(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0127220
(43) 공개일자 2024년08월22일

(51) 국제특허분류(Int. Cl.)
G06F 11/36 (2006.01) *G06F 11/34* (2006.01)
G06F 8/65 (2018.01) *G06F 8/71* (2018.01)
G06F 9/30 (2018.01) *G06F 9/455* (2018.01)
(52) CPC특허분류
G06F 11/3664 (2013.01)
G06F 11/3457 (2013.01)
(21) 출원번호 10-2023-0026536
(22) 출원일자 2023년02월28일
심사청구일자 2023년02월28일
(30) 우선권주장
1020230020211 2023년02월15일 대한민국(KR)

(71) 출원인
국립한밭대학교 산학협력단
대전광역시 유성구 동서대로 125 (덕명동)
(72) 발명자
최창범
대전광역시 유성구 엑스포로 448 엑스포아파트, 404-902
함규식
대전광역시 유성구 동서대로179번길 62-24, 402호
(뒷면에 계속)
(74) 대리인
특허법인팬브릿지

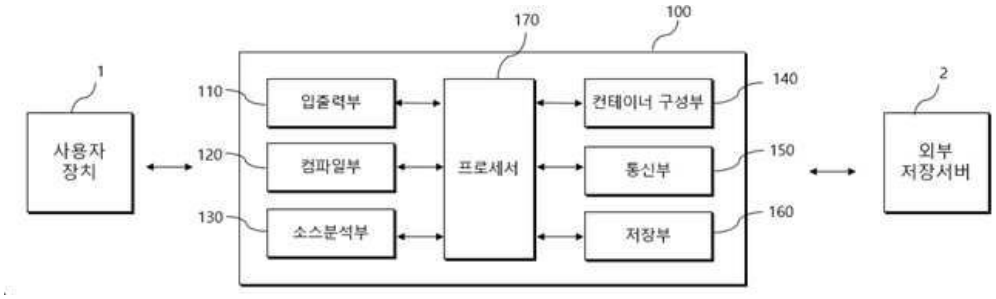
전체 청구항 수 : 총 5 항

(54) 발명의 명칭 **마이크로 서비스 아키텍처 기반의 모델링 및 시뮬레이션 프레임워크**

(57) 요 약

마이크로 서비스 아키텍처 시스템의 제어 방법이 개시된다. 본 명세서는, 시뮬레이션 모델을 포함하는 마이크로 서비스 구현체를 복수의 컴퓨팅 노드에 각각 구성하는 단계; 상기 복수의 컴퓨팅 노드 각각에 상기 마이크로 서비스 아키텍처 구현체의 동작을 제어하는 명령어를 포함하는 스크립트(script)를 저장하는 단계; 및 상기 복수의 컴퓨팅 노드 중 적어도 하나에서 시뮬레이션 모델의 변경 또는 추가되는 상황이 존재하는 경우, 상기 스크립트를 통해 시뮬레이터 업데이트를 수행하도록 제어하는 단계;를 포함하고, 상기 스크립트는 상기 마이크로 서비스 구현체가 중지된 후 재 실행되는 과정에서 업데이트의 실행을 명령하는 명령어를 포함한다. 이에 따라 마이크로 서비스 아키텍처 구현체의의 생성 및 조합을 통하여 초거대 시뮬레이션을 손쉽게 구축 가능하다.

대 표 도 - 도1



(52) CPC특허분류
G06F 11/3688 (2013.01)
G06F 11/3692 (2013.01)
G06F 8/65 (2013.01)
G06F 8/71 (2013.01)
G06F 9/30087 (2013.01)
G06F 9/45558 (2013.01)
G06F 2009/45562 (2013.01)
G06F 2009/45575 (2019.08)
G06F 2009/45591 (2019.08)

(72) 발명자
이재윤
대전광역시 유성구 배울1로 13 대덕테크노밸리2단지아파트, 204-1301

한세영
대전광역시 유성구 학하중앙로67번길 27-15, 301호

이 발명을 지원한 국가연구개발사업

과제고유번호	1345356233
과제번호	2021RIS-004
부처명	교육부
과제관리(전문)기관명	한국연구재단
연구사업명	지자체대학협력기반지역혁신사업
연구과제명	(대전세종충남지역혁신플랫폼)충남대학교
기 여 율	1/1
과제수행기관명	(대전세종충남지역혁신플랫폼)충남대학교
연구기간	2022.05.01 ~ 2023.03.31

명 세 서

청구범위

청구항 1

마이크로 서비스 아키텍처 시스템을 지원하는 컴퓨팅 장치에서 마이크로 서비스 아키텍처 시스템의 제어 방법에 있어서,

시뮬레이션 모델을 포함하는 마이크로 서비스 구현체를 복수의 컴퓨팅 노드에 각각 구성하는 단계;

상기 복수의 컴퓨팅 노드 각각에 상기 마이크로 서비스 아키텍처 구현체의 동작을 제어하는 명령어를 포함하는 스크립트(script)를 저장하는 단계; 및

상기 복수의 컴퓨팅 노드 중 적어도 하나에서 시뮬레이션 모델의 변경 또는 추가되는 상황이 존재하는 경우, 상기 스크립트를 통해 시뮬레이터 업데이트를 수행하도록 제어하는 단계;를 포함하고,

상기 스크립트는 상기 마이크로 서비스 구현체가 중지된 후 재 실행되는 과정에서 업데이트의 실행을 명령하는 명령어를 포함하는 것을 특징으로 하는 마이크로 서비스 아키텍처 시스템의 제어 방법.

청구항 2

제 1 항에 있어서,

상기 복수의 컴퓨팅 노드 중 적어도 하나에서 시뮬레이션 모델의 변경 또는 추가된 경우, 상기 마이크로 서비스 구현체가 재사용되도록 제어하는 단계;

를 더 포함하는 것을 특징으로 하는 마이크로 서비스 아키텍처 시스템의 제어 방법.

청구항 3

제 1 항에 있어서,

상기 스크립트는, 상기 초기 명령어 파일과 업데이트 파일을 포함하고,

상기 초기 명령어 파일은 상기 마이크로 서비스가 초기 구성되는 과정에서 시뮬레이션을 저장소에서 복제하는 명령어를 포함하고,

상기 업데이트 파일은 상기 시뮬레이션 모델의 변경, 새로운 시뮬레이션 모델의 추가 또는 상기 시뮬레이션 모델의 오류 수정이 필요한 경우 상기 시뮬레이션 모델을 업데이트하는 명령어를 포함하는 것을 특징으로 하는 마이크로 서비스 아키텍처 시스템의 제어 방법.

청구항 4

제 3 항에 있어서,

상기 컴퓨팅 장치는, 최신 버전의 스크립트를 저장하는 저장부;를 더 포함하고,

상기 제어 방법은,

상기 복수의 컴퓨팅 노드 중 적어도 하나에서 시뮬레이션 모델의 변경 또는 추가되는 상황이 존재하는 경우, 상기 복수의 컴퓨팅 노드에 포함된 스크립트와 상기 저장부에 포함된 상기 최신 버전의 스크립트를 동기화시키는 단계;

를 더 포함하는 것을 특징으로 하는 마이크로 서비스 아키텍처 시스템의 제어 방법.

청구항 5

하나 이상의 프로세서에 의해 실행되는 복수의 명령들을 포함하는, 컴퓨터- 판독가능 매체에 의해 저장된 컴퓨터 프로그램으로서,

상기 컴퓨터 프로그램은,

시뮬레이션 모델을 포함하는 마이크로 서비스 구현체를 복수의 컴퓨팅 노드에 각각 구성하는 설정하는 명령;

상기 복수의 컴퓨팅 노드 각각에 상기 마이크로 서비스 아키텍처 구현체의 동작을 제어하는 명령어를 포함하는 스크립트(script)를 저장하는 명령; 및

상기 복수의 컴퓨팅 노드 중 적어도 하나에서 시뮬레이션 모델의 변경 또는 추가되는 상황이 존재하는 경우, 상기 스크립트를 통해 시뮬레이터 업데이트를 수행하도록 제어하는 명령;를 포함하고,

상기 스크립트는 상기 마이크로 서비스 구현체가 중지된 후 재 실행되는 과정에서 업데이트의 실행을 명령하는 명령어를 포함하는 것을 특징으로 하는 컴퓨터-판독가능 매체에 의해 저장된 컴퓨터 프로그램.

발명의 설명

기 술 분 야

[0001] 본 명세서는 마이크로 서비스 아키텍처 기반 모델링 및 시뮬레이션 프레임워크에 관한 것이다.

배 경 기 술

[0002] 현대사회의 복잡한 문제를 해결하기 위해서 컴퓨터 시뮬레이션을 활용하고 있다. 컴퓨터 시뮬레이션을 사용하여 문제를 해결하기 위해서는 문제에 대하여 컴퓨터 시뮬레이션 모델로 모델링을 수행하고 모델링 된 시뮬레이션 모델을 조합하여 시뮬레이터를 개발한 후 이를 실행하여 문제를 해결한다. 하지만, 빠르게 변화하는 현대사회와 기술로 인하여 기존에 개발한 시뮬레이션 모델을 재사용하기 어렵거나 시뮬레이터로부터 데이터를 빠르게 획득하기 위하여 시뮬레이터를 구성하는 것이 필요한 상황이다.

발명의 내용

해결하려는 과제

[0003] 전술한 문제를 해결하기 위해 본 명세서는 마이크로 서비스를 구현하기 위한 시뮬레이션 모델을 포함하는 독립 모듈에 있어서, 시뮬레이션 모델의 업데이트가 필요한 경우 마이크로 서비스 아키텍처를 전체적으로 재구성하지 않고, 스크립트 업데이트를 통해 보다 편리하게 마이크로 서비스 아키텍처 시스템을 구현할 수 있는 마이크로 서비스 아키텍처 시스템의 제어 방법을 제공하는 것을 목적으로 한다.

[0004] 본 발명이 이루고자 하는 기술적 과제들은 이상에서 언급한 기술적 과제들로 제한되지 않으며, 언급되지 않은 또 다른 기술적 과제들은 이하의 발명의 상세한 설명으로부터 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0005] 본 명세서의 일 실시예에 따른 마이크로 서비스 아키텍처 시스템의 제어 방법은, 마이크로 서비스 아키텍처 시스템을 지원하는 컴퓨팅 장치에서 수행될 수 있다. 상기 제어 방법은, 시뮬레이션 모델을 포함하는 마이크로 서비스 구현체를 복수의 컴퓨팅 노드에 각각 구성하는 단계; 상기 복수의 컴퓨팅 노드 각각에 상기 마이크로 서비스 아키텍처 구현체의 동작을 제어하는 명령어를 포함하는 스크립트(script)를 저장하는 단계; 및 상기 복수의 컴퓨팅 노드 중 적어도 하나에서 시뮬레이션 모델의 변경 또는 추가되는 상황이 존재하는 경우, 상기 스크립트를 통해 시뮬레이터 업데이트를 수행하도록 제어하는 단계;를 포함하고, 상기 스크립트는 상기 마이크로 서비스 구현체가 중지된 후 재 실행되는 과정에서 업데이트의 실행을 명령하는 명령어를 포함한다.

[0006] 상기 제어 방법은, 상기 복수의 컴퓨팅 노드 중 적어도 하나에서 시뮬레이션 모델의 변경 또는 추가된 경우, 상

기 마이크로 서비스 구현체가 재사용되도록 제어하는 단계;를 더 포함할 수 있다.

[0007] 상기 스크립트는, 상기 초기 명령어 파일과 업데이트 파일을 포함하고, 상기 초기 명령어 파일은 상기 마이크로 서비스가 초기 구성되는 과정에서 시뮬레이션을 저장소에서 복제하는 명령어를 포함하고, 상기 업데이트 파일은 상기 시뮬레이션 모델의 변경, 새로운 시뮬레이션 모델의 추가 또는 상기 시뮬레이션 모델의 오류 수정이 필요한 경우 상기 시뮬레이션 모델을 업데이트하는 명령어를 포함할 수 있다.

[0008] 상기 컴퓨팅 장치는, 최신 버전의 스크립트를 저장하는 저장부;를 더 포함하고, 상기 제어 방법은, 상기 복수의 컴퓨팅 노드 중 적어도 하나에서 시뮬레이션 모델의 변경 또는 추가되는 상황이 존재하는 경우, 상기 복수의 컴퓨팅 노드에 포함된 스크립트와 상기 저장부에 포함된 상기 최신 버전의 스크립트를 동기화시키는 단계;를 더 포함할 수 있다.

[0009] 본 명세서의 다른 실시예에 따른 컴퓨터- 판독가능 매체에 의해 저장된 컴퓨터 프로그램은, 시뮬레이션 모델을 포함하는 마이크로 서비스 구현체를 복수의 컴퓨팅 노드에 각각 구성하는 설정하는 명령; 상기 복수의 컴퓨팅 노드 각각에 상기 마이크로 서비스 아키텍처 구현체의 동작을 제어하는 명령어를 포함하는 스크립트(script)를 저장하는 명령; 및 상기 복수의 컴퓨팅 노드 중 적어도 하나에서 시뮬레이션 모델의 변경 또는 추가되는 상황이 존재하는 경우, 상기 스크립트를 통해 시뮬레이터 업데이트를 수행하도록 제어하는 명령;를 포함하고, 상기 스크립트는 상기 마이크로 서비스 구현체가 중지된 후 재 실행되는 과정에서 업데이트의 실행을 명령하는 명령어를 포함한다.

[0010] 본 명세서의 또 다른 실시예에 따른 컴퓨팅 장치는, 서로 다른 마이크로 서비스를 구현하는 복수의 컴퓨팅 노드(Computing Node), 상기 복수의 컴퓨팅 노드의 동작을 제어하는 매니저 노드(Manager Node), 상기 컴퓨팅 노드와 매니저 모드가 서로 상호 통신이 가능하도록 하는 네트워크 장치, 최신 버전의 스크립트를 저장하고, 상기 매니저 노트와 통신하는 마운티트 볼륨(Mounted Volume)을 포함한다.

[0011] 상기 컴퓨팅 노드는 각각 사전에 배포되거나 실시간으로 추가되는 스크립트(script)를 포함하고 있다.

[0012] 상기 매니저 노드는 상기 복수의 컴퓨팅 노드 중 적어도 하나에서 시뮬레이션 모델의 변경 또는 시뮬레이션 모델의 추가가 존재하는 경우, 마이크로 서비스가 재실행되도록 제어할 수 있다.

[0013] 상기 매니저 노드는 상기 마이크로 서비스가 재실행되는 과정에서 상기 컴퓨팅 장치의 저장부에 저장된 최신버전의 스크립트가 상기 복수의 컴퓨팅 노드 각각에 포함된 스크립트와 자동으로 동기화되도록 할 수 있다.

발명의 효과

[0014] 본 명세서의 일 실시예에 따르면, 문제해결을 위한 시뮬레이션 구성을 위해서는 전문 개발진이 구성되고 해당 시뮬레이션을 구현해야 하나 본 발명을 활용 하면 마이크로 서비스 아키텍처 구현체가 요구하는 입 출력 값 만을 확인하여 편리하게 구성 할 수 있다. 이는 문제해결 당사자가 마이크로 서비스 아키텍처 구현체의의 생성 및 조합을 통하여 초거대 시뮬레이션을 손쉽게 구축 가능하다.

[0015] 또한, 본 명세서의 일 실시예에 따르면, 새로운 시뮬레이션을 구성하는 경우 기 구축된 개발환경과 달리 최신 기술이 접목된 개발환경을 독립적으로 운영할 수 있어 시뮬레이션 개발자에게 개발 자유도 부여 가능하다.

[0016] 본 발명에서 얻을 수 있는 효과는 이상에서 언급한 효과로 제한되지 않으며, 언급하지 않은 또 다른 효과들은 아래의 기재로부터 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0017] 도 1은 본 명세서의 일 실시예에 따라 마이크로 서비스 아키텍처 기반의 모델링 및 시뮬레이션 프레임 워크를 제공하는 시스템 또는 컴퓨팅 장치의 개략적인 블록도이다.

도 2는 본 명세서의 일 실시예에 따른 마이크로 서비스 아키텍처 기반의 모델링 및 시뮬레이션 프레임 워크의 일 예를 설명하기 위한 도면이다.

도 3은 본 명세서의 일 실시예에 따른 마이크로 서비스 아키텍처 시스템의 제어 방법의 흐름도이다.

도 4는 본 명세서의 일 실시예에 따른 마이크로 서비스 아키텍처 시스템의 제어 방법 시뮬레이션 모델 업데이트 과정을 설명하기 위한 도면이다.

도 5는 본 명세서의 일 실시예에 따른 마이크로 서비스 아키텍처 시스템의 제어 방법의 흐름도이다.

도 6은 본 명세서의 일 실시예에 따라 우선 저장소(클라우드, 로컬)에 사용하고자 하는 마이크로 서비스가 존재하는 경우, 사용자 단말, 데이터 베이스, 컴퓨팅 시스템, 노드 사이의 시퀀스 다이어 그램을 나타낸다.

도 7은 본 명세서의 일 실시예에 따라 다중 서버를 이용한 마이크로 서비스 아키텍처 시스템을 구축하는 예이다.

도 8은 본 명세서의 일 실시예에 따라 다중 물리 서버를 이용하여 현실의 네트워크만을 이용하여 마이크로 서비스 아키텍처 시스템을 구축하는 예이다.

발명을 실시하기 위한 구체적인 내용

[0018] 이하, 첨부된 도면을 참조하여 본 명세서에 개시된 실시예를 상세히 설명하되, 도면 부호에 관계없이 동일하거나 유사한 구성요소는 동일한 참조 번호를 부여하고 이에 대한 중복되는 설명은 생략하기로 한다. 이하의 설명에서 사용되는 구성요소에 대한 접미사 "모듈" 및 "부"는 명세서 작성의 용이함만이 고려되어 부여되거나 혼용되는 것으로서, 그 자체로 서로 구별되는 의미 또는 역할을 갖는 것은 아니다. 또한, 본 명세서에 개시된 실시예를 설명함에 있어서 관련된 공지 기술에 대한 구체적인 설명이 본 명세서에 개시된 실시예의 요지를 흐릴 수 있다고 판단되는 경우 그 상세한 설명을 생략한다. 또한, 첨부된 도면은 본 명세서에 개시된 실시예를 쉽게 이해할 수 있도록 하기 위한 것일 뿐, 첨부된 도면에 의해 본 명세서에 개시된 기술적 사상이 제한되지 않으며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

[0019] 제1, 제2 등과 같이 서수를 포함하는 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되지는 않는다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다.

[0020] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다.

[0021] 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다.

[0022] 본 출원에서, "포함한다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0023] 도 1은 본 명세서의 일 실시예에 따라 마이크로 서비스 아키텍처 기반의 모델링 및 시뮬레이션 프레임 워크를 제공하는 시스템 또는 컴퓨팅 장치의 개략적인 블록도이다.

[0024] 도 1을 참조하면, 본 명세서의 일 실시예에 따른 마이크로 서비스 아키텍처 기반의 모델링 및 시뮬레이션 프레임 워크를 제공하는 시스템 또는 컴퓨팅 장치(이하, 컴퓨팅 장치라 함)은 사용자 장치(1) 및/또는 외부 저장서버(2)와 통신 가능하게 구성되며 애플리케이션 개발자가 사용자 장치(10)를 통해 입력한 소스(source) 정보를 이용하여 마이크로 서비스 아키텍처(MicroService Architecture; MSA) 기반의 컨테이너 기반의 컨테이너(container)를 생성하는 역할을 한다.

[0025] 사용자 장치(1)는 컴퓨팅 장치(100)을 이용하여 애플리케이션을 개발하고자 하는 개발자가 사용하기 위한 장치이다. 도 1에서 사용자 장치(1)는 노트북 컴퓨터의 형태로 도시되었으나, 이는 예시적인 것으로서, 사용자 장치(1)는 스마트폰(smartphone) 등 이동 통신 단말기, 개인용 컴퓨터(personal computer), PDA(personal digital assistant), 태블릿(tablet) 컴퓨터, IPTV(Internet Protocol Television) 등을 위한 셋톱박스(set-top box) 등 임의의 컴퓨팅 장치의 형태로 구현될 수도 있다.

[0026] 컴퓨팅 장치(100)는 소스 정보를 입력하고 개발하고자 하는 애플리케이션의 아키텍처 형태 등의 입력정보를 입력함으로써, 애플리케이션의 동작을 위한 컨테이너가 생성될 수 있도록 하는 플랫폼(platform)의 기능을 한다.

[0027] 예를 들어, 컴퓨팅 장치(100)는 컴퓨팅 장치(100)에 접속한 사용자 장치(1)상에 표시되기 위한 관리화면 등 사

용자 인터페이스(user interface)를 제공하는 웹 서버(web server) 또는 애플리케이션 서비스 서버(application service server)의 형태로 구현될 수 있으나, 이에 한정되는 것은 아니다.

[0028] 도 1에서는 설명의 편의를 위하여 개발자가 사용하는 장치인 사용자 장치(1)와 프레임워크 제공 시스템에 해당하는 컴퓨팅 장치(100)가 별개의 장치로 도시되었으며, 본 명세서의 실시예들은 컴퓨팅 장치(100)과 사용자 장치(1)가 별개의 장치인 것으로 상정하여 설명된다. 그러나, 이는 예시적인 것으로서, 다른 실시예에서는 프레임워크 제공을 위한 컴퓨팅 장치(100) 자체가 개발자의 사용자 장치의 형태로 구현될 수도 있음은 물론이다.

[0029] 일 실시예에서, 컴퓨팅 장치(100)는 입출력부(110), 컴파일부(120), 소스 분석부(130), 컨테이너 구성부(140), 통신부(150), 저장부(160), 프로세서(170)를 포함할 수 있다.

[0030] 본 명세서의 일 실시예에 따른 마이크로 서비스 아키텍처 기반의 모델링 및 시뮬레이션 프레임 워크를 제공하는 시스템 또는 컴퓨팅 장치(100)을 구성하는 각각의 부는 반드시 물리적으로 구분되는 별개의 구성요소를 지칭하는 것으로 의도되지 않는다. 즉, 도 1에서 컴퓨팅 장치(100)을 구성하는 각 부는 서로 구분되는 별개의 블록으로 도시되었으나, 이는 시뮬레이션 프레임워크 제공 시스템(100)을 이에 의해 실행되는 동작에 의해 기능적으로 구분한 것이다. 실시예에 따라서는 전술한 각 부 중 일부 또는 전부가 동일한 하나의 장치 내에 집적화될 수 있으며, 또는 하나 이상의 부가 다른 부와 물리적으로 구분되는 별개의 장치로 구현될 수도 있다. 예를들어, 상기 각 부의 구성은 분산 컴퓨팅 환경 하에서 서로 통신 가능하게 연결된 컴포넌트들일 수도 있다.

[0031] 입출력부(110)는 사용자 장치(1)를 통해 개발자의 입력을 수신하고 이에 상응하는 처리 결과를 사용자 장치(1)에 제공하기 위한 구성을 포함한다. 입출력부(110)는 개발자로부터 애플리케이션의 소스 정보를 수신할 수 있다. 상기 소스 정보는 자바(JAVA) 언어에 기반한 애플리케이션의 소스 코드를 예시로 하여 설명될 수 있지만, 프로그래밍 언어의 종류는 이에 한정되는 것은 아니다. 또한, 입출력부(110)는 개발자로부터 애플리케이션에 연관된 컨테이너 생성 요청을 수신할 수 있으며, 이때 컨테이너 생성 요청은 애플리케이션의 아키텍처 형태에 대한 입력정보를 포함할 수 있다.

[0032] 컴파일부(120)는 개발자가 입력한 소스 정보를 컴퓨팅 장치에서 실행 가능한 형태(예컨대, 바이너리 파일 또는 중간언어 파일 등)로 변환하기 위한 부분이다. 이는 공재의 컴파일러(compiler)로부터 통상의 기술자에게 용이하게 이해될 수 있으므로, 자세한 설명은 생략한다.

[0033] 소스 분석부(130)는 컴파일부(120)에 의해 변환되는 소스 정보를 분석하여, 애플리케이션에 포함된 하나 이상의 서비스 사이의 호출 관계에 연관된 관계정보를 생성하는 역할을 한다. 소스 분석부(130)의 동작은 코드 상의 주석을 통해 문서를 생성하는 독시젠(doxygen)이나, 소스 코드 내의 함수 호출을 그래프로 변환하는 그래프비즈(graphviz) 등 공지된 소프트웨어 도구로부터 용이하게 이해될 수 있다.

[0034] 본 발명의 실시예들에서 애플리케이션은 하나 이상의 서비스를 포함할 수 있으며, 각 서비스는 공통의 라이브러리(library)를 제외하고는 상호 간에 직접 호출을 금지하도록 소스 정보가 구성된다. 즉, 애플리케이션의 각 서비스 사이의 호출은 사전에 정의된 입출력 도메인(domain)을 이용한 URL(Uniform Resource Locator) 호출 방식으로 이루어지도록 구성되며, 이에 따라 소스 정보의 분석을 통하여 소스 정보의 각 서비스 사이의 호출 관계가 관계정보로서 생성될 수 있도록 한다.

[0035] 소스 분석부(130)는 생성한 관계정보를 저장부(160)에 저장할 수 있다. 또한, 개발자가 기존의 소스 정보를 수정하여 호출 관계에 변동이 발생하는 경우, 소스 분석부(130)는 새로 생성된 관계정보를 이용하여 저장부(140)에 기 저장된 관계정보를 갱신할 수 있다. 이때 관계정보는, 애플리케이션의 각 서비스와 관련하여 해당 서비스의 동작에 필요한 클래스(class), jsp 파일 등 서비스 정보 및 리소소(resource) 정보들을 정의하는 패키지(package) 정보를 포함할 수 있다.

[0036] 컨테이너 구성부(140)는, 개발자로부터 특정 서비스 또는 기존에 생성된 특정 패키지 정보에 대한 컨테이너 생성 요청에 대한 응답으로, 저장부(160)에 저장된 관계정보를 토대로 컨테이너를 구성하는 역할을 한다. 이때 컨테이너는 컴퓨팅 장치의 운영체제 등과 격리된 환경에서 실행되며, 요청에 상응하는 객체를 생성하고 이를 인자로 한 함수를 호출함으로써 요청에 대한 응답을 생성, 반환하는 등의 기능을 수행하는 것으로서, 예를들어, 도커(docker) 등이 이에 해당한다.

[0037] 본 명세서에서 컨테이너 구성부(140)에 의해 컨테이너를 구성한다는 것은, 새로운 컨테이너를 생성하는 의미할 수도 있고, 또는 패키지 정보와 관계 정보를 기준으로 기존의 컨테이너에 정보를 전송하는 것을 의미할 수도 있다. 이때 관계정보를 기초로 서비스의 동작을 위한 서비스 정보 및 리소스 정보 등이 추출되어 자동으로 컨테이너에 포함된다. 또한, 새로운 컨테이너를 생성하는 의미는 시뮬레이터를 구성하기 위해 복수의 컨테이너가

조합되는 것을 의미할 수도 있다.

[0038] 통신부(150)는, 컨테이너 구성부(140)에 의해 구성되는 컨테이너가 MSA 컨테이너인 경우, 서비스 정보 및 리소스 정보를 외부의 저장 서버(2)에 저장하여 이를 해당 컨테이너의 실행 시 로드할 수 있게 하는 역할을 한다. 예를 들어, 서비스 정보는 클래스 파일, jsp 파일 등을 포함할 수 있으며, 리소스 정보는 서비스의 동작에 필요한 txt, xml 등의 파일을 포함할 수 있다.

[0039] 한편, 외부의 저장서버(2)는 MSA 컨테이너의 서비스 동작을 위한 서비스 및 리소스 정보가 저장된 저장부(미도시)와, MSA 컨테이너의 요청에 대한 응답으로 요청되는 정보를 변환하는 정보 변환 기능모듈(미도시) 등을 포함할 수 있다.

[0040] 도 2는 본 명세서의 일 실시예에 따른 마이크로 서비스 아키텍처 기반의 모델링 및 시뮬레이션 프레임 워크의 일 예를 설명하기 위한 도면이다.

[0041] **문제 해결을 위한 정책 배포를 이용한 유지보수**

[0042] 도 2를 참조하면, 본 명세서는 마이크로 아키텍처 서비스 기반의 경량형 모델링 및 시뮬레이션 프레임워크는 소형 시스템부터 대형 시스템의 구축을 지원한다. 본 명세서에서 제안하는 프레임워크는 네트워크(Network), 매니저 노드(Manager Node), 컴퓨팅 노드(Computing Node), 마운티드 볼륨(Mounted volume)을 포함할 수 있다.

[0043] 매니저 노드와 컴퓨팅 노드는 네트워크를 통하여 서로 상호 통신이 가능하다. 일반 컴퓨팅 노드는 통신이 필요한 시뮬레이터 혹은 프로세서를 포함할 수 있다. 한편, 컴퓨팅 노드는 일반적인 프로세스도 실행 가능하다. 즉, 데이터베이스와 같이 통신이 필요하지 않는 프로세스를 실행하고 있는 마이크로 서비스 아키텍처 구현체를 말한다.

[0044] 각각의 마이크로 서비스 아키텍처 구현체는 사용자에 의해 사전 배포된 정책(Script) 혹은 실시간으로 추가되는 정책(Script)을 보유하고 반영하는 매니저 노드의 실행에 따라 시스템이 동작한다. 이때 배포되는 정책은 마이크로 서비스 아키텍처 구현체가 중지되었다 다시 재 실행되는 과정에서 업데이트를 할 수 있는 명령 스크립트(Script)를 의미할 수 있다. 또한, 스크립트는 노드가 실행됨과 동시에 실행할 명령어를 수정하는 작업을 진행하게 한다. 동작된 시스템의 결과값은 가상환경인 시스템과 현실세계의 Host 환경을 마운트(Mount)하여 현실세계에서 실시간으로 반영되는 결과 값을 확인할 수 있어 가상환경의 상황을 효과적으로 이해하며 유지보수의 효율성을 높일 수 있다.

[0045] 도 3은 본 명세서의 일 실시예에 따른 마이크로 서비스 아키텍처 시스템의 제어 방법의 흐름도이다. 상기 제어 방법은, 도 1에서 개시한 컴퓨팅 장치(100)의 프로세서(170)에 의해 구현될 수 있다.

[0046] 도 3을 참조하면, 마이크로 서비스 아키텍처 시스템을 지원하는 컴퓨팅 장치의 프로세서(170)는 마이크로 서비스 아키텍처 시스템의 제어 과정에서 시뮬레이션 모델을 포함하는 마이크로 서비스 구현체를 복수의 컴퓨팅 노드에 각각 구성할 수 있다(S300).

[0047] 프로세서(170)는 상기 복수의 컴퓨팅 노드 각각에 상기 마이크로 서비스 아키텍처 구현체의 동작을 제어하는 명령어를 포함하는 스크립트(script)를 저장할 수 있다(S310).

[0048] 프로세서(170)는 상기 복수의 컴퓨팅 노드 중 적어도 하나에서 시뮬레이션 모델의 변경 또는 추가되는 상황이 존재하는 경우(S320), 상기 스크립트를 통해 시뮬레이터 업데이트를 수행하도록 제어할 수 있다(S330). 여기서, 상기 스크립트는 상기 마이크로 서비스 구현체가 중지된 후 재 실행되는 과정에서 업데이트의 실행을 명령하는 명령어를 포함할 수 있다. 상기 스크립트는, 상기 초기 명령어 파일과 업데이트 파일을 포함할 수 있다.

[0049] 상기 초기 명령어 파일은 상기 마이크로 서비스가 초기 구성되는 과정에서 시뮬레이션을 저장소에서 복제하는 명령어를 포함하고, 상기 업데이트 파일은 상기 시뮬레이션 모델의 변경, 새로운 시뮬레이션 모델의 추가 또는 상기 시뮬레이션 모델의 오류 수정이 필요한 경우 상기 시뮬레이션 모델을 업데이트하는 명령어를 포함할 수 있다.

[0050] 도 4는 본 명세서의 일 실시예에 따른 마이크로 서비스 아키텍처 시스템의 제어 방법 시뮬레이션 모델 업데이트 과정을 설명하기 위한 도면이다.

[0051] 도 4를 참조하면, 일 예에 따른 마이크로 서비스 A에서 시뮬레이션 모델 A가 시뮬레이션 모델 A'으로 변경된 경우, 마이크로 서비스가 재실행될 때, 스크립트가 최신 버전의 스크립트와 동기화되도록 제어될 수 있다. 프로세서(170)는 마이크로 서비스 A를 재실행함으로써, 저장부(repository)에 저장된 최신 버전의 스크립트의 업데이

트를 자동으로 수행되도록 제어한다. 이에 따라 변경된 시뮬레이션 모델 A'는 최신 버전의 스크립트에 의해 동작이 제어될 수 있다.

[0052] 도 5는 본 명세서의 일 실시예에 따른 마이크로 서비스 아키텍처 시스템의 제어 방법의 흐름도이다.

[0053] 도 5를 참조하면, 프로세서(170)는 마이크로 서비스 시스템을 구성하기 위해 시스템 구성을 위한 마이크로 서비스가 존재하는지 확인한다(Exist Micro Service?). 상기 마이크로 서비스는 시뮬레이터 혹은 DB와 같은 프로세스를 의미할 수 있고, 마이크로 서비스는 컨테이너 혹은 저장소를 이용하여 사용할 수 있다. 따라서 프로세서(170)는 사용을 위한 마이크로 서비스가 이미 존재 한다면 해당 구조체들을 각 목적에 맞게 서로 이어 붙여 거대 시뮬레이터를 구성할 수 있다.

[0054] 목적에 맞게 사용하고자 하는 마이크로 서비스가 존재 하지 않는다면(Exist Micro Service? : N), 프로세서(170)는 개발 환경(Select OS Type) 및 시뮬레이터(Select Simulator)를 선택하는 입력을 수신하고, 새로운 마이크로 서비스를 구축할 수 있다.

[0055] 목적에 맞게 사용하고자 하는 마이크로 서비스가 존재하는 경우(Exist Micro Service? : Y), 해당 마이크로 서비스를 하나의 시스템으로 구성하고 존재 하지 않는다면 마이크로 서비스를 구성하여 저장소에 저장한 후 해당 서비스를 이용하여 시스템을 구축한다.

[0056] 도 6은 본 명세서의 일 실시예에 따라 우선 저장소(클라우드, 로컬)에 사용하고자 하는 마이크로 서비스가 존재하는 경우, 사용자 단말, 데이터 베이스, 컴퓨팅 시스템, 노드 사이의 시퀀스 다이어 그램을 나타낸다.

[0057] 도 7은 본 명세서의 일 실시예에 따라 다중 서버를 이용한 마이크로 서비스 아키텍처 시스템을 구축하는 예이다.

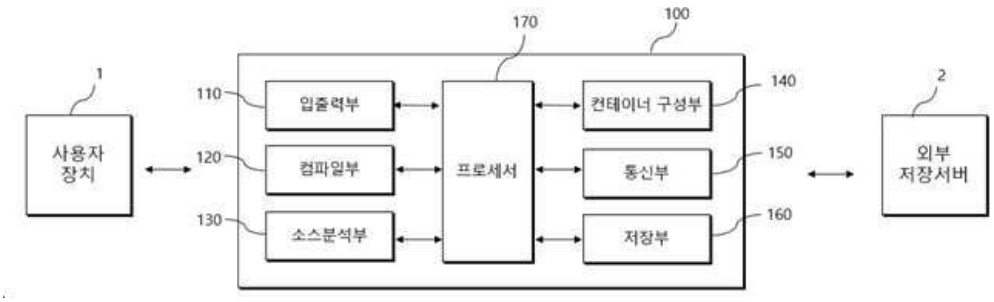
[0058] 도 7을 참조하면, 마이크로 서비스 아키텍처 구조를 기반으로 발명되어 각각의 마이크로 서비스 아키텍처 구현체가 그들 사이의 종속적인 관계보다 독립적인 관계를 하고 있어 단일 서버에 국한 되어야 할 이유가 없기 때문이다. 따라서 쿠버네티스와 같은 마이크로 서비스 아키텍처 구현체 오퍼레이션 툴을 활용하여 마이크로 서비스 아키텍처 구현체를 제한된 리소스에 맞게 배치하고 사용하며 하나의 물리적인 서버에 종속되지 않고 유연한 시스템 구축이 가능하다.

[0059] 도 8은 본 명세서의 일 실시예에 따라 다중 물리 서버를 이용하여 현실의 네트워크만을 이용하여 마이크로 서비스 아키텍처 시스템을 구축하는 예이다. 이는 가상환경을 사용하지 않고 현실세계의 자원과 네트워크만으로 시스템을 구축한다. 제안한 발명은 마이크로 서비스 아키텍처 구조를 기반으로 하여 단일 마이크로 서비스 구현체와 다중 구현체로 이루어진 시뮬레이터 간의 연동이 가능해 유연한 시스템 구축이 가능하다.

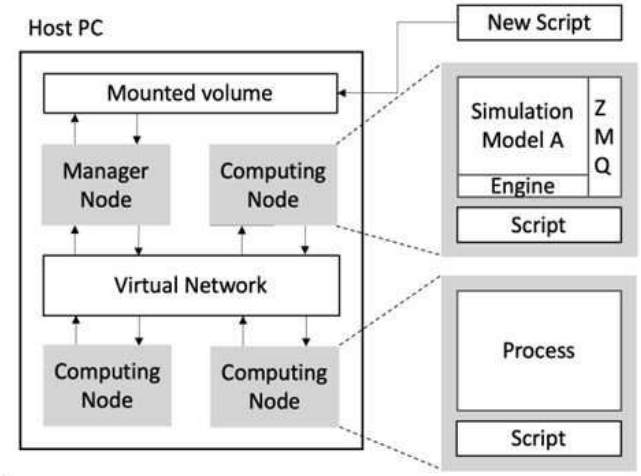
[0060] 전술한 본 발명은, 프로그램이 기록된 매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 매체는, 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 컴퓨터가 읽을 수 있는 매체의 예로는, HDD(Hard Disk Drive), SSD(Solid State Disk), SDD(Silicon Disk Drive), ROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크, 광 데이터 저장 장치 등이 있으며, 또한 캐리어 웨이브(예를 들어, 인터넷을 통한 전송)의 형태로 구현되는 것도 포함한다. 따라서, 상기의 상세한 설명은 모든 면에서 제한적으로 해석되어서는 아니되고 예시적인 것으로 고려되어야 한다. 본 발명의 범위는 첨부된 청구항의 합리적 해석에 의해 결정되어야 하고, 본 발명의 등가적 범위 내에서의 모든 변경은 본 발명의 범위에 포함된다.

도면

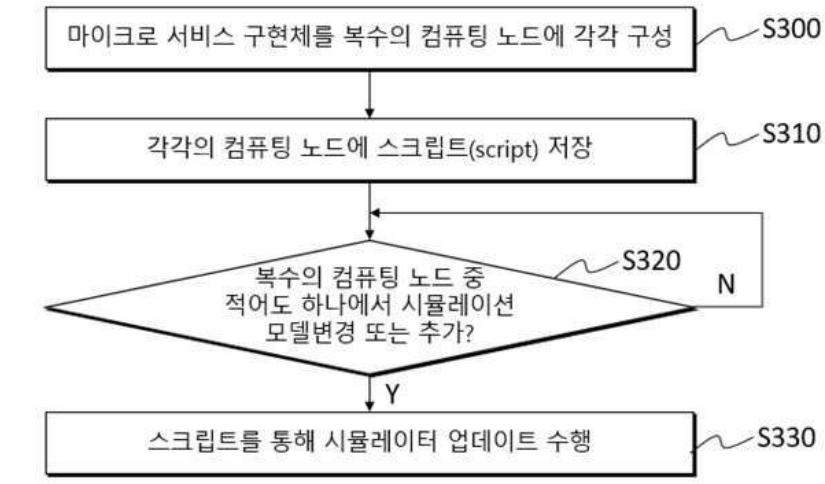
도면1



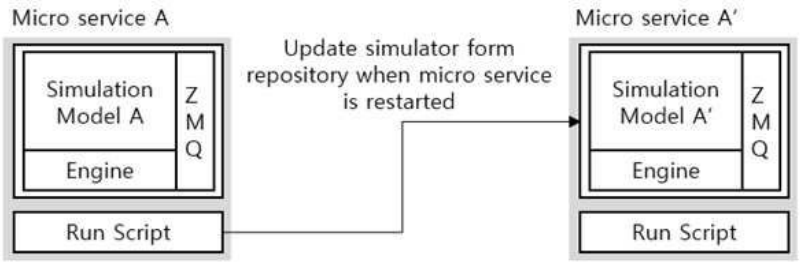
도면2



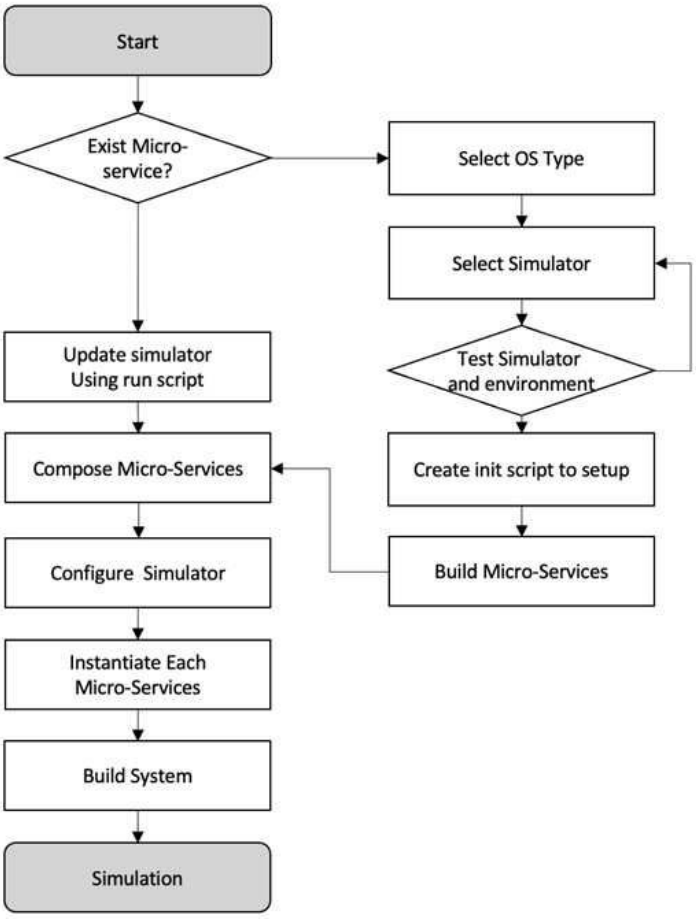
도면3



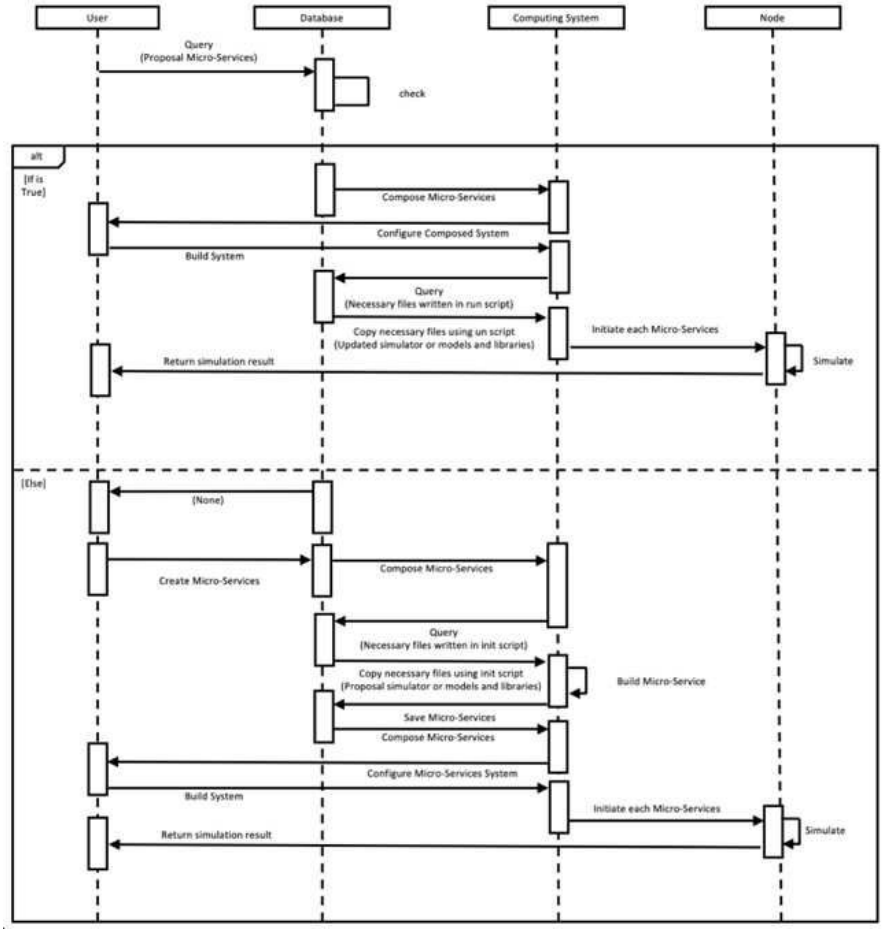
도면4


Micro service A
Simulation Model A
Engine
Z M Q
Run Script
Update simulator form repository when micro service is restarted
Micro service A'
Simulation Model A'
Engine
Z M Q
Run Script

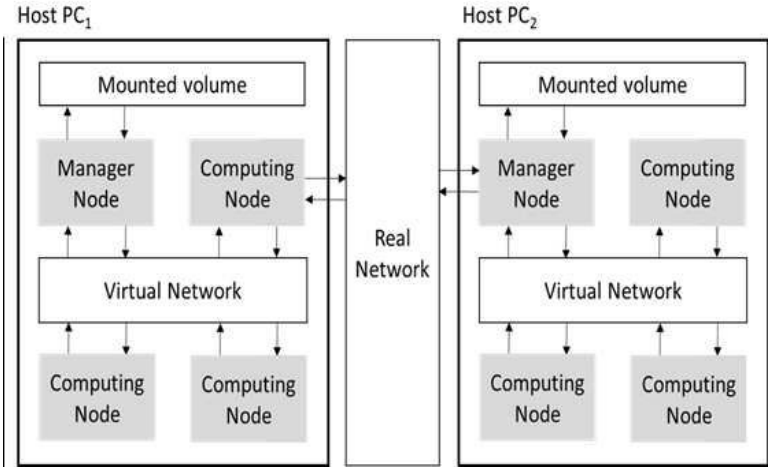
도면5


Start
Exist Micro-service?
Select OS Type
Select Simulator
Test Simulator and environment
Update simulator Using run script
Compose Micro-Services
Create init script to setup
Configure Simulator
Build Micro-Services
Instantiate Each Micro-Services
Build System
Simulation

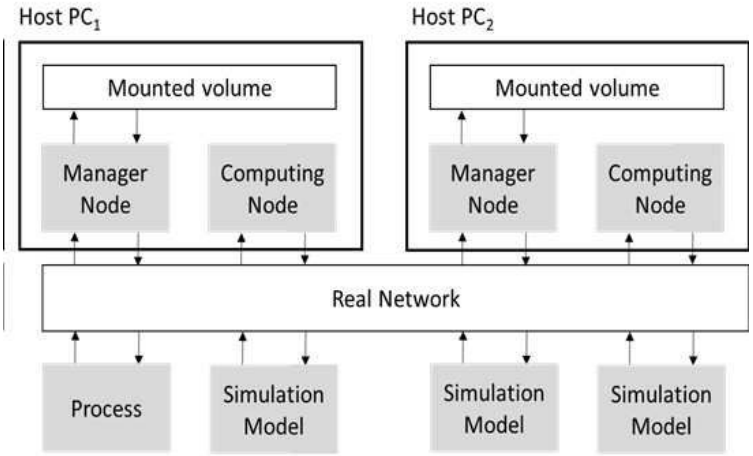
도면6


User
Database
Computing System
Node
Query
(Proposal Micro-Services)
check
alt
[If is True]
Compose Micro-Services
Configure Composed System
Build System
Query
(Necessary files written in run script)
Copy necessary files using un script
(Updated simulator or models and libraries)
Initiate each Micro-Services
Simulate
Return simulation result
[Else]
(None)
Create Micro-Services
Compose Micro-Services
Query
(Necessary files written in init script)
Copy necessary files using init script
(Proposal simulator or models and libraries)
Build Micro-Service
Save Micro-Services
Compose Micro-Services
Configure Micro-Services System
Build System
Initiate each Micro-Services
Simulate
Return simulation result

도면7


Host PC1
Host PC2
Mounted volume
Manager Node
Computing Node
Virtual Network
Computing Node
Computing Node
Real Network
Mounted volume
Manager Node
Computing Node
Virtual Network
Computing Node
Computing Node

도면8


Host PC1
Host PC2
Mounted volume
Mounted volume
Manager Node
Computing Node
Manager Node
Computing Node
Real Network
Process
Simulation Model
Simulation Model
Simulation Model