(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년10월01일
(11) 등록번호 10-2307900
(24) 등록일자 2021년09월27일

(51) 국제특허분류(Int. Cl.)
G06F 9/455 (2018.01) *G06F 11/30* (2006.01)
H04L 29/08 (2006.01)
(52) CPC특허분류
G06F 9/45558 (2013.01)
G06F 11/301 (2013.01)
(21) 출원번호 10-2020-0019490
(22) 출원일자 2020년02월18일
심사청구일자 2020년02월18일
(65) 공개번호 10-2021-0105036
(43) 공개일자 2021년08월26일
(56) 선행기술조사문헌
KR1020090081749 A
(뒷면에 계속)

(73) 특허권자
세종대학교산학협력단
서울특별시 광진구 능동로 209 (군자동, 세종대학교)
(72) 발명자
박기웅
서울특별시 광진구 능동로17길 21, 304호(화양동)
최상훈
서울특별시 광진구 긴고랑로2길 38-2, 301호(중곡동)
이병용
서울특별시 광진구 광나루로13길 32, 201호(군자동, 군자위드빌)
(74) 대리인
양성보

전체 청구항 수 : 총 11 항 심사관 : 유진태

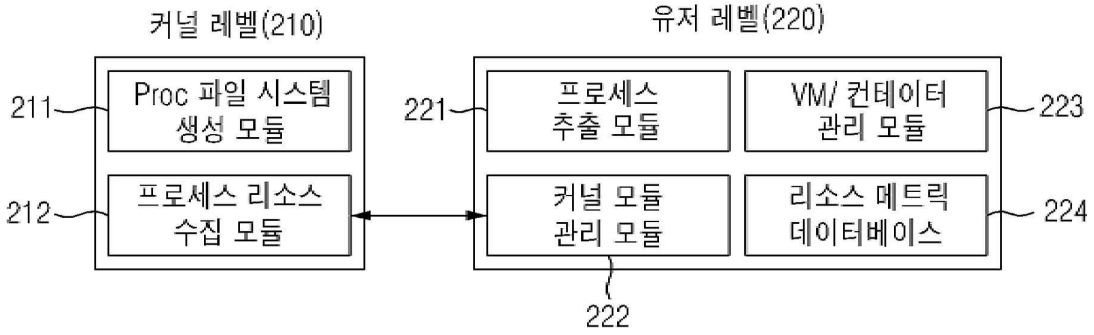
(54) 발명의 명칭 **클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 방법 및 시스템**

(57) 요 약

클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 방법 및 시스템이 제시된다. 본 발명에서 제안하는 클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 방법은 클라우드 플랫폼에서 구동 중인 VM 또는 컨테이너를 가상화 관리 모듈을 통해 조회하는 단계, 가상화 관리 모듈을 통해 조회된 정보에 기초하여 VM 또는 컨테

(뒷면에 계속)

대 표 도 - 도2



이너에 해당하는 프로세스 ID 및 리소스 메트릭 데이터베이스에 저장될 리소스 메트릭의 이름을 추출하는 단계, 추출된 프로세스 ID 및 리소스 메트릭 이름을 매개변수로 전달하여 리눅스 커널에 리소스 메트릭 수집을 위한 커널 모듈을 적재하는 단계, 모니터링 커널 모듈이 적재된 상태에서 VM/컨테이너 스케줄러에 의해 구동 중인 VM 또는 컨테이너의 정보를 업데이트 하고 해시 파일로 생성하는 단계, VM 또는 컨테이너의 해시 업데이트 여부를 판단하는 단계 및 VM 또는 컨테이너의 해시 업데이트가 있는 경우, 현재 적재된 커널 모듈을 제거하는 단계를 포함한다.

(52) CPC특허분류
H04L 67/16 (2013.01)
G06F 2009/45591 (2019.08)

(56) 선행기술조사문헌
KR1020160044623 A
KR1020200013028 A
김민석, 박기웅. '컨테이너 모니터링 툴 프로파일링을 통한 커버리지 영역 분석'. 한국차세대컴퓨팅학회 춘계학술대회, 2018. 05., pp.108-112.
Jimenez 외 5명. 'CoMA: Resource Monitoring of Docker Containers'. Proceedings of the 5th International Conference on Cloud Computing and Services Science, 2015, pp.145-154.

이 발명을 지원한 국가연구개발사업

과제고유번호	1711080983
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	SW컴퓨팅산업원천기술개발
연구과제명	API 호출 단위 자원 할당 및 사용량 계량이 가능한 서버리스 클라우드 컴퓨팅 기술 개발
기 여 율	1/1
과제수행기관명	전자부품연구원
연구기간	2018.04.01 ~ 2020.12.31

공지예외적용 : 있음

명 세 서

청구범위

청구항 1

클라우드 플랫폼에서 구동 중인 VM 또는 컨테이너를 VM/컨테이너 관리 모듈을 통해 조회하는 단계;

VM/컨테이너 관리 모듈을 통해 조회된 VM 또는 컨테이너에 기초하여 프로세스 추출 모듈이 현재 구동중인 VM 또는 컨테이너에 해당하는 프로세스 ID 및 리소스 메트릭 데이터베이스에 저장될 리소스 메트릭의 이름을 추출하고, 프로세스 리소스 수집 모듈이 리소스 메트릭의 이름을 수집하는 단계;

추출된 프로세스 ID 및 리소스 메트릭 이름을 매개변수로 전달하여 커널 모듈 관리 모듈을 통해 리눅스 커널에 리소스 메트릭 수집을 위한 커널 모듈을 적재하는 단계;

모니터링 커널 모듈이 적재된 상태에서 VM/컨테이너 관리 모듈이 VM/컨테이너 스케줄러에 의해 구동 중인 VM 또는 컨테이너의 정보(상기 프로세스 ID 및 상기 리소스 메트릭 이름을 포함함)를 업데이트 하고 상기 VM 또는 컨테이너에 관한 해시 파일로 생성하는 단계;

VM/컨테이너 관리 모듈을 통해 상기 VM 또는 컨테이너에 관한 해시 파일 업데이트 여부를 판단하는 단계; 및

상기 VM 또는 컨테이너에 관한 해시 파일 업데이트가 있는 경우, 커널 모듈 관리 모듈을 통해 현재 적재된 커널 모듈을 제거하는 단계

를 포함하는 리소스 메트릭 수집 방법.

청구항 2

제1항에 있어서,

상기 VM 또는 컨테이너에 관한 해시 파일 업데이트가 없는 경우, 모니터링 커널 모듈이 적재된 상태에서 VM/컨테이너 스케줄러에 의해 구동 중인 VM 또는 컨테이너의 정보를 업데이트 하고 상기 VM 또는 컨테이너에 관한 해시 파일로 생성하는 단계부터 반복하는

리소스 메트릭 수집 방법.

청구항 3

제1항에 있어서,

현재 적재된 커널 모듈을 제거한 후 측정 대상이 되는 새로운 VM 또는 컨테이너 정보를 반영하여 커널 모듈을 재적재하는 과정을 수행하기 위해, VM 또는 컨테이너에 해당하는 프로세스 ID 및 리소스 메트릭 데이터베이스에 저장될 리소스 메트릭의 이름을 추출하는 단계부터 반복하는

리소스 메트릭 수집 방법.

청구항 4

제1항에 있어서,

VM/컨테이너 관리 모듈을 통해 조회된 VM 또는 컨테이너들에 기초하여 프로세스 추출 모듈이 현재 구동중인 VM 또는 컨테이너에 해당하는 프로세스 ID 및 리소스 메트릭 데이터베이스에 저장될 리소스 메트릭의 이름을 추출하고, 프로세스 리소스 수집 모듈이 프로세스 ID 및 리소스 메트릭의 이름을 수집하는 단계는,

커널 레벨에서 구동되는 프로세스 리소스 수집 모듈이 커널 모듈 관리 모듈로부터 프로세스 ID를 입력 받고, 입력 받은 프로세스 ID의 테스크 구조체에 접근하여, 테스크에 대한 MM(Memory Management) 구조체의 주소를 가져와 프로세스 별 리소스 사용량에 접근하기 위해 MM 포인터 주소로부터 메모리 컨트롤러 데이터 구조체의 주소를 획득하는

리소스 메트릭 수집 방법.

청구항 5

제4항에 있어서,

프로세스 리소스 수집 모듈이 메모리 사용량 관련 리소스 메트릭 정보를 획득하기 위해 메모리 컨트롤러 데이터 구조체의 주소와 리소스 메트릭을 전달받고, 해당 프로세스 ID 리소스 사용량을 커널 모듈로 반환하여, 페이지 처리 관련 리소스 메트릭 정보를 획득하기 위해 해당 프로세스 ID의 페이지 처리 관련 값을 커널 모듈로 반환하는

리소스 메트릭 수집 방법.

청구항 6

복수의 리소스 정보를 하나의 Proc 파일 시스템에 대응시키는 Proc 파일 시스템 생성 모듈;

현재 클라우드 환경에서 구동 중인 VM 또는 컨테이너를 조회하는 VM/컨테이너 관리 모듈;

상기 하나의 Proc 파일 시스템에 대응되는 복수의 리소스 정보에 대하여 유저 레벨에서 리소스 메트릭의 수집 대상이 되는 해당 프로세스의 ID 및 리소스 메트릭의 이름을 전달받아, 해당 프로세스에 대한 리소스 메트릭을 수집하는 프로세스 리소스 수집 모듈;

현재 구동중인 VM/컨테이너의 프로세스 ID 및 리소스 메트릭 데이터베이스에 저장될 리소스 메트릭의 이름을 추출하는 프로세스 추출 모듈;

프로세스 추출 모듈에서 추출된 프로세스 ID 및 리소스 메트릭 이름을 매개변수로 하고, 현재 클라우드 환경에서 구동 중인 VM 또는 컨테이너에 대하여 커널 레벨에서 동작하는 커널 모듈을 동적으로 적재 및 제거 하는 커널 모듈 관리 모듈; 및

프로세스 추출 모듈에서 추출된 프로세스 ID 및 리소스 메트릭의 이름을 저장하는 리소스 메트릭 데이터베이스

를 포함하는 리소스 메트릭 수집 시스템.

청구항 7

제6항에 있어서,

프로세스 추출 모듈은,

VM/컨테이너 관리 모듈을 통해 조회된 VM 또는 컨테이너에 기초하여 상기 VM 또는 컨테이너에 해당하는 프로세스 ID 및 리소스 메트릭 데이터베이스에 저장될 리소스 메트릭의 이름을 추출하는

리소스 메트릭 수집 시스템.

청구항 8

제6항에 있어서,

커널 모듈 관리 모듈은,

프로세스 추출 모듈을 통해 추출된 프로세스 ID 및 리소스 메트릭 이름을 매개변수로 전달하여 리눅스 커널에 리소스 메트릭 수집을 위한 커널 모듈을 적재하도록 하고, 상기 VM 또는 컨테이너에 관한 해시 파일 업데이트 여부에 따라, 현재 적재된 커널 모듈을 제거하는

리소스 메트릭 수집 시스템.

청구항 9

제6항에 있어서,

VM/컨테이너 관리 모듈은,

클라우드 플랫폼에서 구동 중인 VM 또는 컨테이너를 조회하고, 모니터링 커널 모듈이 적재된 상태에서 VM/컨테이너 스케줄러에 의해 구동 중인 VM 또는 컨테이너의 정보(상기 프로세스 ID 및 상기 리소스 메트릭 이름을 포함함)를 업데이트 하고 상기 VM 또는 컨테이너에 관한 해시 파일로 생성하고, 상기 VM 또는 컨테이너에 관한 해

시 파일 업데이트 여부를 판단하는

리소스 메트릭 수집 시스템.

청구항 10

제9항에 있어서,

VM/컨테이너 관리 모듈은,

상기 VM 또는 컨테이너에 관한 해시 파일 업데이트가 있는 경우, 현재 적재된 커널 모듈을 제거하고,

상기 VM 또는 컨테이너에 관한 해시 파일 업데이트가 없는 경우, 모니터링 커널 모듈이 적재된 상태에서 VM/컨테이너 스케줄러에 의해 구동 중인 VM 또는 컨테이너의 정보를 업데이트 하고 상기 VM 또는 컨테이너에 관한 해시 파일로 생성하는 과정부터 반복하는

리소스 메트릭 수집 시스템.

청구항 11

제6항에 있어서,

커널 모듈 관리 모듈은,

현재 적재된 커널 모듈을 제거한 후 측정 대상이 되는 새로운 VM 또는 컨테이너 정보를 반영하여 커널 모듈을 재적재하는 과정을 수행하기 위해, VM 또는 컨테이너에 해당하는 프로세스 ID 및 리소스 메트릭 데이터베이스에 저장될 리소스 메트릭의 이름을 추출하는 과정부터 반복하는

리소스 메트릭 수집 시스템.

발명의 설명

기 술 분 야

[0001] 본 발명은 클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 방법 및 시스템에 관한 것이다.

배 경 기 술

[0002] 클라우드(Cloud) 컴퓨팅 이란, 시간과 장소에 구애받지 않고 요구하는 성능에 상응하는 요금을 지불하면, 인터넷을 통해 컨테이너 및 가상 머신 형태의 컴퓨팅 자원을 제공 받을 수 있는 기술을 의미한다. 최근 들어, 가상화 기술이 발전함에 따라 클라우드 컴퓨팅 기술의 성능도 함께 발전하고 있으며, 클라우드 컴퓨팅에 대한 수요 및 시장 규모는 빠르게 증가하고 있다. 클라우드 컴퓨팅 기술은 기존에 직접 서비스 제공을 위한 하드웨어 장비를 구매하는 방식인 레거시(Legacy) 시스템과 비교하여 하드웨어 유지 보수 비용을 아낄 수 있다는 장점이 있으며, 자원 사용량에 따라 유동적으로 가상 자원을 요청하여 사용할 수 있어, 컴퓨팅 자원 사용 측면의 비용 절감 효과가 있다.

[0003] 자원 사용량에 따른 유동적인 성능 조절을 위해 필요한 기술은, 클라우드 모니터링 기술이다. 클라우드 모니터링 기술이란, 현재 대상 컨테이너 및 가상 자원의 상태를 측정 및 관찰하는 기술을 의미하며, 자원 사용량을 측정하는 기능은 클라우드 모니터링 기술의 주요 기능 및 목표 중 하나이다. 특히, 자원 사용량은 클라우드 서비스의 요금 정산을 위해 필요한 지표이기 때문에 클라우드 모니터링 기술은 클라우드 환경에서 중요한 역할을 수행하며, 그 중요성을 방증하듯이 클라우드 서비스를 제공하는 대형 서비스 업체인 Google, Amazon, Microsoft 등의 회사들은 각각 Stackdriver, CloudWatch, Azure Monitor 등의 이름을 가진 클라우드 모니터링 서비스를 제공하고 있다.

[0004] 또한, 클라우드 모니터링 기술은 오토 스케일링(Auto Scaling) 기술을 위해서도 필요한 기술이다. 오토 스케일링이란, 자원 사용량에 따라서 사전에 설정된 자원 사용량 임계치에 따라 컴퓨팅 자원을 유동적으로 확장 및 축소하는 기술을 의미한다. 이를 위해선 현재 사용 중인 자원 사용량이 사전 설정된 임계치에 도달 여부 판단이 필요하며, 따라서 현재 자원 사용량 측정을 위해 소요되는 자원 사용량 수집 시간을 최소화하고, 이를 실시간 반영하는 것이 필요하다.

[0005] 이렇게 클라우드 환경에서 중요한 역할을 수행하는 클라우드 모니터링 기술의 성능을 나타내는 지표에는 클라우

드 모니터링 도구 자체에서 발생하는 오버헤드(Overhead) 및 자원 사용량 수집에 소요되는 시간이 있다. 모니터링 도구 자체에서 오버헤드가 과도하게 발생한다면, 그 자체로서도 모니터링 도구를 사용하는 사용자에게 성능상의 부담을 안길 수 있어 모니터링 도구 자체 오버헤드를 낮출 필요가 있다. 또한, 자원 사용량 수집에 오랜 시간이 소요된다면, 적절한 순간에 오토 스케일링 서비스를 제공할 수 없기에 자원 사용량 수집에 소요되는 시간을 줄일 필요가 있다.

[0006] 따라서, VM/컨테이너의 자원 사용량을 수집하는 동안 오버헤드를 거의 발생시키지 않으며, 자원 사용량 수집에 소요되는 시간을 줄일 수 있는 기술이 요구된다.

발명의 내용

해결하려는 과제

[0007] 본 발명이 이루고자 하는 기술적 과제는 클라우드 플랫폼 환경에서 구동되는 가상머신 및 컨테이너의 리소스 사용량 정보를 정확하고, 빠르고, 최소한의 연산으로 리소스 메트릭을 수집할 수 있는 기술적 사상을 제공하는 것이다. 본 발명을 통해 KVM 기반의 가상머신이나, 도커 엔진 기반의 컨테이너가 사용하는 리소스를 초저지연 수집하는 것이 가능하며, 수집 대상이 증가함에도 지연 및 오버헤드의 증가 폭이 현저히 낮아 초경량 모니터링이 가능하다. 따라서 추후 본 발명을 활용할 수 있는 분야에는 VM/컨테이너 오토스케일링, VM/컨테이너 오케스트레이션, AI 학습을 위한 데이터 생성 등이 있다.

과제의 해결 수단

[0008] 일 측면에 있어서, 본 발명에서 제안하는 클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 방법은 클라우드 플랫폼에서 구동 중인 VM 또는 컨테이너를 가상화 관리 모듈을 통해 조회하는 단계, 가상화 관리 모듈을 통해 조회된 정보에 기초하여 VM 또는 컨테이너에 해당하는 프로세스 ID 및 리소스 메트릭 데이터베이스에 저장될 리소스 메트릭의 이름을 추출하는 단계, 추출된 프로세스 ID 및 리소스 메트릭 이름을 매개변수로 전달하여 리눅스 커널에 리소스 메트릭 수집을 위한 커널 모듈을 적재하는 단계, 모니터링 커널 모듈이 적재된 상태에서 VM/컨테이너 스케줄러에 의해 구동 중인 VM 또는 컨테이너의 정보를 업데이트 하고 해시 파일로 생성하는 단계, VM 또는 컨테이너의 해시 업데이트 여부를 판단하는 단계 및 VM 또는 컨테이너의 해시 업데이트가 있는 경우, 현재 적재된 커널 모듈을 제거하는 단계를 포함한다.

[0009] VM 또는 컨테이너의 해시 업데이트가 없는 경우, 모니터링 커널 모듈이 적재된 상태에서 VM/컨테이너 스케줄러에 의해 구동 중인 VM 또는 컨테이너의 정보를 업데이트 하고 해시 파일로 생성하는 단계부터 반복한다.

[0010] 현재 적재된 커널 모듈을 제거한 후 측정 대상이 되는 새로운 VM 또는 컨테이너 정보를 반영하여 커널 모듈을 재적재하는 과정을 수행하기 위해, VM 또는 컨테이너에 해당하는 프로세스 ID 및 리소스 메트릭 데이터베이스에 저장될 리소스 메트릭의 이름을 추출하는 단계부터 반복한다.

[0011] 가상화 관리 모듈을 통해 조회된 정보에 기초하여 VM 또는 컨테이너에 해당하는 프로세스 ID 및 리소스 메트릭 데이터베이스에 저장될 리소스 메트릭의 이름을 추출하는 단계는 커널 레벨에서 구동되는 프로세스 리소스 수집 모듈이 커널 모듈 관리 모듈로부터 프로세스 ID를 입력 받고, 입력 받은 프로세스 ID의 테스크 구조체에 접근하여, 테스크에 대한 MM(Memory Management) 구조체의 주소를 가져와 프로세스 별 리소스 사용량에 접근하기 위해 MM 포인터 주소로부터 메모리 컨트롤러 데이터 구조체의 주소를 획득한다.

[0012] 메모리 사용량 관련 리소스 메트릭 정보를 획득하기 위해 메모리 컨트롤러 데이터 구조체의 주소와 리소스 메트릭을 전달하고, 해당 프로세스 ID 리소스 사용량을 반환하여, 페이지 처리 관련 리소스 메트릭 정보를 획득하기 위해 해당 프로세스 ID의 페이지 처리 관련 값을 반환한다.

[0013] 또 다른 일 측면에 있어서, 본 발명에서 제안하는 클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 시스템은 복수의 리소스 정보를 하나의 Proc 파일 시스템에 대응시키는 Proc 파일 시스템 생성 모듈, 유저 레벨에서 리소스 메트릭의 수집 대상이되는 프로세스의 ID 정보를 전달받아, 대상 프로세스에 대한 리소스 메트릭을 수집하는 프로세스 리소스 수집 모듈, 현재 구동중인 VM/컨테이너의 프로세스 ID를 획득하는 프로세스 추출 모듈, 커널 레벨에서 동작하는 커널 모듈을 동적으로 적재 및 제거 하는 커널 모듈 관리 모듈, 현재 클라우드 환경에서 구동 중인 VM/컨테이너에 대한 정보를 획득하는 VM/컨테이너 관리 모듈 및 커널 모듈로부터 수집 한 리소스 데이터를 저장하는 리소스 메트릭 데이터베이스를 포함한다.

발명의 효과

[0014] 본 발명의 실시예들에 따르면 측정 대상이 증가함에도 자원 사용량에 관한 정보를 단일 자원 사용량 정보 파일에 저장 및 관리를 수행하기 때문에 자원 사용량 측정 과정에서 리소스 정보 파일까지 도달하는 경로 탐색 과정을 단축하여 이에 따라 발생하는 오버헤드를 감소시킬 수 있다. 또한, 리소스 사용정보 수집을 위한 경로 탐색 과정이 단축되어 자원 사용량 측정에 소요되는 시간을 줄일 수 있으며, 이에 따라 세밀한 자원 사용량 측정을 가능하게 만드는 효과가 있다.

도면의 간단한 설명

[0015] 도 1은 본 발명의 일 실시예에 따른 초-저지연 리소스 메트릭 수집 기술이 적용되는 클라우드 환경이다.

도 2는 본 발명의 일 실시예에 따른 클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 시스템의 구성을 나타내는 도면이다.

도 3은 본 발명의 일 실시예에 따른 클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 방법을 설명하기 위한 흐름도이다.

도 4는 본 발명의 일 실시예에 따른 클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 시스템의 커널 레벨에서 프로세스 리소스 메트릭을 수집하는 과정을 설명하기 위한 도면이다.

도 5는 본 발명의 일 실시예에 따른 클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 시스템에서 리소스 메트릭을 수집하는 과정이다.

도 6은 본 발명의 일 실시예에 따른 클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 시스템을 통해 수집하는 메트릭 데이터의 예시이다.

발명을 실시하기 위한 구체적인 내용

[0016] 본 발명은 클라우드 플랫폼 환경에서 효율적인 리소스 메트릭 모니터링 모듈에 관한 것으로서, 보다 상세하게는 기존의 리눅스 환경에서 구동되는 프로세스들의 리소스 사용량 측정 과정에서 발생하는 불필요한 파일 접근 과정을 최소화하고, 현재 구동 중인 다수의 프로세스들의 리소스 사용량 정보를 하나의 리소스 정보 파일에 기록하는 방법 및 리소스 메트릭 모니터링을 수행하는 모듈에 관한 것이다.

[0017] 여기서 사용되는 용어 및 개념은 다음과 같다:

[0018] 클라우드(클라우드 환경, 클라우드 컴퓨팅): 클라우드란 구름(Cloud)와 같이 무형의 형태로 존재하는 하드웨어, 소프트웨어 등의 컴퓨팅 자원을 자신이 필요한 만큼 빌려 쓰고 이에 대한 사용요금을 지급하는 방식의 컴퓨팅 서비스로, 서로 다른 물리적인 위치에 존재하는 컴퓨팅 자원을 가상화 기술로 통합해 제공하는 환경을 말한다.

[0019] 커널(리눅스 커널, 커널 레벨): 여기서 의미하는 커널은 리눅스 환경에서 사용하는 커널을 의미한다. 커널은 리눅스 운영체제의 핵심적인 부분을 의미하며, 하드웨어 장치(디바이스)에 직접적인 접근이 가능하다. 커널 레벨에서의 오류는 시스템 전체에서 오류를 야기할 수 있기 때문에, 이러한 보안 문제로 커널로의 직접적인 접근은 제한된다.

[0020] 커널모듈(모듈): 일반적으로 커널은 관련 코드 및 기능을 수정할 시에, 커널 전체의 재 컴파일이 필요하다. 하지만, c언어 형태(.c 확장자)로 작성된 커널 모듈 코드를 모듈형태로 컴파일하면(.ko 확장자)를 활용하면 커널 전체의 재컴파일 없이도 간단하게 커널 기능을 확장하는 것이 가능하다. 특정 명령어를 통해 필요시에만 커널 모듈을 커널에 적재하는 것이 가능하며(insmod 명령어), 필요가 없다면 다시 제거 가능하다(rmmod). 커널 모듈 코드는 커널 상에서 작동하기 때문에 매우 빠르게 작동하며 오버헤드가 현저히 낮다는 장점이 있다. 본 발명의 자원 사용량 모니터링 모듈은 커널 모듈 형태를 취하고 있다. 본 발명에서 사용하는 모니터링 커널 모듈을 커널에 적재 시에 모니터링 대상이 되는 VM/컨테이너의 PID정보를 함께 입력해야 해당 VM/컨테이너에 대한 자원 사용량 수집이 가능하다. 만약, 측정 대상 VM/컨테이너에 대한 정보가 변경된다면, 새로운 PID를 반영이 필요하기 때문에, 적재된 커널 모듈을 제거하고 재 적재하는 과정이 필요하다. 이 과정을 위해 본 발명에선 'VM/컨테이너 스케줄러' 가 존재함

[0021] VMI(Virtual Machine Introspection): 가상머신의 내부 메모리에 읽기 혹은 쓰기를 위한 기술을 의미한다. 추가적으로 CPU 레지스터에 접근하는 것도 가능하며 가상머신의 정지 혹은 재구동, 바이너리 데이터 출력 등의 가

상화 내부에 대해 매우 깊은 수준의 탐색 및 제어가 가능하다. 본 발명에선 구동 중인 process의 PID정보 등을 가져오는데 사용한다.

[0022] Proc 파일 시스템: Proc 파일 시스템은 일반적으로 사용자 레벨에서 접근이 제한된 커널레벨의 정보에 쉽게 접근하기 위해 생성된 Proc 파일 시스템이다. Proc 파일시스템에서 Proc은 Process의 줄임말이며 프로세스 정보뿐만 아닌 시스템 전반의 상세한 정보(cpu, 메모리 등의 자원 사용량 정보)를 광범위하게 제공한다. Proc 파일 시스템 자체가 기존 파일 시스템이 가지고있는 상당한 오버헤드를 줄이기 위해 리눅스 커널 메모리상에서 직접 파일시스템을 관리하는 방법을 채택하고 있어, 오버헤드가 현저히 낮고 및 속도가 매우 빠르다. 일반적으로 제공하는 정보 외에도 사용자가 직접 리눅스 커널 모듈을 통해 Proc 파일 시스템을 작성함으로써 원하는 자원 사용량 정보 획득이 가능하다.

[0023] 메트릭(리소스 메트릭): 클라우드 환경에서 사용하는 메트릭의 의미는 자원 사용량 데이터를 의미한다(CPU 사용량, Memory 사용량 등).

[0024] 프로세스 구조체: 본 발명에서 사용하는 프로세스 구조체의 의미는 프로세스에 대한 정보가 모여있는 구조체를 의미한다.

[0025] 해시: 임의의 길이의 데이터를 입력 받아 일정한 길이의 비트열로 반환시켜주는 함수의 결과 값이다. 해시 함수의 특성상 입력 값의 아주 작은 변화로도 전혀 다른 결과가 도출되며, 이러한 특성을 이용하여 본 발명에서 VM/컨테이너 스케쥴러는, VM/컨테이너에 대한 전체 정보를 해시값으로 관리하고, 이 값에 변화가 생기면 측정 대상 VM/컨테이너를 재 반영하기 위해서 모니터링 커널 모듈의 제거 및 바뀐 VM/컨테이너 정보를 반영하여 재적재를 수행한다.

[0026] 이하, 본 발명의 실시 예를 첨부된 도면을 참조하여 상세하게 설명한다.

[0028] 도 1은 본 발명의 일 실시예에 따른 초-저지연 리소스 메트릭 수집 기술이 적용되는 클라우드 환경이다.

[0029] 도 1을 참조하면, 본 발명의 실시 예에 따른 초-저지연 리소스 메트릭 수집 시스템은 원격에 존재하는 클라우드 플랫폼(101)과 클라우드 플랫폼 서비스를 관리하는 관리자(102)가 존재하는 환경에서 적용된다.

[0031] 도 2는 본 발명의 일 실시예에 따른 클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 시스템의 구성을 나타내는 도면이다.

[0032] 도 2를 참조하면, 본 발명의 실시 예에 따른 클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 시스템은 커널 레벨(210)에서 동작하는 Proc 파일 시스템 생성 모듈(211), 프로세스 리소스 수집 모듈(212)과 유저 레벨(220)에서 동작하는 프로세스 추출 모듈(221), 커널 모듈 관리 모듈(222), VM/컨테이너 관리 모듈(223), 리소스 메트릭 데이터베이스(224)를 포함한다.

[0033] Proc 파일 시스템 생성 모듈(211)은 복수의 리소스 정보를 하나의 Proc 파일 시스템에 대응시킨다.

[0034] 프로세스 리소스 수집 모듈(212)은 유저 레벨에서 리소스 메트릭의 수집 대상이되는 프로세스의 ID 정보를 전달받아, 대상 프로세스에 대한 리소스 메트릭을 수집한다.

[0035] 프로세스 추출 모듈(221)은 현재 구동중인 VM/컨테이너의 프로세스 ID를 획득한다. 프로세스 추출 모듈(221)은 VM/컨테이너 관리 모듈을 통해 조회된 정보에 기초하여 VM 또는 컨테이너에 해당하는 프로세스 ID 및 리소스 메트릭 데이터베이스에 저장될 리소스 메트릭의 이름을 추출한다.

[0036] 커널 모듈 관리 모듈(222)은 커널 레벨에서 동작하는 커널 모듈을 동적으로 적재 및 제거한다. 커널 모듈 관리 모듈(222)은 프로세스 추출 모듈을 통해 추출된 프로세스 ID 및 리소스 메트릭 이름을 매개변수로 전달하여 리눅스 커널에 리소스 메트릭 수집을 위한 커널 모듈을 적재하도록 하고, VM 또는 컨테이너의 해시 업데이트 여부에 따라, 현재 적재된 커널 모듈을 제거한다.

[0037] VM/컨테이너 관리 모듈(223)은 현재 클라우드 환경에서 구동 중인 VM/컨테이너에 대한 정보를 획득한다. VM/컨테이너 관리 모듈(223)은 클라우드 플랫폼에서 구동 중인 VM 또는 컨테이너를 조회하고, 모니터링 커널 모듈이 적재된 상태에서 VM/컨테이너 스케줄러에 의해 구동 중인 VM 또는 컨테이너의 정보를 업데이트 하고 해시 파일로 생성하고, VM 또는 컨테이너의 해시 업데이트 여부를 판단한다.

[0038] VM/컨테이너 관리 모듈(223)은 VM 또는 컨테이너의 해시 업데이트가 있는 경우, 현재 적재된 커널 모듈을 제거한다. VM 또는 컨테이너의 해시 업데이트가 없는 경우, 모니터링 커널 모듈이 적재된 상태에서 VM/컨테이너 스

케줄러에 의해 구동 중인 VM 또는 컨테이너의 정보를 업데이트 하고 해시 파일로 생성하는 단계부터 반복한다.

[0039] 리소스 메트릭 데이터베이스(224)는 커널 모듈로부터 수집 한 리소스 데이터를 저장한다.

[0041] 도 3은 본 발명의 일 실시예에 따른 클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 방법을 설명하기 위한 흐름도이다.

[0042] 제안하는 클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 방법은 클라우드 플랫폼에서 구동 중인 VM 또는 컨테이너를 가상화 관리 모듈을 통해 조회하는 단계(310), 가상화 관리 모듈을 통해 조회된 정보에 기초하여 VM 또는 컨테이너에 해당하는 프로세스 ID 및 리소스 메트릭 데이터베이스에 저장될 리소스 메트릭의 이름을 추출하는 단계(320), 추출된 프로세스 ID 및 리소스 메트릭 이름을 매개변수로 전달하여 리눅스 커널에 리소스 메트릭 수집을 위한 커널 모듈을 적재하는 단계(330), 모니터링 커널 모듈이 적재된 상태에서 VM/컨테이너 스케줄러에 의해 구동 중인 VM 또는 컨테이너의 정보를 업데이트 하고 해시 파일로 생성하는 단계(340), VM 또는 컨테이너의 해시 업데이트 여부를 판단하는 단계(350) 및 VM 또는 컨테이너의 해시 업데이트가 있는 경우, 현재 적재된 커널 모듈을 제거하는 단계(360)를 포함한다.

[0043] 단계(310)에서, 현재 클라우드 플랫폼(리눅스 운영체제)에서 VMI와 같은 가상화 관리 모듈을 통해 구동 중인 VM 또는 컨테이너들을 조회한다.

[0044] 단계(320)에서, 가상화 관리 모듈을 통해 조회된 정보에 기초하여 VM 또는 컨테이너에 해당하는 프로세스 ID 및 리소스 메트릭 데이터베이스에 저장될 리소스 메트릭의 이름을 추출한다. 예를 들어, 추출된 형태는 [모니터링 대상 고유ID_메트릭 이름]으로 구성될 수 있다.

[0045] 단계(330)에서, 추출된 프로세스 ID 및 리소스 메트릭 이름을 매개변수로 전달하여 리눅스 커널에 리소스 메트릭 수집을 위한 커널 모듈을 적재한다.

[0046] 단계(340)에서, 모니터링 커널 모듈이 적재된 상태에서 VM/컨테이너 스케줄러에 의해 구동 중인 VM 또는 컨테이너의 정보를 업데이트 하고 해시 파일로 생성한다. 모니터링 커널 모듈이 커널 상에 적재되면, 모니터링 커널 모듈은 구동 중인 VM 또는 컨테이너들의 리소스 사용 정보(메트릭)를 Proc 파일 시스템 영역에서 업데이트한다. 모니터링 커널 모듈이 적재된 상태에서 스케줄러에 의해 구동 중인 VM 또는 컨테이너들의 정보가 해시 파일로 생성된다.

[0047] 단계(350)에서, VM 또는 컨테이너의 해시 업데이트 여부를 판단한다. 리눅스 커널 모듈은 그 특성상 모니터링 대상이 되는 PID정보를 새로 반영하기 위해서는 기존 모니터링 모듈 제거 후 측정 대상이 되는 새로운 VM/컨테이너 정보를 반영하여 재적재 과정이 필요하다. 해시 파일을 생성함은, 자원 사용량 모니터링의 대상 VM/컨테이너의 변화 여부를 알아보기 위해서이다. 해시의 특성상 측정 대상 VM/컨테이너의 수, 측정 대상이 되는 VM/컨테이너의 새로운 생성, 종료 등에 의한 PID 등에 변화가 생기면 해시 파일 상에서 변화가 나타나고, VM/컨테이너 스케줄러는 이를 감지하여 모니터링 커널 모듈의 재적재 과정을 수행한다.

[0048] 단계(360)에서, VM 또는 컨테이너의 해시 업데이트가 있는 경우, 현재 적재된 커널 모듈을 제거한다. 만일, 상기 시스템 내의 해시의 업데이트 없다면, 리소스 메트릭 수집 시스템의 흐름은 다시 VM/컨테이너 스케줄러로 돌아가 기존 수행하던 자원 사용량 모니터링 과정을 계속 수행하며, 해시 파일이 업데이트가 확인되면(다시 말해, 해시 파일 상의 변화가 확인되면), 새로운 VM/컨테이너의 PID를 반영하여 커널 모듈을 재적재하는 과정을 수행하기 위해, 현재 적재된 커널 모듈을 제거(306)하고, VM 또는 컨테이너에 해당하는 프로세스 ID 및 리소스 메트릭 데이터베이스에 저장될 리소스 메트릭의 이름을 추출하는 단계(320)부터 반복하여 업데이트된 VM/컨테이너 정보를 반영한다.

[0050] 도 4는 본 발명의 일 실시예에 따른 클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 시스템의 커널 레벨에서 프로세스 리소스 메트릭을 수집하는 과정을 설명하기 위한 도면이다.

[0051] 도 4를 참조하면, 본 발명의 실시예에 따른 클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 시스템의 커널 레벨에서 리소스 메트릭을 수집하는 과정은 다음과 같다.

[0052] 커널 레벨에서 구동되는 프로세스 리소스 수집 모듈(420)은 특정 프로세스의 리소스 사용량을 수집하기 위해 사용자 영역에서 구동되는 커널 모듈 관리 모듈(430)을 통해 프로세스 ID(PID)를 입력 받는다(421). 입력 받은 PID의 테스크(Task) 구조체(Process Descriptor)에 접근한다(422). 테스크 구조체에 접근 후, 테스크에 대한 MM(Memory Management) 구조체(Memory Descriptor)의 주소를 가져온다(423). 프로세스 별 리소스 사용량에 접근하기 위해 MM 포인터 주소로부터 get_mem_cgroup_from_mm 함수를 활용하여 mem_cgroup(메모리 컨트롤러 데이

터 구조) 주소를 획득한다(424). 메모리 사용량 관련 리소스 메트릭 정보를 획득하기 위해서 memcg_page_state 함수에 획득하고자 하는 mem_cgroup 주소와 리소스 메트릭을 전달하면 해당 PID 리소스 사용량을 반환한다(425). 페이지 처리 관련 리소스 메트릭 정보를 획득하기 위해서 per_cpu 함수에 획득하고자 하는 per_cpu 주소와 리소스 메트릭을 전달하면 해당 PID의 페이지 처리 관련 값을 반환한다 (426). Memory Management에서 제공하는 모든 리소스 메트릭 정보는 본 발명의 리소스 메트릭 모니터링 커널 모듈을 통해 모두 수집 할 수 있다(427). 리소스 메트릭 관려하여 반환 받은 값은 Proc 파일 시스템 생성 모듈(410)에서 생성한 Proc 파일 시스템 내부에 업데이트 된다(428).

[0054] 도 5는 본 발명의 일 실시예에 따른 클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 시스템에서 리소스 메트릭을 수집하는 과정이다.

[0055] 도 5를 참조하면, 본 발명의 실시 예에 따른 클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 시스템에서 리소스 메트릭을 수집하는 과정은 다음과 같다. 앞서 설명된 바와 같이 유저 레벨(520)은 프로세스 추출 모듈(521), 커널 모듈 관리 모듈(522), VM/컨테이너 관리 모듈(523) 및 리소스 메트릭 데이터베이스(524)를 포함한다.

[0056] 유저 레벨(520)에서 구동되는 커널 모듈 관리 모듈(522)을 통해 프로세스 ID와 리소스 메트릭 네이밍 정보를 매개변수로 전달하여 커널 레벨(510)에서 구동되는 커널 모듈을 커널에 적재시킨다(531). 커널 모듈은 전달받은 매개변수를 반영하여 리소스 메트릭 정보를 Proc 파일 시스템 영역에 metric(/proc/metric)이라는 파일명으로 생성한다(532). 상기 metric 파일은 현재 구동중인 VM 또는 컨테이너의 리소스 메트릭 정보가 실시간으로 업데이트된다. 리소스 메트릭 데이터베이스는 metric 파일이 Proc 파일 시스템 영역에 생성되면, 리소스 메트릭 정보를 읽어와 데이터베이스에 저장한다(533).

[0058] 도 6은 본 발명의 일 실시예에 따른 클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 시스템을 통해 수집하는 메트릭 데이터의 예시이다.

[0059] 도 6을 참조하면, 측정 대상 자원 사용량 정보는 다음과 같다. MemTotal은 전체 사용 가능한 총 메모리이다(601). SwapTotal은 사용 가능한 총 스왑 공간이다(602). Active_file은 가장 최근에 사용되었고 일반적으로 필요할 때까지 재생되지 않은 캐시 메모리이다(603). Cached는 페이지의 캐시 메모리이다(604). SwapCache는 주 메모리 내에 있지만 스왑 파일에도 존재하는 메모리이다(605). working_set은 현재 working set의 크기이다(606). Inactive_anon은 최근에 사용되지 않았으며 교체 가능할 수 있는 익명 메모리이다(607). Mem_Usage는 현재 메모리의 사용량이다(608). SwapFree는 남은 스왑 공간이다(609). Inactive_file 성능 영향 없이 재생할 수 있는 캐시 메모리이다(610). Active_anon은 최근에 사용되었고 일반적으로 스왑 아웃되지 않은 익명 메모리이다(611). Dirty는 디스크에 다시 기록 대기 중인 메모리이다(612). Writeback는 디스크에 다시 쓰여지고 있는 메모리이다(613).

[0060] 본 발명의 실시예에 따라 상기 클라우드 플랫폼 환경에서의 초-저지연 리소스 메트릭 수집 시스템은 사용자 정의에 따라 리소스 메트릭 정보를 추가/삭제 할 수 있다.

[0062] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPA(field programmable array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0063] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나

처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

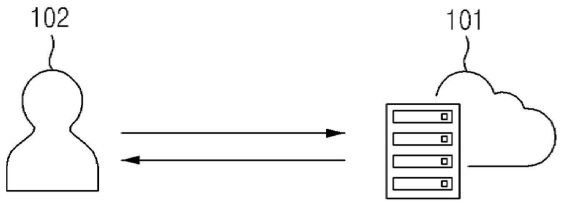
[0064] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

[0065] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

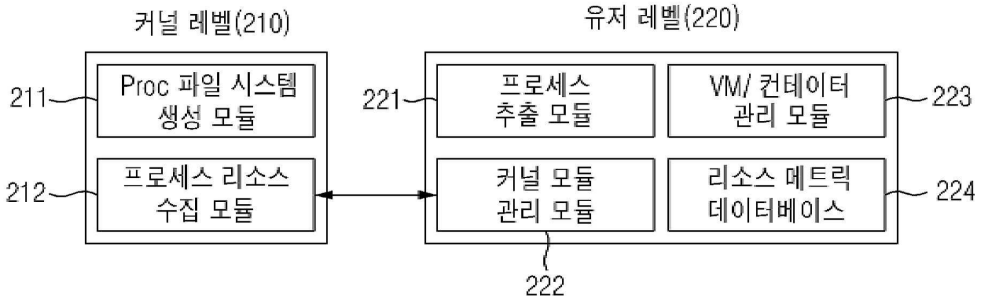
[0066] 그러므로, 다른 구현들, 다른 실시예들 및 발명청구범위와 균등한 것들도 후술하는 발명청구범위의 범위에 속한다.

도면

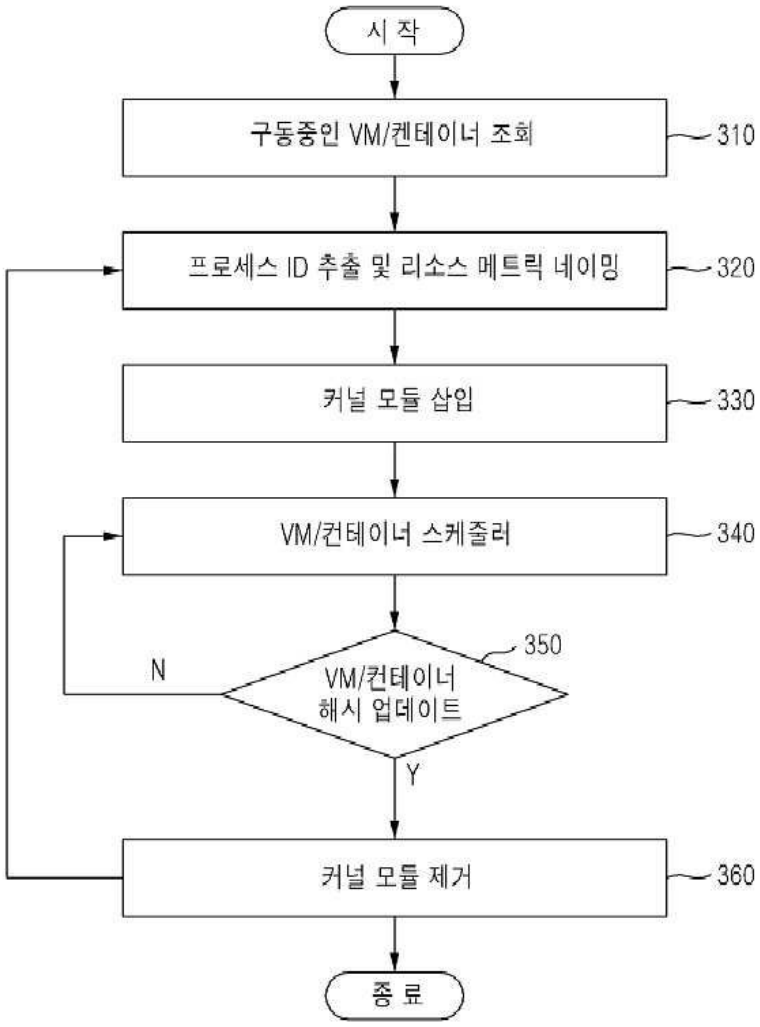
도면1



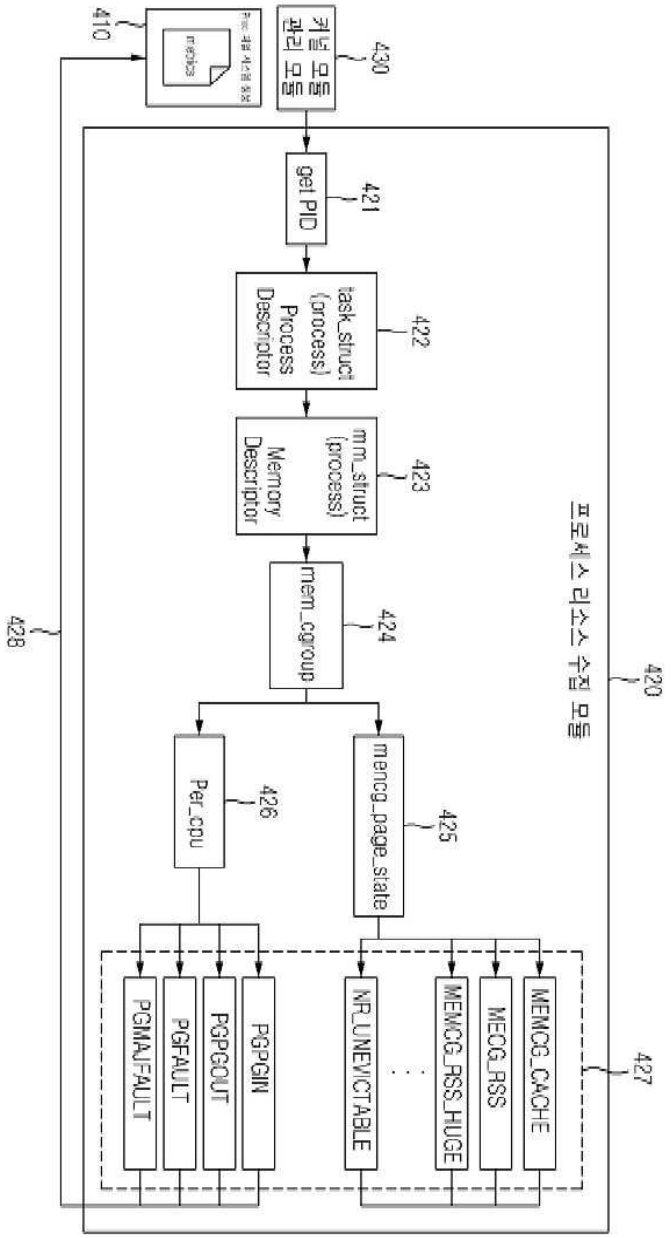
도면2



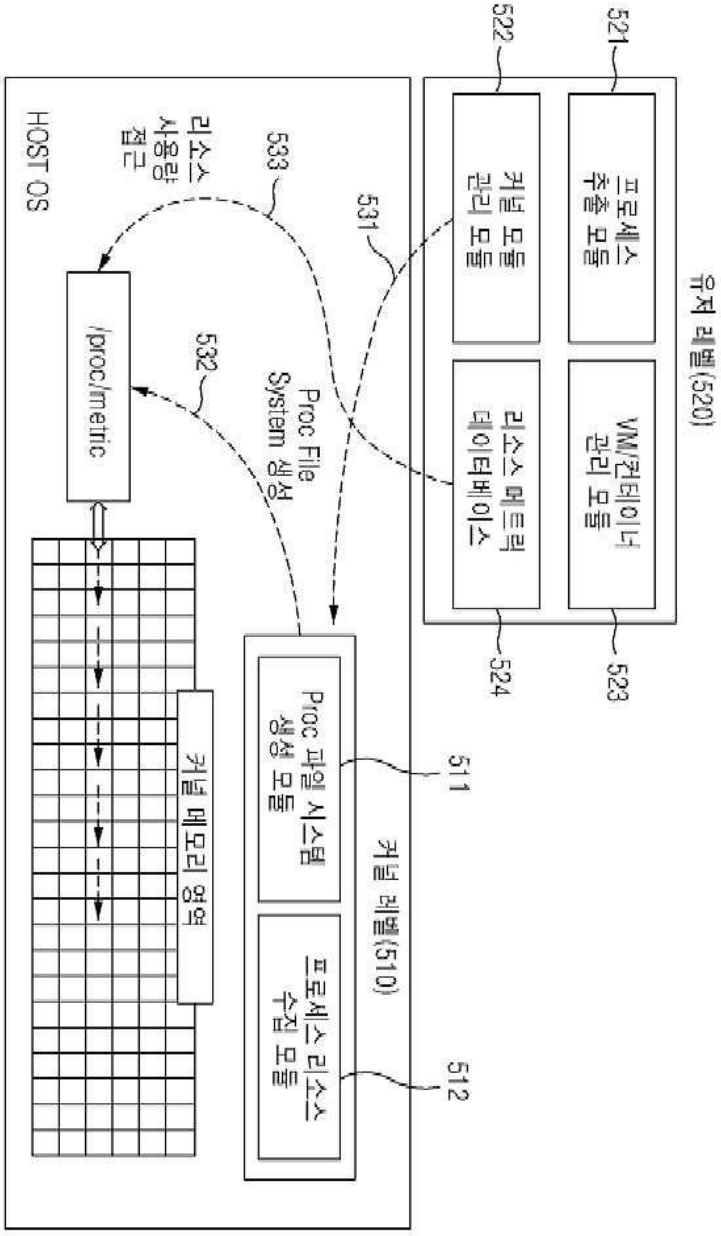
도면3


시 작
구동중인 VM/켄테이너 조회
310
프로세스 ID 추출 및 리소스 메트릭 네이밍
320
커널 모듈 삽입
330
VM/컨테이너 스케줄러
340
350
VM/컨테이너
해시 업데이트
N
Y
커널 모듈 제거
360
종 료

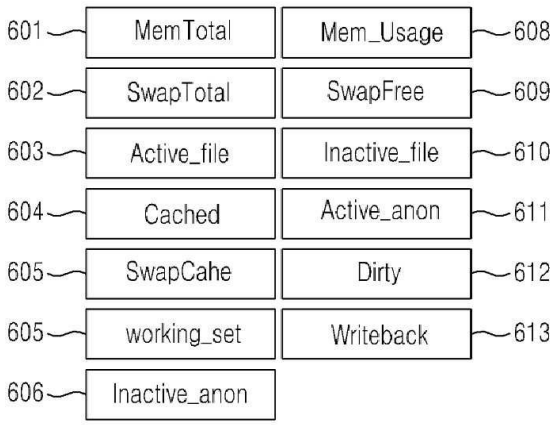
도면4


410
커널 모듈 관리 모듈
430
get PID
421
task_struct (process) Process Descriptor
422
mm_struct (process) Memory Descriptor
423
mem_cgroup
424
Per_cpu
426
memcg_page_state
425
PGMAJFAULT
PGFAULT
PGPGOUT
PGPGIN
NR_UNEVICTABLE
MEMCG_RSS_HUGE
MECG_RSS
MEMCG_CACHE
427
420
프로세스 리소스 수집 모듈
428

도면5


HOST OS
리소스 사용량 접근
533
531
532
/proc/metric
Proc File System 생성
커널 메모리 영역
Proc 파일 시스템 생성 모듈
511
프로세스 리소스 수집 모듈
512
커널 레벨(510)
커널 모듈 관리 모듈
522
리소스 메트릭 데이터베이스
524
프로세스 추출 모듈
521
VM/컨테이너 관리 모듈
523
유저 레벨(520)

도면6


601
MemTotal
Mem_Usage
608
602
SwapTotal
SwapFree
609
603
Active_file
Inactive_file
610
604
Cached
Active_anon
611
605
SwapCahe
Dirty
612
605
working_set
Writeback
613
606
Inactive_anon