다른 상태를 어떻게 연결할지를 기술하는 것이 마코브 체인이며, 이들을 연결시켜주는 매개체 역할을 마코브 행렬이 한다. 시간 매개 변수가 연속시간(continuous time)이면 마코브 프로세스에 해당한다.

[0021] 클라우드 컴퓨팅에서 가상 머신 등의 컴퓨팅 자원을 할당하는 방식은 탄력성을 고려해야 한다. CTMC 모델을 이용한 자원 할당 방식은 정확하지만 탄력성 측면을 보완할 필요가 있다.

[0022] 본 실시예에 따른 동적 자원 할당 장치는 하이브리드 클라우드 환경을 얼랑-C 공식(Erlang-C formula) 또는 M/M/c 큐로 모델링하고 KKT(Karush-Kuhn-Tucker) 모델을 적용하여 탄력적인 자원 할당을 가능하게 한다. 하이브리드 클라우드 환경이 외부 브로커를 사용하도록 변경되더라도 동작 알고리즘 측면에서 차이가 없으므로 본 실시예에 따른 동적 자원 할당 장치를 그대로 적용할 수 있다.

[0023] 도 2는 본 발명의 일 실시예에 따른 하이브리드 클라우드의 동적 자원 할당 장치를 예시한 블록도이다.

[0024] 동적 자원 할당 장치(110)는 적어도 하나의 프로세서(120), 컴퓨터 판독 가능한 저장매체(130) 및 통신 버스(170)를 포함한다.

[0025] 프로세서(120)는 동적 자원 할당 장치(110)로 동작하도록 제어할 수 있다. 예컨대, 프로세서(120)는 컴퓨터 판독 가능한 저장 매체(130)에 저장된 하나 이상의 프로그램들을 실행할 수 있다. 하나 이상의 프로그램들은 하나 이상의 컴퓨터 실행 가능 명령어를 포함할 수 있으며, 컴퓨터 실행 가능 명령어는 프로세서(120)에 의해 실행되는 경우 동적 자원 할당 장치(110)로 하여금 예시적인 실시예에 따른 동작들을 수행하도록 구성될 수 있다.

[0026] 컴퓨터 판독 가능한 저장 매체(130)는 컴퓨터 실행 가능 명령어 내지 프로그램 코드, 프로그램 데이터 및/또는 다른 적합한 형태의 정보를 저장하도록 구성된다. 컴퓨터 판독 가능한 저장 매체(130)에 저장된 프로그램(140)은 프로세서(120)에 의해 실행 가능한 명령어의 집합을 포함한다. 일 실시예에서, 컴퓨터 판독한 가능 저장 매체(130)는 메모리(랜덤 액세스 메모리와 같은 휘발성 메모리, 비휘발성 메모리, 또는 이들의 적절한 조합), 하

나 이상의 자기 디스크 저장 디바이스들, 광학 디스크 저장 디바이스들, 플래시 메모리 디바이스들, 그 밖에 호흡 구간 검출 장치(110)에 의해 액세스되고 원하는 정보를 저장할 수 있는 다른 형태의 저장 매체, 또는 이들의 적합한 조합일 수 있다.

[0027] 통신 버스(170)는 프로세서(120), 컴퓨터 판독 가능한 저장 매체(140)를 포함하여 동적 자원 할당 장치(110)의 다른 다양한 컴포넌트들을 상호 연결한다.

[0028] 동적 자원 할당 장치(110)는 또한 하나 이상의 입출력 장치(24)를 위한 인터페이스를 제공하는 하나 이상의 입출력 인터페이스(150) 및 하나 이상의 통신 인터페이스(160)를 포함할 수 있다. 입출력 인터페이스(150) 및 통신 인터페이스(160)는 통신 버스(170)에 연결된다. 입출력 장치(미도시)는 입출력 인터페이스(150)를 통해 동적 자원 할당 장치(110)의 다른 컴포넌트들에 연결될 수 있다.

[0029] 동적 자원 할당 장치(110)는 하이브리드 클라우드 환경을 얼랑-C 공식(Erlang-C formula) 또는 M/M/c 큐로 모델링하고 KKT(Karush-Kuhn-Tucker) 모델을 적용하여 탄력적인 자원 할당을 가능하게 한다.

[0030] 얼랑-C 공식(Erlang-C formula)은 고객 전화가 푸아송 분포에 따라 시스템에 도착한다고 할 때 고객이 서비스를 받을 수 있는 확률을 규정한 공식이다. 시스템은 고객의 서비스를 요청받았을 때, 서비스 창구가 모두 통화 중일 때는 이 고객을 대기 행렬(queue)에 기억시키고 차례대로 서비스를 제공한다. 대기 행렬의 형태는 M/M/c 큐 등으로 정의될 수 있다.

[0031] 제약조건(constraint)을 가지는 최적화 문제에서 연립부등식 제한조건은 KKT 조건을 만족한다. KKT(Karush-Kuhn-Tucker) 조건은 (1) 모든 독립 변수 x_1, x_2, ... , x_N에 대한 미분값이 0이다. (2) 모든 라그랑주 승수 λ_1, ... , λ_M과 제한조건 부등식(λ에 대한 미분값)의 곱이 0이다. (3) 라그랑주 승수는 음수가 아니어야 한다.

[0032] 동적 자원 할당 장치(110)는 얼랑-C 공식에 KKT 모델을 적용할 수 없는 문제를 해결하기 위해서 얼랑-C 공식에 대해서 근사치 예측을 통해 KKT 모델을 적용한다.

[0033] 클라우드 컴퓨팅은 데이터가 가상 머신 간에 교환되므로 네트워크 지연을 피할 수 없다. 장비 지연(equipment delay, T_{equip})은 상수값으로 표현될 수 있다. 전파 지연(propagation delay, T_{prop})은 발신자와 가상 머신 간의 거리를 분할하여 도출될 수 있다. 전파 지연 T_{prop}은 데이터 크기에 독립적이다. 전송 지연(transmission delay, T_{trans})은 데이터 크기에 영향을 받는다.

수학식 1

[0034] $$T_{trans} = \frac{L}{R}$$

[0035] L은 데이터 크기(비트 단위)이고 R은 평균 종단 밴드폭(bits per second, bps 단위)이다. 종단 네트워크 지연은 $T_{equip} + T_{prop} + T_{trans}$로 표현된다.

[0036] 하이브리드 클라우드 모델에서는 태스크 처리를 요청하는 복수의 IoT 장치가 존재한다. k 번째 IoT 장치로부터 태스크 X_k의 도착은 푸아송 분포에 따르는 것으로 가정할 수 있다.

수학식 2

[0037] $$\sum_{i=1}^{n} X_i \sim Poisson(\lambda)$$

[0038] λ는 평균 도착 속도이고, n은 IoT 장치의 전체 개수를 의미한다. j 번째 클라우드에서 가상 머신의 서비스 속도는 동일하고 평균 μ_j를 갖는 지수적 분포를 따른다. 각 클라우드는 M/M/c 큐로 취급될 수 있다. 클라우드 서비스 속도 μ_j를 참고하여 인덱스 j=0은 개인 클라우드를 의미하고 인덱스 j가 0이 아닌 자연수이면 j 번째 공

공 클라우드를 의미한다.

[0039] 처리된 데이터 크기는 다양하므로, IoT 장치로부터의 입력 데이터의 평균 데이터는 L_I, 평균 출력 데이터는 데이터 트래픽 계수(data traffic coefficient, DTC)를 곱하여 L_O로 표현될 수 있다.

수학식 3

[0040] $$L_O = L_I \times DTC$$

[0041] 개인 클라우드로부터 j 번째 클라우드로의 데이터 트래픽 계수는 α_j라고 하고 반대 방향의 데이터 트래픽 계수는 β_j라고 한다.

[0042] 하이브리드 클라우드의 설치 비용은 개인 클라우드에 대한 비용과 공공 클라우드에 대한 비용으로 두 배이다. 공공 클라우드에 대한 비용은 수학식 4와 같이 표현되고, 개인 클라우드에 대한 비용은 수학식 5와 같이 표현된다.

수학식 4

[0043] $$\mathbb{C}_{pub} = N_j \left(\phi_j + \psi_j \mu_j^{d_j} \right)$$

[0044] ϕ_j는 렌탈 비용과 고정 소비 전력의 합이고, $\psi_j \mu_j^{d_j}$는 동적 소비 전력이고, N_j는 j 번째 클라우드에서 가상 머신의 개수이다.

수학식 5

[0045] $$\mathbb{C}_{pri} = N_0 \left(\phi_0 + \psi_0 \mu_0^{d_0} + \psi_G \mu_G^{d_G} \right)$$

[0046] N_0은 개인 클라우드에서 가상 머신의 개수이고, ϕ_0는 가상 머신 별 구매 비용이다. 구매 비용은 네트워크 장비 비용, 가상 머신 소프트웨어 라이센스, 애플리케이션 소프트웨어 라이선스를 포함한다. $\psi_0 \mu_0^{d_0} + \psi_G \mu_G^{d_G}$는 동적 소비 전력 총합이다.

수학식 6

[0047] $$\mathbb{C} = N_0 \left(\phi_0 + \psi_0 \mu_0^{d_0} + \psi_G \mu_G^{d_G} \right) + \sum_{\forall j \in J, j \neq 0} N_j \left(\phi_j + \psi_j \mu_j^{d_j} \right)$$

[0048] 수학식 6에서 괄호를 Γ_0과 Γ_j으로 치환한다.

수학식 7

$$\mathbb{C} = N_0\Gamma_0 + \sum_{\forall j \in J, j \neq 0} N_j\Gamma_j$$

[0049]

[0050] 클라우드 모델에서 전처리(pre-processing)와 스케줄링(scheduling)을 합쳐서 준비작업(groundwork)이라고 한다. 준비작업의 서비스 시간은 파라미터 μ_G를 갖는 지수적 분포로 가정할 수 있다. 개인 클라우드의 모든 가상 머신은 필요한 태스크를 수행하고 남은 필요한 태스크를 수행하여 할당된 작업을 완료할 수 있다. 준비작업의 서비스 시간은 남은 태스크를 완료하는데 필요한 서비스 시간과 비교하여 상대적으로 적게 소요되는 것으로 가정한다.

[0051] 도착하는 입력 데이터 트래픽은 푸아송 분포를 따르므로, 개인 클라우드의 준비작업은 M/M/N_0 큐 모델로 모델링 될 수 있다. 준비작업의 지연 τ_G는 수학식 8과 같이 표현된다.

수학식 8

$$\tau_G = \frac{C\left(N_0, \frac{\lambda}{\mu_G}\right)}{N_0\mu_G - \lambda} + \frac{1}{\mu_G}$$

[0052]

[0053] C()는 얼랑-C 공식으로 정의되며 수학식 7과 같이 표현될 수 있다.

수학식 9

$$C\left(N_0, \frac{\lambda}{\mu_G}\right) = \frac{1}{1+(1-\rho_G)\left(\frac{N_0!}{(N_0\rho_G)^{N_{Pri}}}\right)\sum_{k=0}^{N_0-1}\frac{(N_0\rho_G)^k}{k!}}$$

[0054]

[0055] 가상 머신 사용 ρ_G는 1보다 작아야 하고, 큐는 무한대로 증가한다.

수학식 10

$$\frac{\lambda}{\mu_G} < N_0$$

[0056]

[0057] 준비작업이 완료되면, 개인 클라우드는 필요에 따라 전처리 데이터를 공공 클라우드로 전송한다. 개인 클라우드로부터 j 번째 공공 클라우드로 데이터를 전송하여 발생하는 전송 지연은 수학식 11과 같이 표현된다.

수학식 11

$$T_{0j} = \frac{L\alpha_j}{R_0}$$

[0058]

[0059] 지연 시간은 수학식 12와 같이 표현된다.

수학식 12

$$\tau_{0j} = T_{equip} + T_{prop} + \frac{L\alpha_j}{R_0}$$

[0060]

[0061] 네트워크 처리 후 공공 클라우드는 남은 태스크를 처리한다.

[0062] (정리 1) j 번째 공공 클라우드로의 데이터 도착은 평균 속도 $\omega_j\lambda$를 갖는 푸아송 분포를 따른다. 각 공공 클라우드는 $M/M/N_j$ 큐 모델로 모델링할 수 있다. j 번째 공공 클라우드에서 실행 시간은 수학식 13과 같이 표현된다.

수학식 13

$$\tau_j = \frac{C\left(N_j, \frac{\omega_j\lambda}{\mu_j}\right)}{N_j\mu_j - \omega_j\lambda} + \frac{1}{\mu_j}$$

[0063]

[0064] C()는 얼랑-C 공식으로 정의되며 수학식 14와 같이 표현된다.

수학식 14

$$C\left(N_j, \frac{\omega_j\lambda}{\mu_j}\right) = \frac{1}{1+(1-\rho_j)\left(\frac{N_j!}{(N_j\rho_j)^{N_j}}\right)\sum_{k=0}^{N_j-1}\frac{(N_j\rho_j)^k}{k!}}$$

[0065]

[0066] j 번째 공공 클라우드에서 남은 태스크를 완료하면, j 번째 공공 클라우드는 출력 데이터를 개인 클라우드로 역전송한다. 전송 지연을 고려한 지연 시간은 수학식 15와 같이 표현된다.

수학식 15

$$\tau_{j0} = \frac{L\alpha_j\beta_j}{R_j} + T_{equip} + T_{prop}.$$

[0067]

[0068] j 번째 공공 클라우드를 사용한 남은 태스크를 처리하는 데 필요한 시간은 수학식 16과 같이 표현된다.

수학식 16

[0069] $$D_j = \tau_G + \tau_j + \tau_N$$

[0070] $\tau_N = \tau_{0j} + \tau_{j0}$을 의미한다.

[0071] 개인 클라우드를 사용할 때에도 준비작업이 필요하다.

수학식 17

[0072] $$\tau_0 = \frac{C\left(N_0, \frac{\omega_0 \lambda}{\mu_0}\right)}{N_0\mu_0 - \omega_0\lambda} + \frac{1}{\mu_0}$$

[0073] C()는 얼랑-C 공식으로 정의되며 수학식 18과 같이 표현된다.

수학식 18

[0074] $$C\left(N_0, \frac{\omega_0 \lambda}{\mu_0}\right) = \frac{1}{1+(1-\rho_0)\left(\frac{N_0!}{(N_0\rho_0)^{N_0}}\right)\sum_{k=0}^{N_0-1}\frac{(N_0\rho_0)^k}{k!}}$$

[0075] 개인 클라우드에서 $\tau_N = 0$이다.

수학식 19

[0076] $$D_0 = \tau_G + \tau_0.$$

[0077] 최소 비용 하이브리드 클라우드 컴퓨팅 방식은 지연 임계치 D^* 범위에서 작업을 완료해야 한다. 해결과제 P1의 목적은 지연 임계치를 만족하는 최소화된 가상 머신의 개수 $N_0^*, N_1^*, \cdots, N_j^*$ 를 찾는 것이다.

[0078] **(P1)**

$$\min_{\boldsymbol{N}} \quad \mathbb{C}$$

$$\text{Subject to} \quad \begin{aligned} D_0 &\leq D^* \\ D_1 &\leq D^* \\ &\vdots \\ D_M &\leq D^* \end{aligned}$$

[0079] 벡터 $N = (N_0, N_1, \ldots, N_M)$이다.

[0080] 수학식 8, 13, 17에서 얼랑-C 공식이 미분 가능하지 않으므로, 해결과제 P1에 KKT 조건을 적용할 수 없다. 본 실시예에 따른 동적 자원 할당 장치는 얼랑-C 공식에 상한을 설정한다. 수학식 9, 14, 18에 대해서

$C\left(N_0, \frac{\lambda}{\mu_G}\right) \leq U_G$, $C\left(N_j, \frac{\omega_j \lambda}{\mu_j}\right) \leq U_j$, $C\left(N_0, \frac{\omega_0 \lambda}{\mu_0}\right) \leq U_0$ 상한을 설정하면 수학식 20, 21, 22와 같이 표현된다.

수학식 20

[0081]
$$U_G = 1 + \frac{N_0(1-\rho_G)^2}{2\rho_G} - \frac{1-\rho_G}{2\rho_G}\sqrt{4N_0\rho_G + N_0^2(1-\rho_G)^2}$$

수학식 21

[0082]
$$U_0 = 1 + \frac{N_0(1-\rho_0)^2}{2\rho_0} - \frac{1-\rho_0}{2\rho_0}\sqrt{4N_0\rho_0 + N_0^2(1-\rho_0)^2}$$

수학식 22

[0083]
$$U_j = 1 + \frac{N_j(1-\rho_j)^2}{2\rho_j} - \frac{1-\rho_j}{2\rho_j}\sqrt{4N_j\rho_j + N_j^2(1-\rho_j)^2}$$

[0084] 수학식 20, 21, 22를 미분하면 수학식 23, 24, 25와 같이 표현된다.

수학식 23

[0085]
$$\frac{\partial U_G}{\partial N_0} = \frac{3-2\rho_G}{2\rho_G} - \sqrt{4N_0\rho_G + N_0^2(1-\rho_G)^2} - \frac{N_0(1-\rho_G)^2}{2\rho_G\sqrt{4N_0\rho_G + N_0^2(1-\rho_G)^2}}$$

수학식 24

[0086]
$$\frac{\partial U_0}{\partial N_0} = \frac{3-2\rho_0}{2\rho_0} - \sqrt{4N_0\rho_0 + N_0^2(1-\rho_0)^2} - \frac{N_0(1-\rho_0)^2}{2\rho_0\sqrt{4N_0\rho_0 + N_0^2(1-\rho_0)^2}}$$

수학식 25

$$\frac{\partial U_j}{\partial N_j} = \frac{3-2\rho_j}{2\rho_j} - \sqrt{4N_j\rho_j + N_j^2(1-\rho_j)^2} - \frac{N_j(1-\rho_j)^2}{2\rho_j\sqrt{4N_j\rho_j + N_j^2(1-\rho_j)^2}}$$

[0087]

[0088] 지연 시간에 얼랑-C 공식의 상한을 적용한 지연 상한을 수학식 26, 27, 28과 같이 표현할 수 있다.

수학식 26

$$\neg_G = \frac{U_G}{N_0\mu_G - \lambda} + \frac{1}{\mu_G}$$

[0089]

수학식 27

$$\neg_0 = \frac{U_0}{N_0\mu_0 - \omega_0\lambda} + \frac{1}{\mu_0}$$

[0090]

수학식 28

$$\neg_j = \frac{U_j}{N_j\mu_j - \omega_j\lambda} + \frac{1}{\mu_j}$$

[0091]

[0092] 벡터 g는 수학식 29와 같이 표현된다.

수학식 29

[0093] $$\boldsymbol{g} = [g_0, g_1, \cdots, g_M]^T$$

수학식 30

[0094] $$g_0 = \neg_G + \neg_0 - D^*$$

수학식 31

[0095] $$g_{\forall j \in J, j \neq 0} = \gamma_G + \gamma_j + \tau_N - D^*.$$

[0096] 수학식 29, 30, 31을 통해 모든 j에 대한 부등식 $D_j - D^* \leq g_j$ 을 설정할 수 있다. g는 미분 가능하므로, 해결과제 P1을 해결과제 P2로 치환할 수 있다. 해결과제 P2에 의해 3 개의 정리가 성립될 수 있다.

(P2)

$$\begin{aligned} \min_{N} \quad & \mathbb{C} \\ \text{Subject to} \quad & g_0 \leq 0 \\ & g_1 \leq 0 \\ & \vdots \\ & g_M \leq 0 \end{aligned}$$

[0097]

[0098] (정리 2) 해결과제 P2를 풀면서 찾은 해는 해결과제 P1의 제약조건을 만족시키는 데 충분하다.

[0099] (정리 3) 해결과제 P2의 부등식 제약조건은 모두 컨벡스(convex)하다.

[0100] $\rho_j = \frac{\omega_j \lambda}{N_j \mu_j}$ 를 참조하면 γ_j 는 수학식 32 및 수학식 33과 같이 표현되고, 분수를 분해하면 수학식 34와 같이 표현된다.

수학식 32

[0101] $$\gamma_j = \frac{1 + \frac{\left(N_j - \frac{\omega_j \lambda}{\mu_j}\right)^2}{2\frac{\omega_j \lambda}{\mu_j}} - \frac{N_j - \frac{\omega_j \lambda}{\mu_j}}{2\frac{\omega_j \lambda}{\mu_j}} \sqrt{4\frac{\omega_j \lambda}{\mu_j} + \left(N_j - \frac{\omega_j \lambda}{\mu_j}\right)^2}}{\mu_j \left(N_j - \frac{\omega_j \lambda}{\mu_j}\right)}.$$

수학식 33

[0102] $$\gamma_j = \frac{1 + \frac{(N_j - \delta)^2}{2\delta} - \frac{N_j - \delta}{2\delta} \sqrt{4\delta + (N_j - \delta)^2}}{\mu_j (N_j - \delta)}$$

수학식 34

[0103] $$\gamma_j = \left(\frac{1}{\mu_j}\right) \left\{ \frac{1}{N_j - \delta} + \frac{N_j - \delta}{2\delta} - \frac{\sqrt{4\delta + (N_j - \delta)^2}}{2\delta} \right\}$$

[0104] (정리 4) KKT를 사용하여 최적화된 값을 찾으면, $N^*_{\forall j \in J}$는 해결과제 P2의 최적화된 전역 해(global optimal solutions)이다.

[0105] 해결과제 P2에 대한 KKT 조건은 수학식 35, 36, 37과 같이 표현된다.

수학식 35

[0106] $$\boldsymbol{\pi}^* \geq 0$$

수학식 36

[0107] $$\nabla \mathbb{C}\left(\boldsymbol{N}^*\right) + \nabla \boldsymbol{g}\left(\boldsymbol{N}^*\right) \cdot \boldsymbol{\pi}^* = 0$$

수학식 37

[0108] $$\boldsymbol{\pi}^{*T} \cdot \boldsymbol{g}\left(\boldsymbol{N}^*\right) = 0$$

[0109] *는 해를 나타내고, π는 $\boldsymbol{\pi} = \left[\pi_0, \pi_1, \cdots, \pi_M\right]^T$로 정의된 KKT의 곱셈 벡터이다.

[0110] 해의 타당성은 수학식 38과 같이 표현된다.

수학식 38

[0111] $$\boldsymbol{g}\left(\boldsymbol{N}^*\right) \leq 0$$

수학식 39

[0112] $$\nabla \mathbb{C}\left(\boldsymbol{N}\right) = \left[\Gamma_0, \Gamma_1, \cdots, \Gamma_M\right]$$

수학식 40

$$\nabla \boldsymbol{g}(\boldsymbol{N}) = \begin{pmatrix} \frac{\partial U_G}{\partial N_0}(N_0\mu_G-\lambda)-U_G\mu_G}{(N_0\mu_G-\lambda)^2} + \frac{\frac{\partial U_0}{\partial N_0}(N_0\mu_0-\omega_0\lambda)-U_0\mu_0}{(N_0\mu_0-\omega_0\lambda)^2} \\ \frac{\frac{\partial U_G}{\partial N_0}(N_0\mu_G-\lambda)-U_G\mu_G}{(N_0\mu_G-\lambda)^2} \\ \vdots \\ \frac{\frac{\partial U_G}{\partial N_0}(N_0\mu_G-\lambda)-U_G\mu_G}{(N_0\mu_G-\lambda)^2} \end{pmatrix}$$

$$\begin{pmatrix} 0 & \cdots & 0 \\ \frac{\frac{\partial U_1}{\partial N_1}(N_1\mu_1-\omega_1\lambda)-U_1\mu_1}{(N_1\mu_1-\omega_1\lambda)^2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\frac{\partial U_M}{\partial N_M}(N_M\mu_M-\omega_M\lambda)-U_M\mu_M}{(N_M\mu_M-\omega_M\lambda)^2} \end{pmatrix}$$

[0113]

수학식 41

$$\Gamma_j = -\pi_0 \left(\frac{\frac{\partial U_G}{\partial N_0}(N_0\mu_G-\lambda)-U_G\mu_G}{(N_0\mu_G-\lambda)^2} \right) \\ -\pi_j \left(\frac{\frac{\partial U_j}{\partial N_j}(N_j\mu_j-\omega_j\lambda)-U_j\mu_j}{(N_j\mu_j-\omega_j\lambda)^2} \right) \quad for\ \forall j$$

[0114]

[0115] 수학식 37의 조건을 만족하려면 수학식 42를 만족해야 한다.

수학식 42

$$\pi_0 (\gamma_G+\gamma_0-D^*) + \sum_{j=1}^{M} \pi_j (\gamma_G+\gamma_j+\tau_N-D^*) = 0$$

[0116]

[0117] 준비작업의 서비스 시간은 남은 태스크를 완료하는데 필요한 서비스 시간과 비교하여 상대적으로 적게 소요된다.

수학식 43

$$\gamma_G \ll \gamma_j \quad for\ \forall j$$

[0118]

[0119] $\beth_G$는 임의의 작은 수 $\epsilon \ll D^*$ 보다 작다.

수학식 44

[0120] $$\beth_0 \approx D^* - \epsilon$$

수학식 45

[0121] $$\beth_j \approx D^* - \tau_N - \epsilon \; for \; j \neq 0$$

[0122] 이러한 과정을 거쳐 해는 수학식 46과 같이 표현된다.

수학식 46

[0123] $$N_j^* = \frac{\zeta_j}{3} - \frac{6\nu_j - \nu_j^2\eta_j^2}{3\zeta_j} + \frac{1}{3}\nu_j\left(\eta_j + 3\right) \;\; for \; \forall j$$

수학식 47

[0124] $$\nu_j = \frac{\omega_j \lambda}{\mu_j} \; for \; \forall j$$

수학식 48

[0125] $$\eta_j = \begin{cases} \mu_0 D^* - \mu_0\epsilon - 1 & for \; j = 0 \\ \mu_j D^* - \mu_j\tau_N - \mu_j\epsilon - 1 & for \; j \neq 0 \end{cases}$$

수학식 49

[0126] $$\zeta_j = \frac{\sqrt[3]{2\nu_j^3\eta_j^6 - 18\nu_j^2\eta_j^4 + 3\sqrt{3}\sqrt{27\nu_j^2\eta_j^4 - 4\nu_j^3\eta_j^6} + 27\nu_j\eta_j^2}}{\sqrt[3]{2}\eta_j}$$
$$for \; \forall j.$$

[0127] 클라우드 컴퓨팅에서 자동 스케일링 방식을 적용할 수 있다.

[0128] m_j는 j 번째 클라우드에서 활성화된 가상 머신의 개수이다.

[0129] k_j는 j 번째 클라우드에서 태스크의 개수이다.

[0130] (m_j, k_j)는 j 번째 클라우드의 상태이다.

[0131] $[a(m_j), b(m_j)]$는 j 번째 클라우드의 상태를 결정하는 정수이다. $b(m_j) > a(m_j) >= m-1$, $b(m_j) >= a((m+1)_j)$, $a(1_j) < a(2_j) < a(3_j) < \ldots$, $b(1_j) < b(2_j) < b(3_j) < \ldots$ 를 만족한다.

[0132] 시계열적인 분석은 주기적인 워크로드에 적합하지 않다. 예컨대, IoT 장치는 밤시간보다 낮시간에 훨씬 더 많이 클라우드에 접근한다. 이러한 주기적인 워크로드를 고려할 수 있도록 지표 $o(m_j)$와 $u(m_j)$를 이용한 예상 초과 상태(likely-over-provisioned state) 및 예상 부족 상태(likely-under-provisioned state)를 더 포함하는 상태 조건을 설정한다.

수학식 50

[0133]
$$\text{State} = \begin{cases} Likely-over-provisioned, & if\ o(m_j) \le m_j \\ Over-provisioned, & if\ k_j \le a(m_j) \\ Normal, & if\ a(m_j) < k_j \le b(m_j) \\ Under-provisioned, & if\ b(m_j) < k_j \\ Likely-under-provisioned, & if\ m_j < u(m_j) \end{cases}$$

[0134] 지표 $o(m_j)$와 $u(m_j)$는 $o(m_j) = L_o N_j^*$ 및 $u(m_j) = L_\nu N_j^*$ 으로 설정되고, L_o 및 L_ν는 조절 가능한 임의의 양의 실수이다. $o(m_j)$와 $u(m_j)$는 주기적 워크로드 상황에서 활성화된 IoT 장치의 개수를 예측하는 데 적합하다.

[0135] 도 3은 본 발명의 다른 실시예에 따른 하이브리드 클라우드의 동적 자원 할당 방법을 예시한 블록도이다. 하이브리드 클라우드의 동적 자원 할당 방법은 하이브리드 클라우드의 동적 자원 할당 장치에 의해 수행될 수 있다.

[0136] 단계 S10에서, 동적 자원 할당 방법은 데이터 트래픽을 입력받는다.

[0137] 단계 S20에서, 동적 자원 할당 방법은 클라우드의 상태 조건이 정상 상태에 해당하는지 여부를 판단한다. 상태 조건이 정상 상태에 해당하지 않으면, 동적 자원 할당 방법은 하이브리드 클라우드에서 활성화된 가상 머신의 개수 및 태스크의 개수를 체크한다(S30).

[0138] 단계 S40에서, 동적 자원 할당 방법은 클라우드의 상태 조건이 예상 초과 상태(likely-over-provisioned state)에 해당하는지 여부를 판단한다. 상태 조건이 예상 초과 상태에 해당하면, 동적 자원 할당 방법은 대기 시간 이후에, 얼랑-C 공식 모델링하고 상한을 적용하고 상한에 미분을 적용하고 KKT 조건을 적용하여 최적화된 개수를 할당한다(S50).

[0139] 단계 S60에서, 동적 자원 할당 방법은 클라우드의 상태 조건이 예상 부족 상태(likely-under-provisioned state)에 해당하는지 여부를 판단한다. 상태 조건이 예상 부족 상태에 해당하면, 동적 자원 할당 방법은 즉각적으로, 얼랑-C 공식 모델링하고 상한을 적용하고 상한에 미분을 적용하고 KKT 조건을 적용하여 최적화된 개수를 할당한다(S70).

[0140] 단계 S80에서, 동적 자원 할당 방법은 클라우드의 상태 조건이 초과 상태(over-provisioned state) 또는 부족 상태(under-provisioned state)에 해당하는지 여부를 판단한다. 상태 조건이 초과 상태 또는 부족 상태에 해당하면, 동적 자원 할당 방법은 상태 조건이 정상 상태가 될 때까지, CTMC 모델을 적용하여 최적화된 개수를 할당한다(S90).

[0141] 동적 자원 할당 방법을 슈도 코드로 표현하면 표 1과 같다.

표 1

Algorithm 1. MARS

1) **INPUT** data traffic λ
2) **IF** state (m_j, k_j) is not in normal conditions
3) **CHECK** size of m_j and k_j
4) **IF** in the likely-over-provisioned state
5) **THEN ALLOCATE** VMs corresponding to the number of $N_j^* = \frac{\zeta_j}{3\eta_j} - \frac{6\nu_j - \nu_j^2\eta_j^2}{3\zeta_j} + \frac{1}{3}\nu_j(\eta_j + 3)$ to the jth cloud, for $\forall j$ **AFTER** $\frac{\alpha_j \lambda}{\mu_j N_j}$
6) **ELSE IF** in the *likely-over-provisioned* state
7) **THEN ALLOCATE** VMs corresponding to the number of $N_j^* = \frac{\zeta_j}{3\eta_j} - \frac{6\nu_j - \nu_j^2\eta_j^2}{3\zeta_j} + \frac{1}{3}\nu_j(\eta_j + 3)$ to the jth cloud, for $\forall j$
8) **IF** not in the normal state
9) **START** CTMC model until the state returns to normal state
10) **END**

[0142]

[0143] 도 4 내지 도 7은 본 발명의 실시예들에 따라 수행된 모의실험 결과를 도시한 것이다.

[0144] 클라우드 컴퓨팅의 자원 할당의 탄력성 측정은 수학식 51를 이용한다.

수학식 51

[0145]
$$\boldsymbol{E} = \frac{S_{normal}}{S_{total}} = 1 - \frac{S_{over} + S_{under}}{S_{total}}$$

[0146] 도 4 내지 도 7에 도시된 바와 같이, 본 실시예에 따른 동적 자원 할당 모델(MARS)이 CTMC 모델보다 탄력성 측면에서 우수한 효과가 있음을 쉽게 파악할 수 있다.

[0147] 동적 자원 할당 장치는 하드웨어, 펌웨어, 소프트웨어 또는 이들의 조합에 의해 로직회로 내에서 구현될 수 있고, 범용 또는 특정 목적 컴퓨터를 이용하여 구현될 수도 있다. 장치는 고정배선형(Hardwired) 기기, 필드 프로그램 가능한 게이트 어레이(Field Programmable Gate Array, FPGA), 주문형 반도체(Application Specific Integrated Circuit, ASIC) 등을 이용하여 구현될 수 있다. 또한, 장치는 하나 이상의 프로세서 및 컨트롤러를 포함한 시스템온칩(System on Chip, SoC)으로 구현될 수 있다.

[0148] 동적 자원 할당 장치는 하드웨어적 요소가 마련된 컴퓨팅 디바이스 또는 서버에 소프트웨어, 하드웨어, 또는 이들의 조합하는 형태로 탑재될 수 있다. 컴퓨팅 디바이스 또는 서버는 각종 기기 또는 유무선 통신망과 통신을 수행하기 위한 통신 모뎀 등의 통신장치, 프로그램을 실행하기 위한 데이터를 저장하는 메모리, 프로그램을 실행하여 연산 및 명령하기 위한 마이크로프로세서 등을 전부 또는 일부 포함한 다양한 장치를 의미할 수 있다.

[0149] 도 3에서는 각각의 과정을 순차적으로 실행하는 것으로 기재하고 있으나 이는 예시적으로 설명한 것에 불과하고, 이 분야의 기술자라면 본 발명의 실시예의 본질적인 특성에서 벗어나지 않는 범위에서 도 3에 기재된 순서를 변경하여 실행하거나 또는 하나 이상의 과정을 병렬적으로 실행하거나 다른 과정을 추가하는 것으로 다

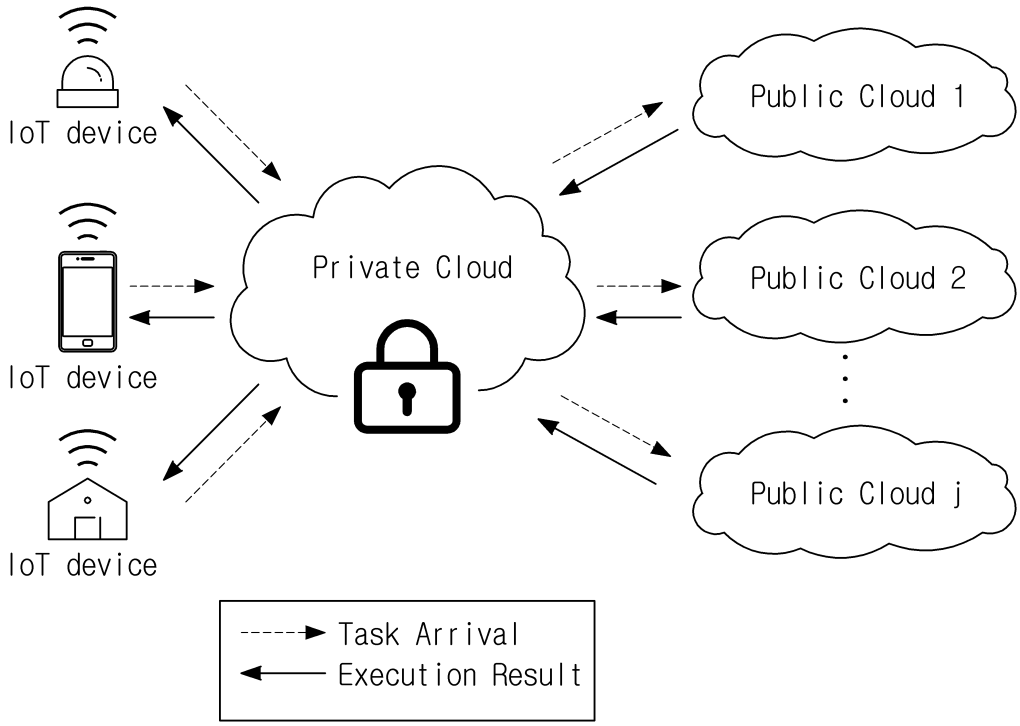
양하게 수정 및 변형하여 적용 가능할 것이다.

[0150] 본 실시예들에 따른 동작은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능한 매체에 기록될 수 있다. 컴퓨터 판독 가능한 매체는 실행을 위해 프로세서에 명령어를 제공하는데 참여한 임의의 매체를 나타낸다. 컴퓨터 판독 가능한 매체는 프로그램 명령, 데이터 파일, 데이터 구조 또는 이들의 조합을 포함할 수 있다. 예를 들면, 자기 매체, 광기록 매체, 메모리 등이 있을 수 있다. 컴퓨터 프로그램은 네트워크로 연결된 컴퓨터 시스템 상에 분산되어 분산 방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수도 있다. 본 실시예를 구현하기 위한 기능적인(Functional) 프로그램, 코드, 및 코드 세그먼트들은 본 실시예가 속하는 기술분야의 프로그래머들에 의해 용이하게 추론될 수 있을 것이다.

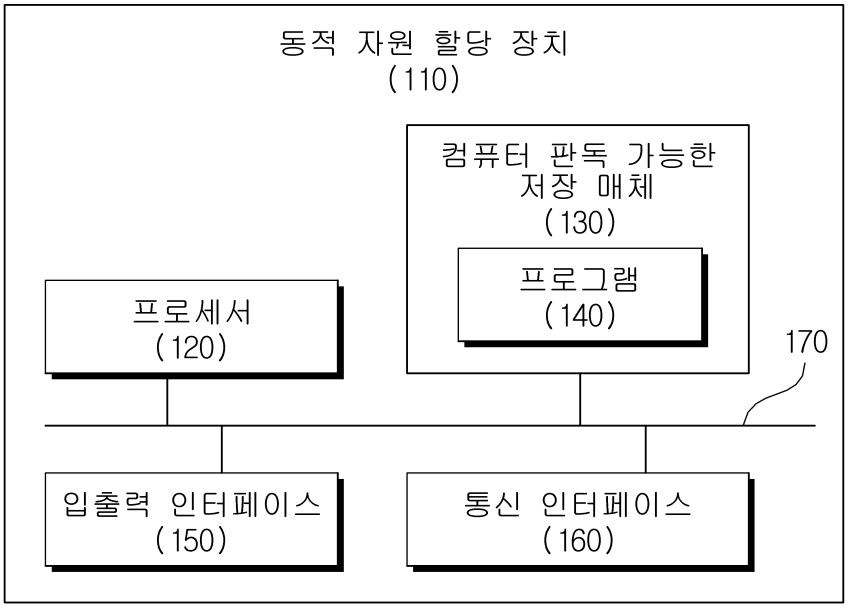
[0151] 본 실시예들은 본 실시예의 기술 사상을 설명하기 위한 것이고, 이러한 실시예에 의하여 본 실시예의 기술 사상의 범위가 한정되는 것은 아니다. 본 실시예의 보호 범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 실시예의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

도면

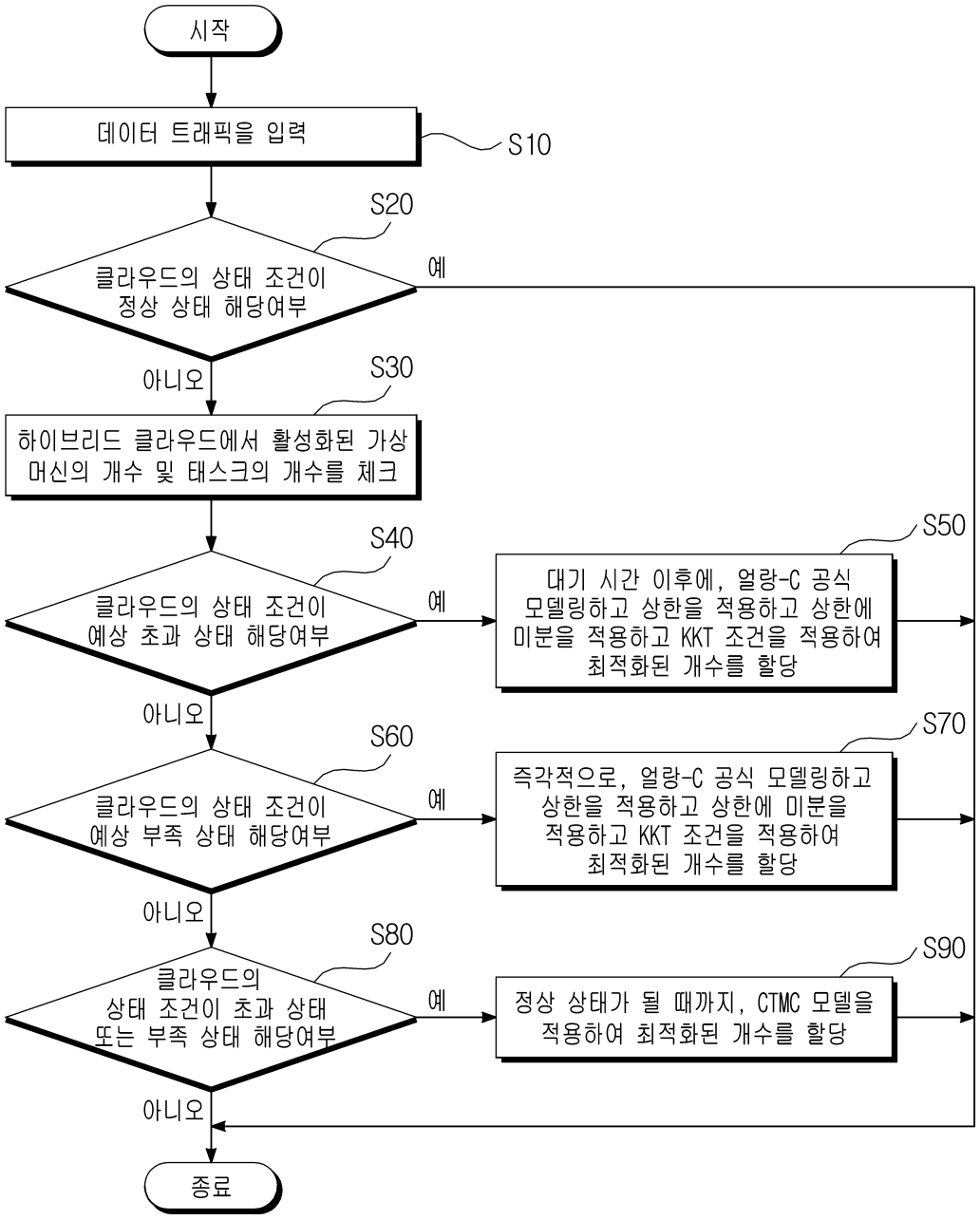
도면1



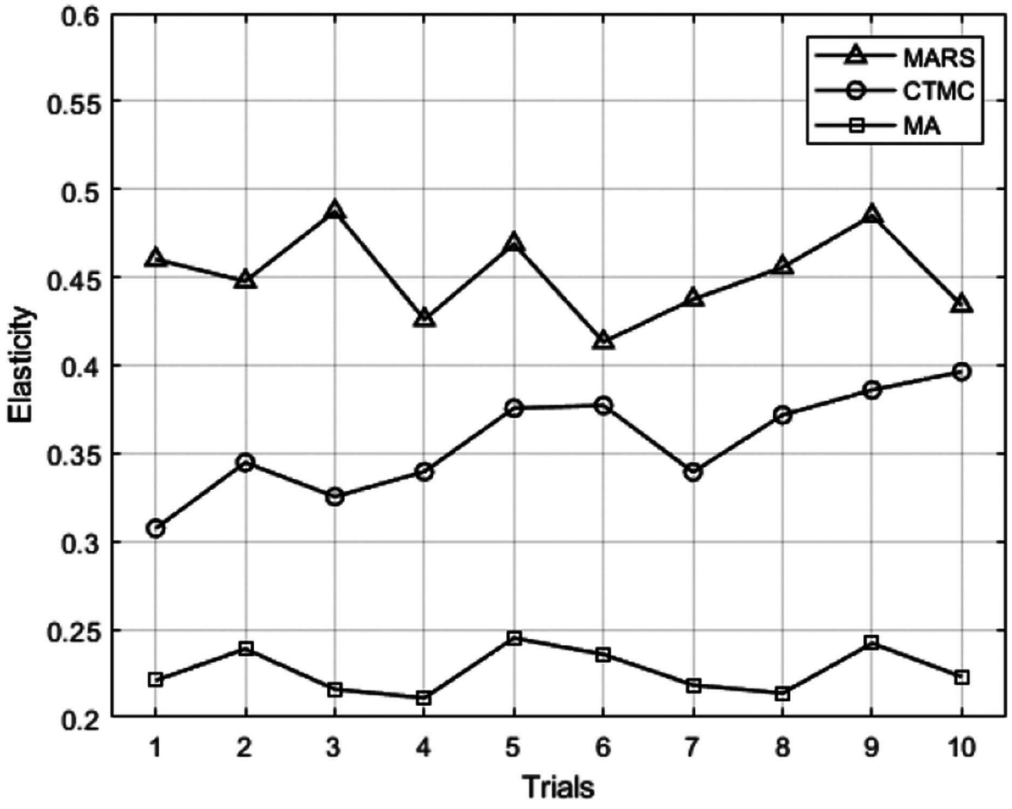
도면2


동적 자원 할당 장치
(110)
컴퓨터 판독 가능한
저장 매체
(130)
프로그램
(140)
프로세서
(120)
170
입출력 인터페이스
(150)
통신 인터페이스
(160)

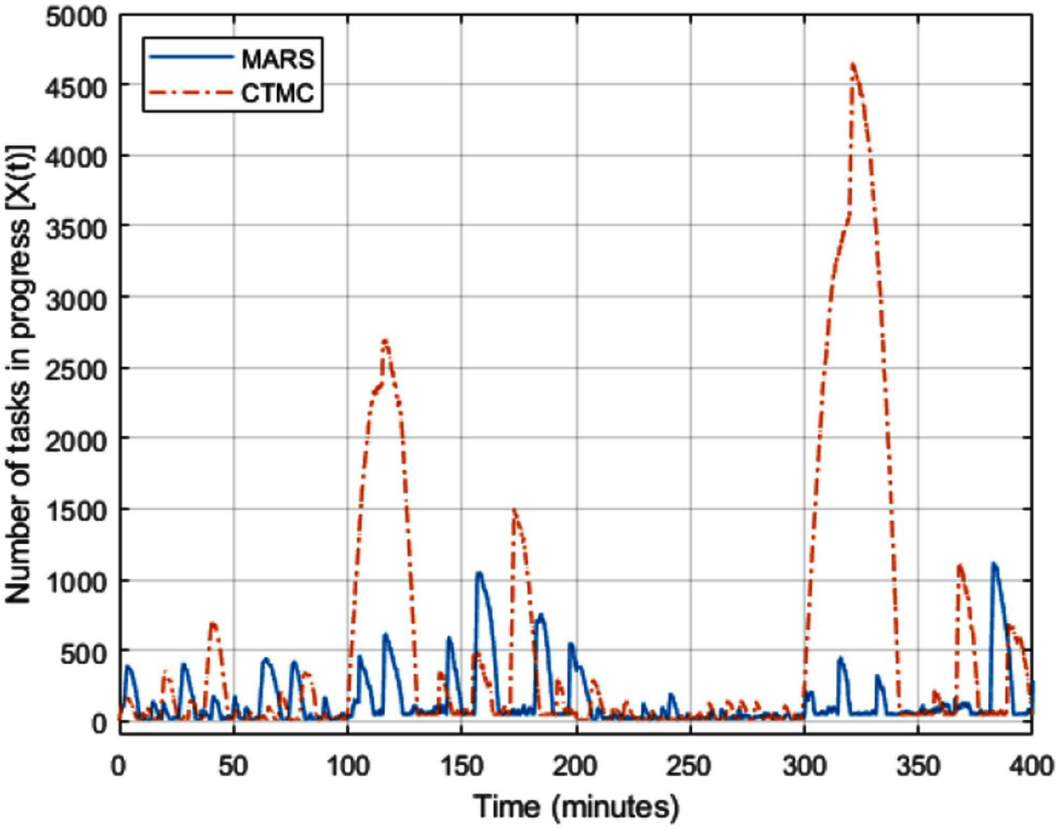
도면3


시작
데이터 트래픽을 입력
S10
S20
클라우드의 상태 조건이 정상 상태 해당여부
예
아니오
S30
하이브리드 클라우드에서 활성화된 가상 머신의 개수 및 태스크의 개수를 체크
S40
클라우드의 상태 조건이 예상 초과 상태 해당여부
예
S50
대기 시간 이후에, 얼랑-C 공식 모델링하고 상한을 적용하고 상한에 미분을 적용하고 KKT 조건을 적용하여 최적화된 개수를 할당
아니오
S60
클라우드의 상태 조건이 예상 부족 상태 해당여부
예
S70
즉각적으로, 얼랑-C 공식 모델링하고 상한을 적용하고 상한에 미분을 적용하고 KKT 조건을 적용하여 최적화된 개수를 할당
아니오
S80
클라우드의 상태 조건이 초과 상태 또는 부족 상태 해당여부
예
S90
정상 상태가 될 때까지, CTMC 모델을 적용하여 최적화된 개수를 할당
아니오
종료

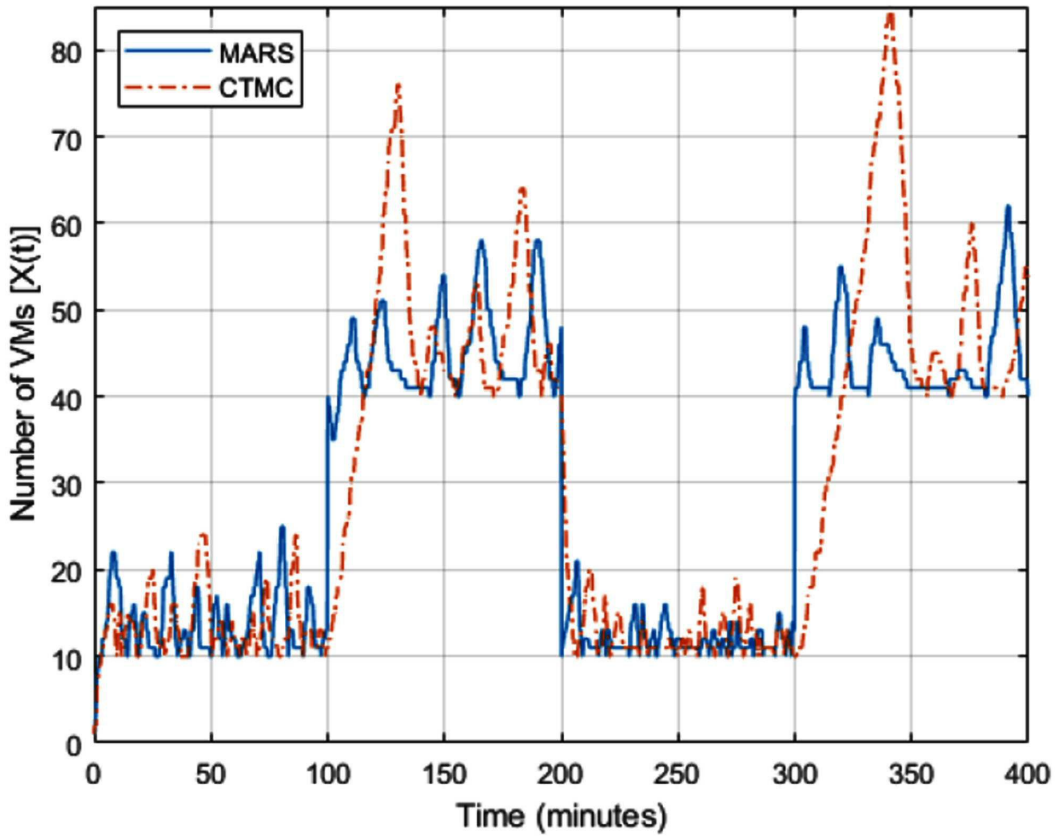
도면4


MARS
CTMC
MA
Elasticity
Trials
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
1
2
3
4
5
6
7
8
9
10

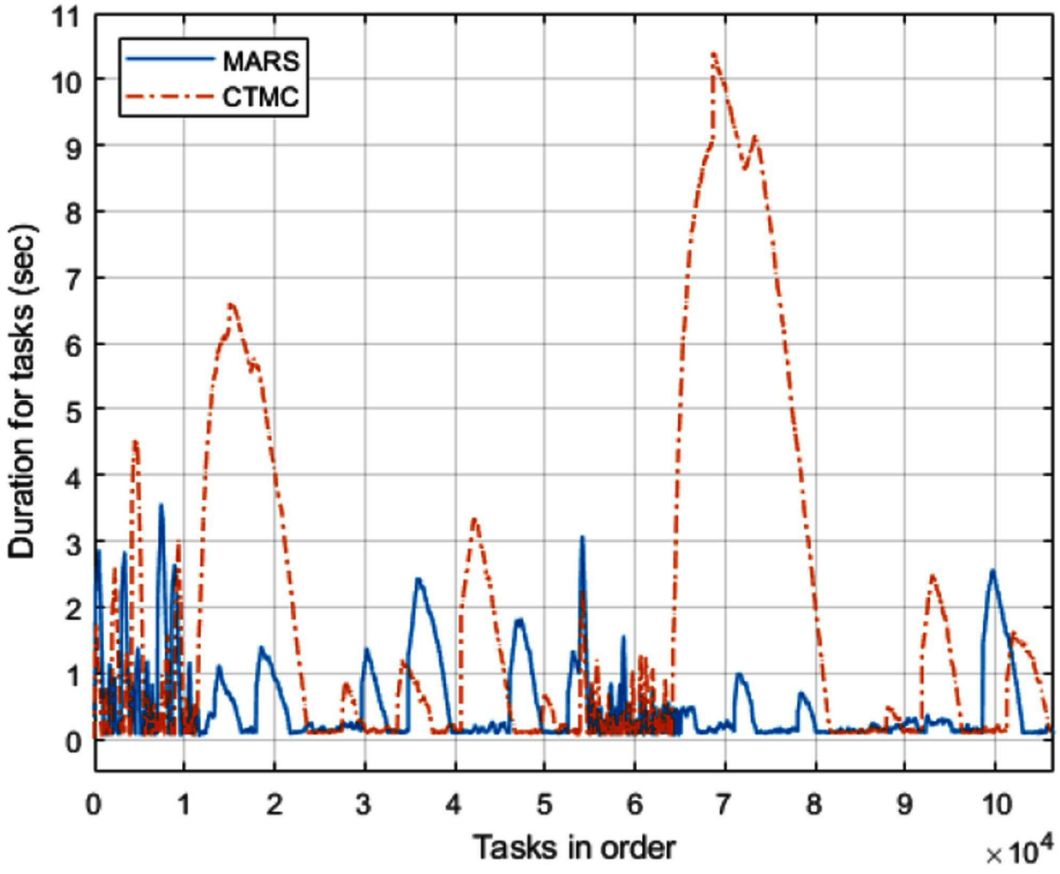
도면5


MARS
CTMC
Number of tasks in progress [X(t)]
Time (minutes)
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250
300
350
400

도면6


MARS
CTMC
Number of VMs [X(t)]
Time (minutes)
0
10
20
30
40
50
60
70
80
0
50
100
150
200
250
300
350
400

도면7


MARS
CTMC
Duration for tasks (sec)
Tasks in order
×10^4
0
1
2
3
4
5
6
7
8
9
10
11