(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년10월15일
(11) 등록번호 10-2032521
(24) 등록일자 2019년10월08일

(51) 국제특허분류(Int. Cl.)
G06F 9/455 (2018.01) *G06F 9/50* (2018.01)
(52) CPC특허분류
G06F 9/45558 (2013.01)
G06F 9/5027 (2013.01)
(21) 출원번호 10-2018-0169620
(22) 출원일자 2018년12월26일
심사청구일자 2018년12월26일
(56) 선행기술조사문헌
US20070097132 A1*
(뒷면에 계속)

(73) 특허권자
래블업(주)
서울특별시 강남구 테헤란로 145 ,1684(역삼동)
(72) 발명자
김준기
서울특별시 강남구 테헤란로 145, 16층(역삼동)
신정규
서울특별시 강남구 테헤란로 145, 16층(역삼동)
박종현
서울특별시 강남구 테헤란로 145, 16층(역삼동)
(74) 대리인
특허법인세원

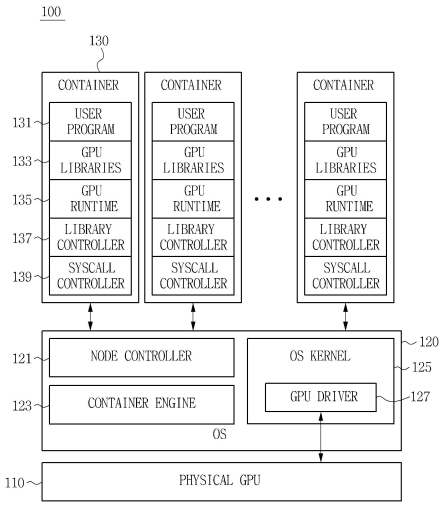
전체 청구항 수 : 총 5 항 심사관 : 유진태

(54) 발명의 명칭 **컨테이너 기반의 GPU 가상화 방법 및 시스템**

(57) 요 약

컨테이너 기반의 GPU 가상화 방법은, 컨테이너가 생성되면 노드 컨트롤러가 GPU 자원 제약 정보를 포함하는 설정 파일과 API 프로파일을 상기 컨테이너에 전송하는 단계와, 상기 컨테이너가 실행되면 상기 컨테이너에 구비되는 라이브러리 컨트롤러가 라이브러리 호출을 가로채어 GPU 자원량과 관련된 인자를 변경하고, 시스템콜 컨트롤러가 시스템콜 호출을 가로채어 인자 및 반환값을 변경하여 가상 GPU를 구현하는 단계를 포함한다.

대 표 도 - 도1



(52) CPC특허분류
G06F 2009/45562 (2013.01)
G06F 2009/4557 (2019.08)

(56) 선행기술조사문헌
KR1020180045347 A*
KR1020090026579 A*
KR101716715 B1
KR1020060079088 A
*는 심사관에 의하여 인용된 문헌

이 발명을 지원한 국가연구개발사업
과제고유번호 2018-0-01367
부처명 과학기술정보통신부
연구관리전문기관 정보통신기획평가원
연구사업명 SW컴퓨팅산업원천기술사업
연구과제명 GPU 자원의 사용률 향상을 위한 동적 GPU 클라우드 시스템 핵심 기술 개발
기 여 율 1/1
주관기관 래블업 주식회사
연구기간 2018.05.01 ~ 2019.12.31

명 세 서

청구범위

청구항 1

컨테이너 기반의 GPU 가상화 방법에 있어서,

컨테이너가 생성되면 노드 컨트롤러가 GPU 자원 제약 정보를 포함하는 설정파일과 API 프로파일을 상기 컨테이너에 전송하는 단계; 및

상기 컨테이너가 실행되면 상기 컨테이너에 구비되는 라이브러리 컨트롤러가 라이브러리 호출을 가로채어 GPU 자원량과 관련된 인자를 변경하고, 시스템콜 컨트롤러가 시스템콜 호출을 가로채어 인자 및 반환값을 변경하여 가상 GPU를 구현하는 단계를 포함하고,

상기 노드 컨트롤러는,

GPU 자원의 가용량을 확인하고, 노드 컨트롤러의 자원 정보를 초기화하고, 관리부에게 자원의 가용량을 보고하며, 상기 관리부로부터 할당받은 작업을 수신하되, 상기 노드 컨트롤러의 자원 가용량 정보를 요청한 양만큼 차감하여 갱신하는 컨테이너 기반의 GPU 가상화 방법.

청구항 2

삭제

청구항 3

제1항에 있어서,

상기 노드 컨트롤러는,

컨테이너가 생성되면 자원 제약 정보를 포함하는 설정파일을 컨테이너에 저장하고, 컨테이너가 실행 후 종료가 감지되면, 노드 컨트롤러의 자원 가용량 정보를 요청한 양 만큼 회수하여 갱신하는 컨테이너 기반의 GPU 가상화 방법.

청구항 4

제1항에 있어서,

상기 라이브러리 컨트롤러는,

사용자프로그램의 라이브러리 함수 호출 이벤트를 수신하면, GPU 자원 조회 및 할당 관련 API호출인지 판단하고, GPU 자원량과 관련된 인자, 구조체 필드, 반환값 중 적어도 하나를 변경하여 원래 라이브러리 함수를 호출하는 컨테이너 기반의 GPU 가상화 방법.

청구항 5

제1항에 있어서,

상기 시스템콜 컨트롤러는,

사용자프로그램의 시스템콜 호출 이벤트를 수신하면, 사전 정의된 API프로파일에 따라 허용, 차단, 변경 중 적어도 하나의 시스템콜 호출인지를 판단하고, API프로파일 규칙에 따라 허용, 차단, 또는 원래 시스템콜의 호출 전후 인자 및 반환값을 변경하는 컨테이너 기반의 GPU 가상화 방법.

청구항 6

컨테이너 기반의 GPU 가상화 시스템에 있어서,

자원 제약 정보를 포함하는 설정파일과 시스템콜/API프로파일을 컨테이너(130)에 전달하여 컨테이너 내에 전달

하는 노드컨트롤러를 포함하는 운영체제; 및

사용자프로그램의 라이브러리 함수 호출 이벤트를 수신하면, GPU 자원 조회 및 할당 관련 API호출인지 판단하고, GPU 자원량과 관련된 인자, 구조체 필드, 반환값 중 적어도 하나를 변경하여 원래 라이브러리 함수를 호출하는 라이브러리 컨트롤러와, 사용자프로그램의 시스템콜 호출 이벤트를 수신하면, 사전 정의된 API프로파일에 따라 허용, 차단, 변경 중 적어도 하나의 시스템콜 호출인지를 판단하고, API프로파일 규칙에 따라 허용, 차단, 또는 원래 시스템콜의 호출 전후 인자 및 반환값을 변경하는 시스템콜 컨트롤러를 포함하는 컨테이너로 구성되고,

상기 노드 컨트롤러는,

GPU 자원의 가용량을 확인하고, 노드 컨트롤러의 자원 정보를 초기화하고, 관리부에게 자원의 가용량을 보고하며, 상기 관리부로부터 할당받은 작업을 수신하되, 상기 노드 컨트롤러의 자원 가용량 정보를 요청한 양만큼 차감하여 갱신하는 컨테이너 기반의 GPU 가상화 시스템.

발명의 설명

기 술 분 야

[0001] 본 발명은 컨테이너 기반의 GPU 가상화 방법 및 시스템에 관한 것으로, 특히 컨테이너 내의 라이브러리 컨트롤러 및 시스템콜 컨트롤러에 의해 GPU 자원과 관련된 인자값 등을 변경하여 GPU 가상화를 구현하는 컨테이너 기반의 GPU 가상화 방법 및 시스템에 관한 것이다.

배 경 기 술

[0002] 최근 다중 사용자를 위한 대규모 컴퓨팅의 효율성, 보안성 및 호환성 향상을 위해 가상화기술이 많이 사용되고 있다. 대표적으로 가상머신(Virtual Machine)이 있으며, 애플리케이션, 서버, 스토리지 및 네트워크 등 다양한 분야에 적용되고 있다. 그러나 가상머신은 CPU부터 디스크, 네트워크, I/O까지 물리적 하드웨어 요소를 모두 가상화하기 때문에 호환성과 격리 수준은 가장 높지만 컴퓨팅 자원의 추가 소모량(오버헤드)이 크다는 단점을 가진다.

[0003] 한편, 컨테이너(Container)는 가상화가 아닌 운영체제 수준 격리 기술을 사용해 가상머신의 단점을 극복하는 가상화 기술로 부상하고 있다. 컨테이너는 커널 레벨 실행 환경은 호스트의 운영체제 커널을 공유하되 사용자 레벨 실행 환경은 완전히 격리된 파일시스템과 커널이 제공하는 자원 요소들의 가상화된 이름 공간을 사용하는 방식으로 구현된다. 격리된 파일시스템의 내용은 애플리케이션과 이를 구동하는데 필요한 모든 종속물, 라이브러리, 기타 바이너리와 구성 파일 등을 하나의 패키지로 묶어 구성된다. 가상화된 이름 공간으로 구분되어 컨테이너에게 제공되는 커널의 자원 요소에는 프로세스 ID, 네트워크 소켓, 사용자 계정, 프로세스 간 통신(IPC)를 위한 공유 메모리 등이 있다. 그 외의 하드웨어 접근은 컨테이너가 아닌 경우와 동일하게 처리되므로 호스트 하드웨어의 성능을 오버헤드 없이 온전히 활용할 수 있다. 여기에 운영체제는 컨테이너별로 최대 사용 가능한 하드웨어 자원의 양을 제한할 수 있는 옵션을 제공한다.

[0004] 최근 딥러닝 기술의 발달과 대규모 연산 수요의 증가에 따라 컴퓨팅 자원을 최적으로 공유하고 관리하는 기술이 요청되고 있다. 성능 향상을 위해, 딥러닝의 연산 특성에 최적화된 연산 가속 하드웨어들이 등장하고 있으며, GPU 또한 그 중 하나이다. 하지만 기존 운영체제에서 제공하는 컨테이너 기반 가상화 기술은 컨테이너별로 CPU, 메모리, 디스크 및 파일시스템에 대한 자원 공유 및 제한만 지원하고 있으며, GPU와 같은 연산 가속 하드웨어에 대해서는 여러 컨테이너가 동시에 공유할 수 있는 기술이 제공되지 않고 있다. 이에 따라 GPU를 효율적으로 공유 및 관리하는 데에 어려움이 발생하고 있다.

선행기술문헌

특허문헌

[0005] (특허문헌 0001) 한국등록특허 제10-1848450호

발명의 내용

해결하려는 과제

[0006] 본 발명이 해결하고자 하는 과제는 컨테이너를 이용하여 물리적 가상화가 아닌 운영체제 수준의 가상화를 통해 GPU 자원을 동적으로 할당 및 공유가 가능한 컨테이너 기반의 GPU 가상화 방법 및 시스템을 제공하는 데 있다.

과제의 해결 수단

[0007] 본 발명의 일 실시예에 따른 컨테이너 기반의 GPU 가상화 방법은, 컨테이너가 생성되면 노드 컨트롤러가 GPU 자원 제약 정보를 포함하는 설정파일과 API 프로파일을 상기 컨테이너에 전송하는 단계와, 상기 컨테이너가 실행되면 상기 컨테이너에 구비되는 라이브러리 컨트롤러가 라이브러리 호출을 가로채어 GPU 자원량과 관련된 인자를 변경하고, 시스템콜 컨트롤러가 시스템콜 호출을 가로채어 인자 및 반환값을 변경하여 가상 GPU를 구현하는 단계를 포함한다.

[0008] 본 발명의 일 실시예에 따른 컨테이너 기반의 GPU 가상화 시스템은 자원 제약 정보를 포함하는 설정파일과 시스템콜/API프로파일을 컨테이너에 전달하여 컨테이너 내에 전달하는 노드컨트롤러를 포함하는 운영체제와, 사용자 프로그램의 라이브러리 함수 호출 이벤트를 수신하면, GPU 자원 조회 및 할당 관련 API호출인지 판단하고, GPU 자원량과 관련된 인자, 구조체 필드, 반환값 중 적어도 하나를 변경하여 원래 라이브러리 함수를 호출하는 라이브러리 컨트롤러와, 사용자프로그램의 시스템콜 호출 이벤트를 수신하면, 사전 정의된 API프로파일에 따라 허용, 차단, 변경 중 적어도 하나의 시스템콜 호출인지를 판단하고, API프로파일 규칙에 따라 허용, 차단, 또는 원래 시스템콜의 호출 전후 인자 및 반환값을 변경하는 시스템콜 컨트롤러를 포함하는 컨테이너로 구성된다.

발명의 효과

[0009] 본 발명에 의하면, 컨테이너 가상화 기술을 확장함으로써 단일 컨테이너에 단일 GPU를 할당하거나 또는 단일 컨테이너에 다중 GPU를 할당하거나 또는 단일 GPU를 다중 컨테이너가 공유하거나 또는 다중 GPU를 다중 컨테이너가 공유하는 GPU 컴퓨팅 시스템 구현이 가능하다.

[0010] 또한, 컨테이너를 이용하여 구현함으로써 가상머신과 비교하여 시스템 자원을 더 효율적으로 이용할 수 있고, 애플리케이션 이동이 가능하며 스케일링이 간단하여 업데이트가 용이한 효과가 있다.

도면의 간단한 설명

[0011] 도 1은 본 발명의 일실시예에 따른 컨테이너 기반의 GPU 가상화 시스템의 소프트웨어 구조를 설명하기 위한 도면이다.

도 2는 본 발명의 일실시예에 따른 컨테이너 기반의 GPU 가상화 방법을 설명하는 순서도이다.

도 3은 본 발명의 일실시예에 따른 노드 컨트롤러의 동작 방법을 설명하는 순서도이다.

도 4는 본 발명의 일실시예에 따른 라이브러리 컨트롤러의 동작 방법을 설명하는 순서도이다.

도 5는 본 발명의 일실시예에 따른 시스템콜 컨트롤러의 동작방법을 설명하는 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0012] 본 명세서에 개시되어 있는 본 발명의 개념에 따른 실시 예들에 대해서 특정한 구조적 또는 기능적 설명은 단지 본 발명의 개념에 따른 실시 예들을 설명하기 위한 목적으로 예시된 것으로서, 본 발명의 개념에 따른 실시 예들은 다양한 형태들로 실시될 수 있으며 본 명세서에 설명된 실시 예들에 한정되지 않는다.

[0013] 본 발명의 개념에 따른 실시 예들은 다양한 변경들을 가할 수 있고 여러 가지 형태들을 가질 수 있으므로 실시 예들을 도면에 예시하고 본 명세서에서 상세하게 설명하고자 한다. 그러나 이는 본 발명의 개념에 따른 실시 예들을 특정한 개시 형태들에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물, 또는 대체물을 포함한다.

[0014] 본 명세서에서 사용한 용어는 단지 특정한 실시 예를 설명하기 위해 사용된 것으로서, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 본 명세서에 기재된 특징, 숫자, 단계, 동작, 구성 요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성 요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야

한다.

[0015] 이하, 본 명세서에 첨부된 도면들을 참조하여 본 발명의 실시 예들을 상세히 설명한다.

[0016] 도 1은 본 발명의 일실시예에 따른 컨테이너 기반의 GPU 가상화 시스템의 소프트웨어 구조를 설명하기 위한 도면이다.

[0017] 도 1을 참조하면, GPU 가상화 시스템(100)의 소프트웨어 구조는 물리적 GPU(PHYSICAL GPU; 110), 운영체제(120), 다수개의 컨테이너(130)로 구성된다.

[0018] 운영체제(120)는 노드컨트롤러(121), 컨테이너엔진(123), 운영체제 커널(125)로 구성된다. 운영체제(120)는 운영체제 커널(125) 내에 설치된 GPU드라이버(127)를 통해 물리적GPU(110)와 통신한다.

[0019] 노드컨트롤러(121)는 자원 제약 정보를 포함하는 설정파일과 시스템콜/API프로파일을 컨테이너(130)에 전달하여 컨테이너 내에 저장할 수 있다. 노드컨트롤러(121)는 GPU자원의 가용량을 확인하고, 노드 컨트롤러의 자원 정보를 초기화할 수 있다. 상기 GPU자원은 GPU Processing Units와 GPU Memory(메모리)일 수 있으나 이에 대해 한정하는 것은 아니다. 노드컨트롤러(121)는 관리부(Manager)에 확인된 자원의 가용량을 보고하며, 상기 관리부로부터 할당받은 작업을 수신할 수 있다. 노드 컨트롤러(121)는 GPU자원의 가용량 정보를 갱신할 수 있고, 이때, 요청한 양만큼 차감할 수 있다. 노드컨트롤러(121)는 컨테이너가 생성되면 자원 제약 정보를 포함하는 설정파일을 컨테이너에 전달하고, 컨테이너가 실행 후 종료가 감지되면, 노드 컨트롤러의 자원 가용량 정보를 요청한 양만큼 회수하여 갱신할 수 있다. 노드컨트롤러(121)는 사용자의 코드 실행 요청을 컨테이너 내에서 실행할 수 있다.

[0020] 컨테이너엔진(123)는 컨테이너(130)를 생성하고 배포하며, 각 컨테이너(130)가 해당되는 응용 프로그램을 실행할 수 있도록 GPU자원을 할당하는 역할을 수행한다. 컨테이너엔진(123)은 컨테이너를 실행하고 종료할 수 있다.

[0021] 컨테이너(130)는 사용자프로그램을 구동할 수 있도록 필요한 각종 프로그램, 소스코드 등과 라이브러리 등을 묶은 이미지를 포함하는 공간이다. 사용자프로그램의 구동은 운영체제(120)에서 실질적으로 이루어지게 된다. 즉, 운영체제(120)는 컨테이너엔진(123)을 통해 각각의 컨테이너(130)에 접근하여 해당하는 사용자 프로그램을 실행하고 처리할 수 있다.

[0022] 컨테이너(130)는 사용자프로그램(131), GPU라이브러리(133), GPU런타임(135), 라이브러리 컨트롤러(137), 시스템콜 컨트롤러(139)로 구성된다.

[0023] 사용자프로그램(131)은 노드컨트롤러의 사용자의 코드 실행 요청을 컨테이너 내에서 실행하도록 동작할 수 있다.

[0024] GPU라이브러리(133)는 딥러닝 프레임워크가 동작하도록 라이브러리를 포함할 수 있으며, 예컨대 텐서플로우(TensorFlow), Caffe, Pytorch, CNTK, Chainer 와 같은 딥러닝 프레임 워크 중 적어도 하나가 동작할 수 있다.

[0025] GPU런타임(135)은 GPU에서 수행하는 병렬처리 알고리즘인 CUDA, OpenCL, ROCM이 설치되어 사용될 수 있다. 상기 CUDA는 머신러닝 분야에서 활용되는 GPU 미들웨어로서 GPU런타임에서 동작할 수 있다. 상기 OpenCL은 머신러닝 분야와 고성능 컴퓨팅(HPC, High Performance Computing)에 활용하는 병렬 컴퓨팅과 크로스플랫폼으로 동작할 수 있다.

[0026] 라이브러리 컨트롤러(137)는 사용자프로그램의 라이브러리 함수 호출 이벤트를 수신하면, GPU 자원 조회 및 할당 관련 API호출인지 판단하고, GPU 자원량과 관련된 인자, 구조체 필드, 반환값 중 적어도 하나를 변경하여 원래 라이브러리 함수를 호출할 수 있다. GPU자원 조회 및 할당 관련 API호출이 아니면, 원래 라이브러리 함수를 인자 변경 없이 호출하고 반환값을 그대로 리턴할 수 있다.

[0027] 시스템콜 컨트롤러(139)는 사용자프로그램의 시스템콜 호출 이벤트를 수신하면, 사전 정의된 API프로파일에 따라 허용, 차단, 변경 중 적어도 하나의 시스템콜 호출인지를 판단하고, API프로파일 규칙에 따라 허용, 차단, 또는 원래 시스템콜의 호출 전후 인자 및 반환값을 변경할 수 있다. 사전 정의된 API프로파일에 따라 허용, 차단, 변경 중 적어도 하나의 시스템콜 호출이 아니면 원래 시스템콜을 인자 변경 없이 호출하고 반환값을 그대로 리턴할 수 있다.

[0028] 즉, 컨테이너 내의 라이브러리 컨트롤러(137)가 라이브러리 호출을 가로채어 GPU 자원량과 관련된 인자를 변경하고, 시스템콜 컨트롤러(139)가 시스템콜 호출을 가로채어 인자 및 반환값을 변경함으로써 가상 GPU를 구현할 수 있다.

[0029] 도 2는 본 발명의 일실시예에 따른 컨테이너 가상화 방법을 설명하는 순서도이다.

[0030] 도 2를 참조하면, 컨테이너가 생성되면(S201), 노드 컨트롤러(121)가 GPU 자원 제약 정보를 포함하는 설정파일과 시스템콜/API 프로파일을 상기 컨테이너에 전송한다(S203). 컨테이너 내의 라이브러리 컨트롤러와 시스템콜 컨트롤러가 자원 제약 정보를 포함하는 설정파일을 수신하여 저장할 수 있다.

[0031] 컨테이너가 실행되면 상기 컨테이너에 구비되는 라이브러리 컨트롤러(137)가 라이브러리 호출을 가로채어 GPU 자원량과 관련된 인자를 변경하고, 시스템콜 컨트롤러(139)가 시스템콜 호출을 가로채어 인자 및 반환값을 변경하여 가상 GPU를 구현한다(S205). 이때, 라이브러리 컨트롤러(137)는 GPU 자원량과 관련된 인자 뿐 아니라 구조체 필드, 반환값을 변경하여 원래 라이브러리 함수를 호출할 수 있다.

[0032] 도 3은 본 발명의 일실시예에 따른 노드 컨트롤러의 동작 방법을 설명하는 순서도이다.

[0033] 도 3을 참조하면, 노드 컨트롤러는 먼저 GPU의 자원 가용량을 확인한다(S301). 그리고 노드 컨트롤러는 자원 정보를 초기화한다(S303).

[0034] 서버 실행 루프에 의해 이하 프로세스가 반복 진행될 수 있다(S305). 확인된 GPU 자원 가용량을 관리부(Manager)에 자원 가용량을 보고하고(S307), 상기 관리부로부터 할당받은 작업(job spec)을 수신한다(S309). 노드컨트롤러(121)는 자원 가용량 정보를 갱신한다(S311). 이때, 요청한 양만큼 차감될 수 있다. 이후에, 컨테이너가 생성되고(S313), 라이브러리 컨트롤러와 시스템콜 컨트롤러가 읽을 자원 제약 정보를 포함하는 설정파일을 컨테이너에 전송하여 저장한다(S315). 이후 컨테이너가 실행되고(S317), 컨테이너 종료가 감지되면 노트 컨트롤러의 자원 가용량 정보를 갱신한다(S319). 이때, 요청한 양을 회수할 수 있다.

[0035] 도 4는 본 발명의 일실시예에 따른 라이브러리 컨트롤러의 동작 방법을 설명하는 순서도이다.

[0036] 도 4를 참조하면, 사용자 프로그램의 라이브러리 함수 호출 이벤트를 수신한다(S401). 이후에, GPU 자원 조회 및 할당 관련 API 콜인지 판단한다(S403).

[0037] 판단 결과에 따라 GPU 자원 조회 및 할당 관련 API콜인 경우에는, GPU 자원량과 관련된 인자, 구조체 필드, 반환값 중 적어도 하나를 변경한다(S405). 이때, 내장된 API 프로파일 및 컨테이너 설정파일에 기초하여 변경될 수 있다.

[0038] 이후에 변경 후 원래 라이브러리 함수를 호출한다(S407).

[0039] 판단 결과에 따라 GPU 자원 조회 및 할당 관련 API콜인 경우가 아니면 원래 라이브러리 함수를 인자 변경 없이 호출하고 반환값을 그대로 리턴한다(S409).

[0040] 도 5는 본 발명의 일실시예에 따른 시스템콜 컨트롤러의 동작방법을 설명하는 순서도이다.

[0041] 도 5를 참조하면, 사용자 프로그램의 시스템콜 호출 이벤트를 수신한다(S501). 사전 정의된 API프로파일에 변경이 필요한 시스템콜인지 판단한다(S503). 이때, 변경 뿐 아니라 허용, 차단해야 하는 경우인지도 판단할 수 있다. 판단 결과에 따라 허용, 차단 및 변경이 필요한 시스템콜이면, API프로파일 규칙에 따라 허용, 차단 또는 원래 시스템콜의 호출 전후 인자 및 반환값을 변경한다(S405).

[0042] 판단 결과에 따라 허용, 차단 및 변경이 필요하지 않은 시스템콜이면, 원래 라이브러리 함수를 인자 변경 없이 호출하고 반환값을 그대로 리턴한다(S407).

[0043] 본 발명은 도면에 도시된 실시 예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시 예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 등록청구범위의 기술적 사상에 의해 정해져야 할 것이다.

부호의 설명

[0044] 100; GPU 가상화 시스템 110; 물리적GPU(PHYSICAL GPU)

120; 운영체제(OS) 121; 노드 컨트롤러

123; 컨테이너 엔진 125; 운영체제 커널

127; GPU 드라이버 130; 컨테이너(Container)

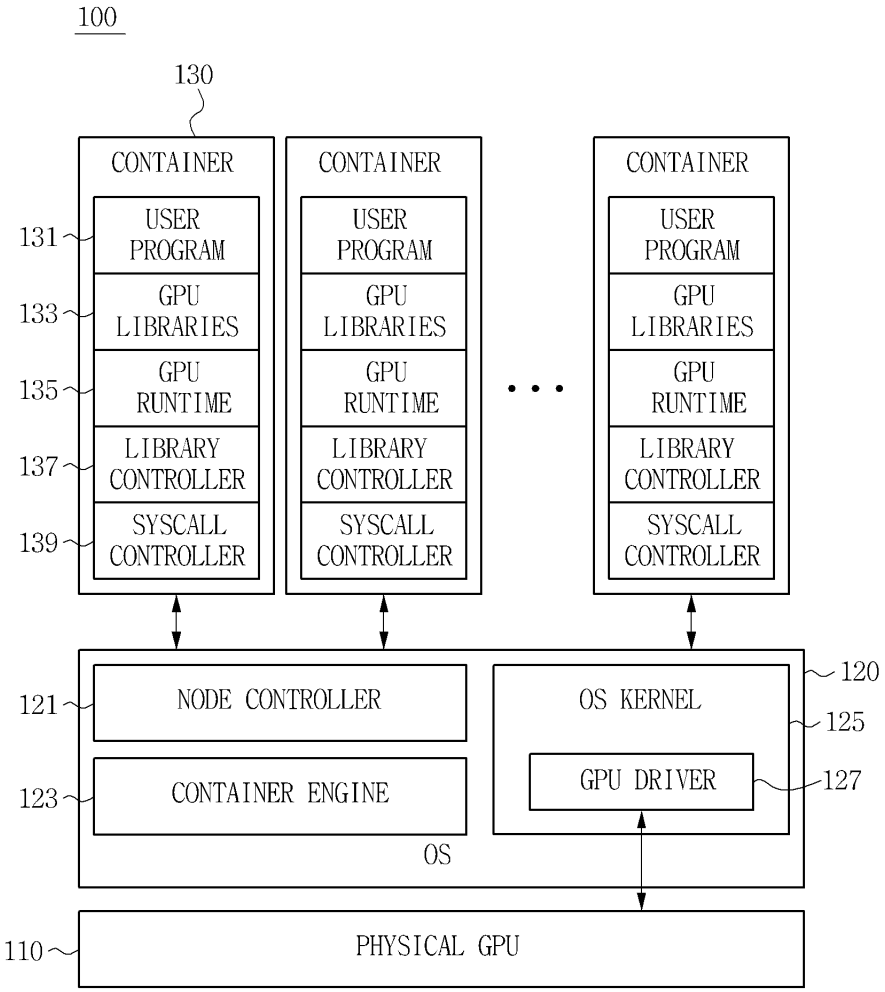
131; 사용자프로그램 133; GPU 라이브러리

135; GPU 런타임　137; 라이브러리 컨트롤러

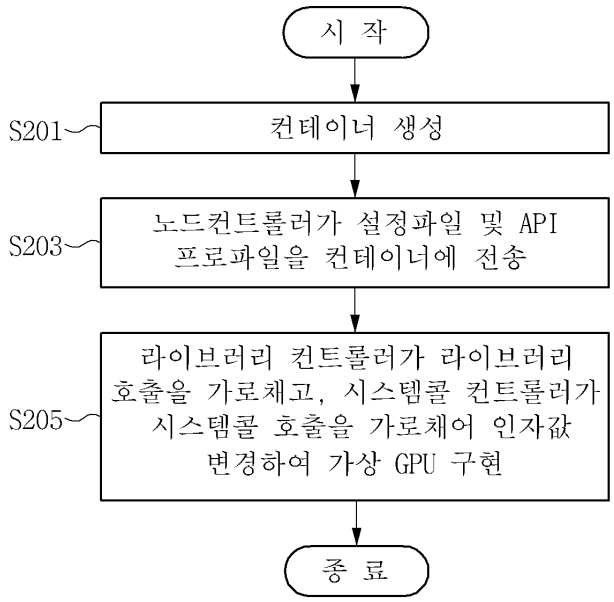
139; 시스템콜 컨트롤러

도면

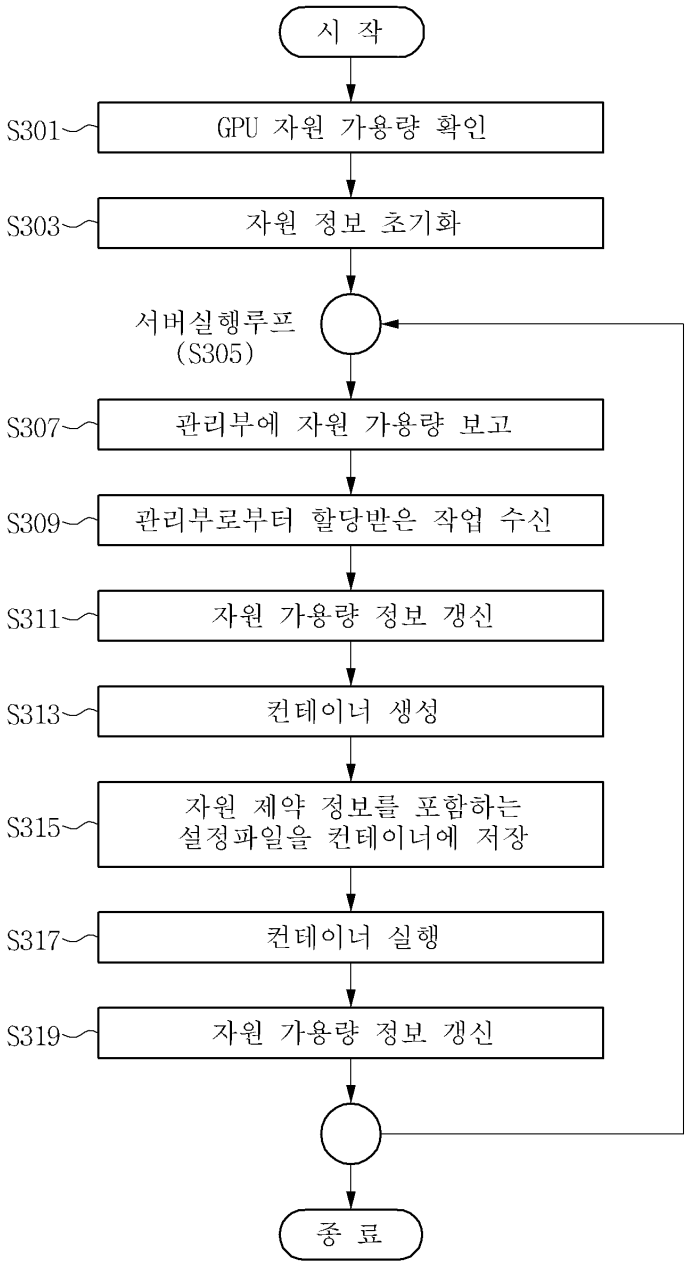
도면1



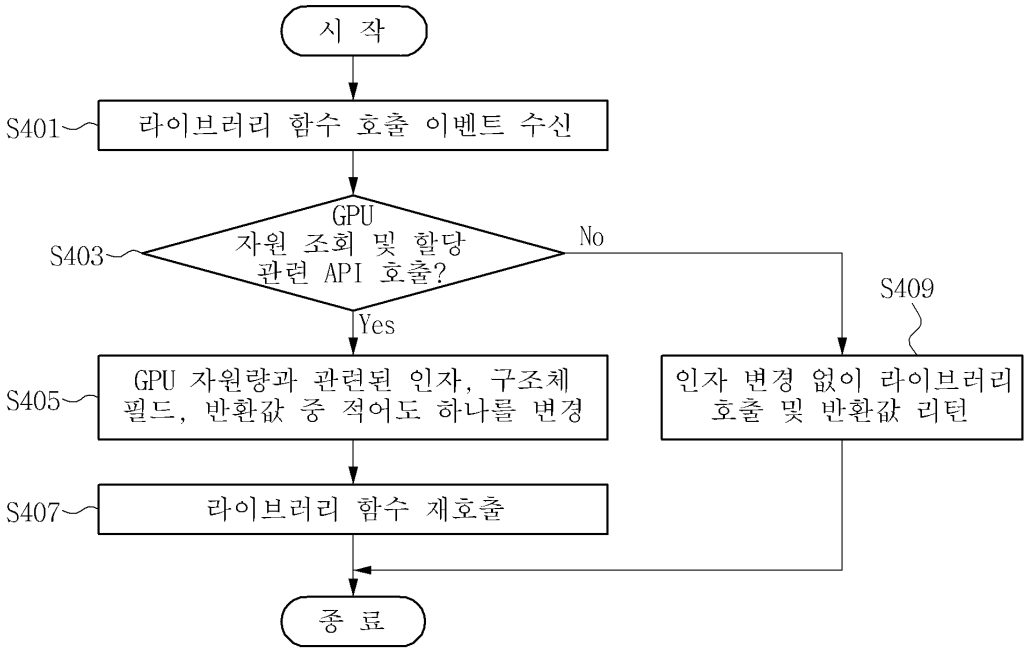
도면2


시 작
S201
컨테이너 생성
S203
노드컨트롤러가 설정파일 및 API 프로파일을 컨테이너에 전송
S205
라이브러리 컨트롤러가 라이브러리 호출을 가로채고, 시스템콜 컨트롤러가 시스템콜 호출을 가로채어 인자값 변경하여 가상 GPU 구현
종 료

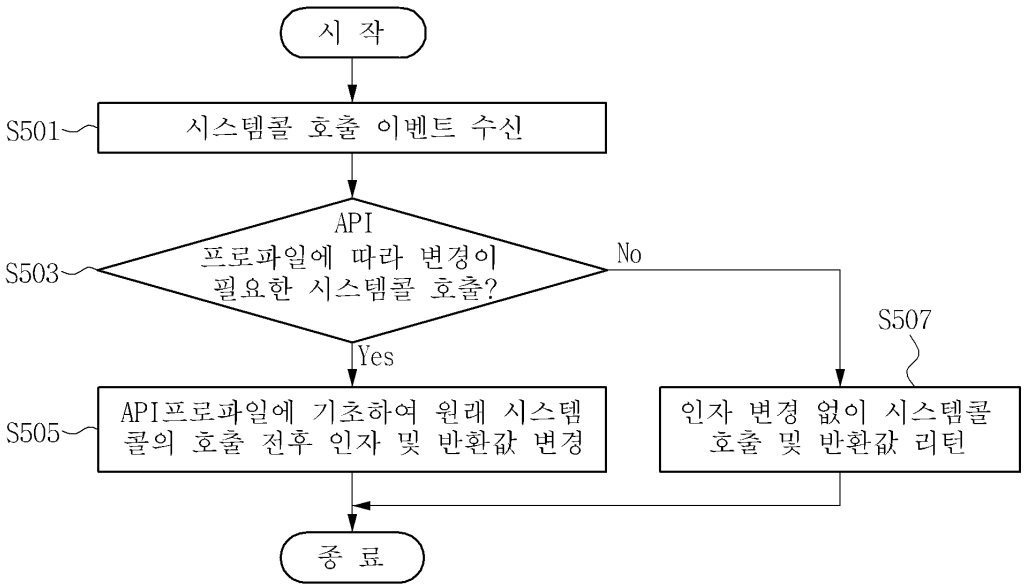
도면3


시 작
S301
GPU 자원 가용량 확인
S303
자원 정보 초기화
서버실행루프
(S305)
S307
관리부에 자원 가용량 보고
S309
관리부로부터 할당받은 작업 수신
S311
자원 가용량 정보 갱신
S313
컨테이너 생성
S315
자원 제약 정보를 포함하는
설정파일을 컨테이너에 저장
S317
컨테이너 실행
S319
자원 가용량 정보 갱신
종 료

도면4


시 작
S401
라이브러리 함수 호출 이벤트 수신
S403
GPU 자원 조회 및 할당 관련 API 호출?
No
Yes
S409
인자 변경 없이 라이브러리 호출 및 반환값 리턴
S405
GPU 자원량과 관련된 인자, 구조체 필드, 반환값 중 적어도 하나를 변경
S407
라이브러리 함수 재호출
종 료

도면5


시 작
S501
시스템콜 호출 이벤트 수신
S503
API 프로파일에 따라 변경이 필요한 시스템콜 호출?
No
Yes
S507
인자 변경 없이 시스템콜 호출 및 반환값 리턴
S505
API프로파일에 기초하여 원래 시스템콜의 호출 전후 인자 및 반환값 변경
종 료