(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0074537
(43) 공개일자 2022년06월03일

(51) 국제특허분류(Int. Cl.)
G06F 9/46 (2006.01) *G06F 9/455* (2018.01)
G06F 9/52 (2018.01)
(52) CPC특허분류
G06F 9/466 (2013.01)
G06F 9/45545 (2013.01)
(21) 출원번호 10-2020-0163144
(22) 출원일자 2020년11월27일
심사청구일자 2020년11월27일

(71) 출원인
이화여자대학교 산학협력단
서울특별시 서대문구 이화여대길 52 (대현동, 이화여자대학교)
(72) 발명자
반효경
서울특별시 관악구 관악로30길 12, 107동 2001호 (봉천동, 관악우성아파트)
조경운
서울특별시 서초구 방배천로2안길 118(방배동)
(74) 대리인
특허법인 무한

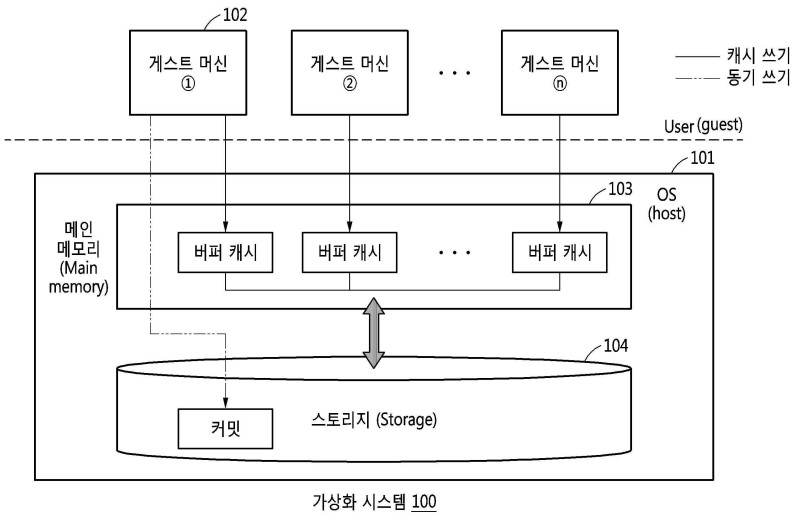
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 **가상 머신 환경에서의 블록 커밋 방법 및 그 방법을 수행하는 가상화 시스템**

(57) 요 약

본 발명은 가상 머신 환경에서의 블록 커밋 방법 및 그 방법을 수행하는 가상화 시스템에 관한 것으로, 자세하게, 블록 커밋 방법은 해시 테이블을 기반으로 복수의 게스트 머신 각각에서 요청된 수정 블록을 각각 구분하고, 커밋을 요청한 게스트 머신의 수정 블록만을 호스트 머신의 스토리지에 커밋하기 위해 트랜잭션을 분할하여 수정 블록의 커밋을 관리한다.

대 표 도 - 도1



가상화 시스템 100

(52) CPC특허분류
G06F 9/463 (2013.01)
G06F 9/52 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711103102
과제번호	2019-0-00074-002
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	SW컴퓨팅산업원천기술개발(R&D,정보화)
연구과제명	워크로드 특성에 동적으로 학습하는 차세대메모리 기반 시스템 SW 기술 개발
기 여 율	1/1
과제수행기관명	이화여자대학교산학협력단
연구기간	2020.01.01 ~ 2020.12.31

명 세 서

청구범위

청구항 1

호스트 머신의 커밋 주기(t)에 따른 복수의 게스트 머신(VM: Virtual Machine)으로부터 요청된 수정 블록을 포함하는 러닝 트랜잭션을 관리하는 단계;

상기 복수의 게스트 머신에 의한 커밋 트리거가 발생하면, 커밋 트리거에 의한 수정 블록을 포함하는 러닝 트랜잭션을 커밋 트랜잭션으로 전환하는 단계; 및

상기 전환된 커밋 트랜잭션에 포함되지 않은 나머지 수정 블록을 포함하는 러닝 트랜잭션을 유지하는 단계;

를 포함하는 블록 커밋 방법.

청구항 2

제1항에 있어서,

상기 러닝 트랜잭션을 관리하는 단계는,

상기 복수의 게스트 머신 각각으로부터 동기 쓰기(Sync Write)로 요청된 수정 블록이 연결 리스트로 연결된 러닝 트랜잭션을 관리하는 블록 커밋 방법.

청구항 3

제1항에 있어서,

상기 러닝 트랜잭션을 관리하는 단계는,

해시 테이블(Hash Table)에 따른 복수의 게스트 머신 각각으로부터 요청된 수정 블록의 아이노드(Inode)를 이용하여 상기 복수의 게스트 머신 각각에 따라 구분된 수정 블록을 포함하는 러닝 트랜잭션으로 관리하는 블록 커밋 방법.

청구항 4

제1항에 있어서,

상기 커밋 트랜잭션으로 전환하는 단계는,

상기 복수의 게스트 머신 중 어느 하나의 게스트 머신의 커밋 주기에 따른 커밋 트리거가 발생되면, 상기 커밋 트리거가 발생한 게스트 머신이 요청한 수정 블록을 포함하는 러닝 트랜잭션을 커밋 트랜잭션으로 전환하는 블록 커밋 방법.

청구항 5

제1항에 있어서,

상기 커밋 트랜잭션으로 전환하는 단계는,

상기 호스트 머신의 커밋 주기가 도래하는 경우, 러닝 트랜잭션에 포함된 모든 수정 블록을 커밋 트랜잭션의 목록으로 전환하는 블록 커밋 방법.

청구항 6

제1항에 있어서,

상기 러닝 트랜잭션을 유지하는 단계는,

상기 커밋 트랜잭션에 포함되지 않은 나머지 수정 블록 및/또는 호스트의 커밋 주기(t+1)에서 복수의 게스트 머신으로부터 요청된 수정 블록을 연결 리스트로 연결하여 러닝 트랜잭션으로 유지하는 블록 커밋 방법.

청구항 7

프로세서를 포함하는 가상화 시스템에 있어서,

상기 프로세서는,

호스트 머신의 커밋 주기(t)에 따른 복수의 게스트 머신으로부터 요청된 수정 블록을 포함하는 러닝 트랜잭션을 관리하고,

상기 복수의 게스트 머신에 의한 커밋 트리거가 발생하면, 커밋 트리거에 의한 수정 블록을 포함하는 러닝 트랜잭션을 커밋 트랜잭션으로 전환하고,

상기 전환된 커밋 트랜잭션에 포함되지 않은 나머지 수정 블록을 포함하는 러닝 트랜잭션을 유지하는 가상화 시스템.

청구항 8

제7항에 있어서,

상기 프로세서는,

상기 복수의 게스트 머신 각각으로부터 동기 쓰기로 요청된 수정 블록이 연결 리스트로 연결된 러닝 트랜잭션을 관리하는 가상화 시스템.

청구항 9

제7항에 있어서,

상기 프로세서는,

해시 테이블에 따른 복수의 게스트 머신 각각으로부터 요청된 수정 블록의 아이노드를 이용하여 상기 복수의 게스트 머신 각각에 따라 구분된 수정 블록을 포함하는 러닝 트랜잭션으로 관리하는 가상화 시스템.

청구항 10

제7항에 있어서,

상기 프로세서는,

상기 복수의 게스트 머신 중 어느 하나의 게스트 머신의 커밋 주기에 따른 커밋 트리거가 발생되면, 상기 커밋 트리거가 발생한 게스트 머신이 요청한 수정 블록을 포함하는 러닝 트랜잭션을 커밋 트랜잭션으로 전환하는 가상화 시스템.

청구항 11

제7항에 있어서,

상기 프로세서는,

상기 호스트 머신의 커밋 주기가 도래하는 경우, 러닝 트랜잭션에 포함된 모든 수정 블록을 커밋 트랜잭션의 목록으로 전환하는 블록 커밋 방법.

청구항 12

제7항에 있어서,

상기 프로세서는,

상기 커밋 트랜잭션에 포함되지 않은 나머지 수정 블록 및/또는 호스트의 커밋 주기(t+1)에서 복수의 게스트 머신으로부터 요청된 수정 블록을 연결 리스트로 연결하여 러닝 트랜잭션으로 유지하는 가상화 시스템.

발명의 설명

기 술 분 야

[0001] 본 발명은 가상 머신 환경에서의 블록 커밋 방법 및 그 방법을 수행하는 가상화 시스템에 관한 것으로, 보다 구체적으로는 가상 머신 환경에서의 호스트 머신과 복수의 게스트 머신 간에 발생되는 커밋 요청에 따른 트랜잭션을 관리하는 장치 및 방법에 관한 것이다.

배 경 기 술

[0002] 일반적으로, 저널링 파일 시스템은 호스트 시스템이 예상치 못한 문제의 발생으로 중지되었을 경우, 스토리지의 내용을 복구하기 위한 동작을 수행한다. 예를 들어, Ext4, ReiserFS, Btrfs, ZFS 등 저널링 파일 시스템은 모든 수정된 파일 블록들을 단일 트랜잭션으로 관리한다. 저널링 파일 시스템은 단일 트랜잭션을 이용하여 모든 수정된 파일 블록들을 주기적으로 스토리지의 저널 영역에 커밋(commit)한다.

[0003] 이러한 동작에 따른 가상 머신 환경에서의 저널링 파일 시스템은 하나의 게스트 머신으로부터 커밋이 요청되는 경우, 커밋을 원하지 않는 게스트 머신의 요청과 무관하게 호스트 머신 상에 존재하는 모든 게스트 머신의 수정 블록들에 대한 커밋을 수행한다.

[0004] 뿐만 아니라, 저널링 파일 시스템에서는 미리 정의된 커밋 주기가 도래하거나 응용 프로그램에서 명시적으로 동기적 쓰기 요청을 하는 경우, 수정된 모든 수정 블록을 스토리지에 커밋한다. 이때, 주기적 커밋은 갑작스런 전원 오류 등으로부터 수정된 데이터가 증발하는 것을 막는 역할을 한다.

[0005] 그러나, 게스트 머신의 파일 시스템이 발생시키는 주기적인 커밋 요청에 있어, 호스트 머신에서는 응용의 명시적인 동기적 쓰기 요청으로 관찰됨에 따라 모든 게스트 머신이 5초 간격의 주기적인 커밋 요청을 하게 되면, 호스트 입장에서 지나치게 빈번한 커밋이 발생하게 된다. 또한, 빈번하게 발생되는 커밋에 따른 게스트 머신의 개수가 많아질수록 쓰기 트래픽이 급격하게 증가하게 된다.

[0006] 가상 머신 환경에서 게스트 머신의 쓰기 요청에 따라 호스트 머신에서 발생되는 커밋에 의한 쓰기 트래픽을 최소화하는 기법이 필요하다.

발명의 내용

해결하려는 과제

[0007] 본 발명은 해시 테이블을 기반으로 복수의 게스트 머신 각각에서 요청된 수정 블록을 별도로 관리함으로써, 게스트 머신 간에 독립적으로 발생되는 커밋 요청을 고립화하는 방법 및 장치를 제공한다.

[0008] 본 발명은 가상 머신 환경에서 커밋을 요청한 게스트 머신의 수정 블록만을 호스트 머신의 스토리지에 커밋함으로써, 기존 호스트 머신에서 빈번하게 발생되는 커밋을 최소화하며, 커밋을 요청하지 않은 타 게스트 머신의 수정 블록에 의한 불필요한 스토리지의 커밋을 제거하는 방법 및 장치를 제공한다.

과제의 해결 수단

[0009] 본 발명의 일실시예에 따른 블록 커밋 방법은 호스트 머신의 커밋 주기(t)에 따른 복수의 게스트 머신(VM: Virtual Machine)으로부터 요청된 수정 블록을 포함하는 러닝 트랜잭션(Running Transaction)을 관리하는 단계; 복수의 게스트 머신에 의한 커밋 트리거가 발생하면, 커밋 트리거에 의한 수정 블록을 포함하는 러닝 트랜잭션을 커밋 트랜잭션으로 전환하는 단계; 및 전환된 커밋 트랜잭션에 포함되지 않은 나머지 수정 블록을 포함하는 러닝 트랜잭션을 유지하는 단계;를 포함할 수 있다.

[0010] 본 발명의 실시예에 따른 러닝 트랜잭션을 관리하는 단계는, 복수의 게스트 머신 각각으로부터 동기 쓰기(Sync Write)로 요청된 수정 블록이 연결 리스트로 연결된 러닝 트랜잭션을 관리할 수 있다.

[0011] 본 발명의 실시예에 따른 러닝 트랜잭션을 관리하는 단계는, 해시 테이블(Hash Table)에 따른 복수의 게스트 머신 각각으로부터 요청된 수정 블록의 아이노드(Inode)를 이용하여 복수의 게스트 머신 각각에 따라 구분된 수정 블록을 포함하는 러닝 트랜잭션으로 관리할 수 있다.

[0012] 본 발명의 실시예에 따른 커밋 트랜잭션으로 전환하는 단계는, 복수의 게스트 머신 중 어느 하나의 게스트 머신의 커밋 주기에 따른 커밋 트리거가 발생되면, 커밋 트리거가 발생한 게스트 머신이 요청한 수정 블록을 포함하는 러닝 트랜잭션을 커밋 트랜잭션으로 전환할 수 있다.

[0013] 본 발명의 실시예에 따른 커밋 트랜잭션으로 전환하는 단계는, 호스트 머신의 커밋 주기가 도래하는 경우, 러닝

트랜잭션에 포함된 모든 수정 블록을 커밋 트랜잭션의 목록으로 전환할 수 있다.

[0014] 본 발명의 실시예에 따른 러닝 트랜잭션을 유지하는 단계는, 커밋 트랜잭션에 포함되지 않은 나머지 수정 블록 및/또는 호스트의 커밋 주기(t+1)에서 복수의 게스트 머신으로부터 요청된 수정 블록을 연결 리스트로 연결하여 러닝 트랜잭션으로 유지할 수 있다.

[0015] 본 발명의 일실시예에 따른 프로세서를 포함하는 가상화 시스템에 있어서, 프로세서는, 호스트 머신의 커밋 주기(t)에 따른 복수의 게스트 머신으로부터 요청된 수정 블록을 포함하는 러닝 트랜잭션을 관리하고, 복수의 게스트 머신에 의한 커밋 트리거가 발생하면, 커밋 트리거에 의한 수정 블록을 포함하는 러닝 트랜잭션을 커밋 트랜잭션으로 전환하고, 전환된 커밋 트랜잭션에 포함되지 않은 나머지 수정 블록을 포함하는 러닝 트랜잭션을 유지할 수 있다.

[0016] 본 발명의 실시예에 따른 프로세서는, 복수의 게스트 머신 각각으로부터 동기 쓰기로 요청된 수정 블록이 연결 리스트로 연결된 러닝 트랜잭션을 관리할 수 있다.

[0017] 본 발명의 실시예에 따른 프로세서는, 해시 테이블에 따른 복수의 게스트 머신 각각으로부터 요청된 수정 블록의 아이노드를 이용하여 복수의 게스트 머신 각각에 따라 구분된 수정 블록을 포함하는 러닝 트랜잭션으로 관리할 수 있다.

[0018] 본 발명의 실시예에 따른 프로세서는, 복수의 게스트 머신 중 어느 하나의 게스트 머신의 커밋 주기에 따른 커밋 트리거가 발생되면, 커밋 트리거가 발생한 게스트 머신이 요청한 수정 블록을 포함하는 러닝 트랜잭션을 커밋 트랜잭션으로 전환할 수 있다.

[0019] 본 발명의 실시예에 따른 프로세서는, 호스트 머신의 커밋 주기가 도래하는 경우, 러닝 트랜잭션에 포함된 모든 수정 블록을 커밋 트랜잭션의 목록으로 전환할 수 있다.

[0020] 본 발명의 실시예에 따른 프로세서는, 커밋 트랜잭션에 포함되지 않은 나머지 수정 블록 및/또는 호스트의 커밋 주기(t+1)에서 복수의 게스트 머신으로부터 요청된 수정 블록을 연결 리스트로 연결하여 러닝 트랜잭션으로 유지할 수 있다.

발명의 효과

[0021] 본 발명의 일실시예에 따른 블록 커밋 방법은 해시 테이블을 기반으로 복수의 게스트 머신 각각에서 요청된 수정 블록을 별도로 관리함으로써, 게스트 머신 간에 독립적으로 발생되는 커밋 요청을 고립화한다.

[0022] 본 발명의 일실시예에 따른 블록 커밋 방법은 가상 머신 환경에서 커밋을 요청한 게스트 머신의 수정 블록만을 호스트 머신의 스토리지에 커밋함으로써, 기존 호스트 머신에서 빈번하게 발생되는 커밋을 최소화하며, 커밋을 요청하지 않은 타 게스트 머신의 수정 블록에 의한 불필요한 스토리지의 커밋을 제거한다.

도면의 간단한 설명

[0023] 도 1은 본 발명의 일실시예에 따른 게스트 머신과 호스트 머신 간 커밋을 수행하기 위한 가상화 시스템을 도시한 도면이다.

도 2는 본 발명의 일실시예에 따른 게스트 머신의 수정 블록에 관한 커밋 요청에 따른 트랜잭션을 분할하는 동작을 설명하기 위해 도시한 도면이다.

도 3은 본 발명의 일실시예에 따른 블록 커밋 방법을 설명하기 위한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0024] 실시예들에 대한 특정한 구조적 또는 기능적 설명들은 단지 예시를 위한 목적으로 개시된 것으로서, 다양한 형태로 변경되어 구현될 수 있다. 따라서, 실제 구현되는 형태는 개시된 특정 실시예로만 한정되는 것이 아니며, 본 명세서의 범위는 실시예들로 설명한 기술적 사상에 포함되는 변경, 균등물, 또는 대체물을 포함한다.

[0025] 제1 또는 제2 등의 용어를 다양한 구성요소들을 설명하는데 사용될 수 있지만, 이런 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 해석되어야 한다. 예를 들어, 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소는 제1 구성요소로도 명명될 수 있다.

[0026] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어

있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다.

[0027] 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 설명된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함으로 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0028] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 해당 기술분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 명세서에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0029] 이하, 실시예들을 첨부된 도면들을 참조하여 상세하게 설명한다. 첨부 도면을 참조하여 설명함에 있어, 도면부호에 관계없이 동일한 구성 요소는 동일한 참조 부호를 부여하고, 이에 대한 중복되는 설명은 생략하기로 한다.

[0031] 도 1은 본 발명의 일실시예에 따른 게스트 머신과 호스트 머신 간 커밋을 수행하기 위한 가상화 시스템을 도시한 도면이다.

[0032] 도 1을 참고하면, 가상화 시스템(100)은 가상 머신 환경에서의 커밋 요청에 따른 트랜잭션을 관리하기 위해 호스트 머신(101)과 복수의 게스트 머신(VM: Virtual Machine)을 포함할 수 있다. 보다 자세하게, 가상화 시스템(100)은 해시 테이블을 기반으로 복수의 게스트 머신 각각에서 요청된 수정 블록을 각각 구분하고, 커밋을 요청한 게스트 머신(102)의 수정 블록만을 호스트 머신(101)의 스토리지에 커밋하기 위해 트랜잭션을 분할하여 수정 블록을 커밋할 수 있다.

[0033] 가상화 시스템(100)은 커밋 연산의 구현을 위해 다음의 트랜젝션 리스트를 유지할 수 있으며, 일반적인 파일시스템/저널링 기법을 활용할 수 있다.

[0034] ① 러닝 트랜잭션(Running Transaction)은 직전 커밋 주기 이후 수정된 모든 데이터를 관리하는 트랜잭션이다.

[0035] ② 커밋 트랜잭션(Commit Transaction)은 호스트 머신 또는, 게스트 머신의 커밋 주기가 도래하는 경우, 상기 러닝 트랜잭션이 전환된 트랜잭션이다.

[0036] ③ 체크포인트 트랜잭션(Checkpoint Transaction)은 커밋 수행에 따른 커밋이 완료된 데이터를 관리하는 트랜잭션이다.

[0037] 상술한 각각의 트랜잭션은 다음의 상호 관계에 따라 활용될 수 있다. 가상화 시스템(100)은 호스트 머신의 커밋 주기가 도래하면, 직전 커밋 주기 이후 모든 수정 블록을 포함하는 러닝 트랜잭션을 커밋 트랜잭션으로 전환할 수 있다. 그리고, 트랜잭션 간 전환이 완료되면, 새로운 러닝 트랜잭션을 생성하고, 호스트 머신의 다음 커밋 주기가 도래할 때까지의 수정 블록들을 관리할 수 있다. 여기서, 커밋 작업은 커밋 트랜잭션에 속한 수정 블록들의 스토리지(104)의 저널 영역이라는 곳에 기록되는 것이며, 커밋 작업이 완료되면, 커밋이 완료된 블록을 체크포인트 트랜잭션을 통해 메모리에 관리할 수 있다.

[0038] 이때, 커밋은 스토리지(104)의 원래 위치가 아닌 저널 영역에 기록하는 과정이고, 추후 체크포인트 트랜잭션에 관한 커밋 주기가 도래하면, 체크포인트 트랜잭션에 속한 블록들을 파일 시스템의 원래 위치에 기록하는 체크포인팅 연산을 수행하게 된다.

[0039] 이에 따른, 가상화 시스템(100)은 호스트 머신(101)의 커밋 주기(t)에 따른 복수의 게스트 머신으로부터 요청된 수정 블록을 포함하는 러닝 트랜잭션을 관리할 수 있다. 호스트 머신(101)은 일정 시간을 간격으로 복수의 게스트 머신으로부터 요청된 수정 블록에 대한 커밋을 수행할 수 있다. 복수의 게스트 머신 각각은 사전에 정의된 커밋 주기를 갖고, 호스트 머신(101)으로 수정된 내용을 포함하는 수정 블록에 대한 동기 쓰기를 요청할 수 있다. 일례로, 가상화 시스템(100)의 저널링 파일 시스템에서 설정된 커밋 기간은 5 초로 설정되고, 복수의 게스트 머신 및 호스트 머신(101)은 자체적인 저널링 커밋을 수행할 수 있다.

[0040] 가상화 시스템(100)은 복수의 게스트 머신에 의한 커밋 트리거가 발생하면, 커밋 트리거에 의한 수정 블록을 포함하는 러닝 트랜잭션을 커밋 트랜잭션으로 전환할 수 있다. 가상화 시스템(100)은 전환된 커밋 트랜잭션에 포

함된 수정 블록에 한하여 스토리지의 커밋을 수행할 수 있다.

[0041] 자세하게, 저널링을 수행하기 위해 게스트 머신(101)은 호스트 머신(101)에 전송되는 최신 커밋 이후의 수정 블록을 포함하는 트랜잭션 목록을 유지할 수 있다. 그리고, 게스트 머신(102)은 호스트 머신(101)의 스토리지로의 동기 쓰기를 요청할 수 있다. 일례로, 게스트 머신(101)은 가상 머신 환경에서의 장벽 요청을 하이퍼 바이저에 전달하고, 하이퍼 바이저는 장병 요청에 따른 호스트 머신(102)에 대한 파일 동기화에 대한 호출을 수행할 수 있다.

[0042] 파일 동기화에 따른 호스트 머신(101)은 동기 쓰기가 요청된 각 게스트 머신을 위해 별도의 트랜잭션을 이용하여 수정 블록에 대한 커밋을 수행할 수 있다. 다시 말해, 게스트 머신 간에는 데이터의 의존성이 없으므로, 본 발명은 각 게스트 머신을 위해 별도의 트랜잭션을 관리함으로써, 기존 호스트 머신에서 빈번하게 발생되는 커밋을 최소화할 수 있다.

[0043] 또한, 별도의 트랜잭션을 이용해 커밋을 수행하는 것은 게스트 머신 간의 커밋 요청을 고립화하기 위함 일 수 있다. 즉, 호스트 머신은 게스트 머신으로부터 수정 블록에 관한 커밋 요청에 따른 커밋 트리거가 발생하면, 커밋 트리거가 발생된 게스트 머신에 속한 수정 블록들만을 커밋하고, 하며, 커밋을 요청하지 않은 타 게스트 머신의 수정 블록에 의한 불필요한 스토리지의 커밋을 제거할 수 있다.

[0044] 호스트 머신(101)은 전환된 커밋 트랜잭션에 포함되지 않은 나머지 수정 블록을 포함하는 러닝 트랜잭션을 유지할 수 있다. 또한, 호스트 머신(101)은 호스트 머신(101)의 커밋 주기가 도래하는 경우, 러닝 트랜잭션에 포함된 모든 수정 블록을 커밋 트랜잭션의 목록으로 전환함으로써, 모든 수정 블록에 대한 스토리지(104)의 커밋을 수행할 수 있다.

[0046] 도 2는 본 발명의 일실시예에 따른 게스트 머신의 수정 블록에 관한 커밋 요청에 따른 트랜잭션을 분할하는 동작을 설명하기 위해 도시한 도면이다.

[0047] 도 2의 (a)를 살펴보면, 가상화 시스템은 서도 다른 게스트 머신이 수정한 수정 블록들을 해쉬 테이블(204) 기반의 자료 구조를 활용하여 각 게스트 머신 별로 구분하여 관리할 수 있다. 다시 말해, 각 게스트 머신의 파일 시스템은 호스트 머신의 파일 시스템에서 별도의 파일로 인식될 수 있다. 본 발명은 서로 다른 게스트 머신에서 전달되는 수정 블록의 아이노드(inode)에 기반해서 각 게스트 머신 별 해쉬 테이블(204)의 저장 위치를 식별할 수 있다. 일례로, 각 게스트 머신의 파일 시스템들이 호스트 머신에서는 하나의 파일로 관리되므로, 서로 다른 게스트 머신은 파일의 inode를 통해 별도로 분리해서 관리하는 것이 가능할 수 있다.

[0048] 가상화 시스템은 서로 다른 게스트 머신의 수정 블록을 해시 테이블(204)의 서로 다른 저장 위치로 분리 및 저장할 수 있다. 가상화 시스템은 서로 다른 게스트 머신의 수정 블록에 대해서 연결 리스트로 연결하고, 이를 러닝 트랜잭션(201)으로 관리할 수 있다.

[0049] 도 2의 (b)를 살펴보면, 가상화 시스템은 게스트 머신(202)으로부터 동기 쓰기에 대한 요청을 수신할 수 있다. 가상화 시스템은 동기 쓰기에 대한 요청에 따른 해시 테이블(204)을 사용하여 게스트 머신(202)의 수정 블록(203)을 식별할 수 있다. 다시 말해, 가상화 시스템은 복수의 게스트 머신 중 어느 하나의 게스트 머신의 커밋 주기에 따른 커밋 트리거가 발생되면, 해시 테이블을 이용하여 수정 블록의 아이노드(Inode)에 따른 해시 테이블의 저장 위치를 판단할 수 있다.

[0050] 가상화 시스템은 판단된 해시 테이블의 저장 위치에 따른 수정 블록의 아이 노드를 저널 계층으로 전송한 후, 저널 데몬을 활성화할 수 있다. 활성화된 저널 데몬은 러닝 트랜잭션을 두 개의 하위 트랜잭션으로 분할할 수 있다. 다시 말해, 하나의 하위 트랜잭션은 커밋을 요청한 게스트 머신에 대응하여 해시 테이블의 저장 위치에 저장된 수정 블록을 포함할 수 있다. 그리고, 또 하나의 하위 트랜잭션은 커밋을 요청한 게스트 머신의 수정 블록을 제외한 나머지 수정 블록을 포함할 수 있다.

[0051] 그리고, 게스트 머신의 수정 블록을 포함하는 트랜잭션은 커밋을 수행하기 위한 커밋 트랜잭션으로 전환될 수 있다. 나머지 수정 블록을 포함하는 트랜잭션은 러닝 트랜잭션으로 유지될 수 있다. 일례로, 하나의 트랜잭션은 현재 요청 중인 게스트 머신의 수정 블록(도 2의 블록 D)으로 구성되고 다른 하나의 트랜잭션은 나머지 게스트 머신의 수정 블록(도 2의 블록 A, B, C, E, F, G, H)으로 구성될 수 있다. 도 2의 블록 D를 포함하는 트랜잭션은 커밋 트랜잭션의 역할을 수행하고, 도 2의 블록 A, B, C, E, F, G, H 를 포함하는 트랜잭션은 계속해서 실행중인 러닝 트랜잭션의 역할을 수행할 수 있다.

[0052] 또한, 가상화 시스템은 커밋이 호스트 머신 자체에 의해 트리거되면, 트랜잭션의 분할을 수행하지 않고, 러닝 트랜잭션 내의 모든 수정 블록을 커밋할 수 있다.

[0053] 결국, 게스트 머신으로부터 동기 쓰기에 대한 요청이 도착한 경우, 호스트 머신은 파일 시스템을 통해 저널링 프로세스를 활성화시킬 수 있다. 활성화된 저널링 프로세스는 러닝 트랜잭션을 2개의 서브 트랜잭션을 생성할 수 있다. 저널링 프로세서는 생성된 2개의 서브 트랜잭션에 대해 서밋 요청을 한 게스트 머신으로부터 발생한 수정 블록들을 포함하는 서브 트랜잭션과 그 외의 수정 블록들을 포함하는 서브 트랜잭션으로 구분할 수 있다.

[0054] 게스트 머신으로부터 발생한 수정 블록을 포함하는 서브 트랜잭션은 커밋 트랜잭션으로 전환되고, 그 외의 수정 블록을 포함하는 서브 트랜잭션은 러닝 트랜잭션으로 유지될 수 있다.

[0056] 도 3은 본 발명의 일실시예에 따른 블록 커밋 방법을 설명하기 위한 흐름도이다.

[0057] 단계(301)에서 가상화 시스템은 호스트 머신의 커밋 주기(t)에 따른 복수의 게스트 머신(VM)으로부터 요청된 수정 블록을 포함하는 러닝 트랜잭션을 관리할 수 있다. 가상화 시스템은 복수의 게스트 머신 각각으로부터 동기 쓰기(Cache Write)로 요청된 수정 블록이 연결 리스트로 연결된 러닝 트랜잭션을 관리할 수 있다.

[0058] 가상화 시스템은 해시 테이블(Hash Table)을 이용하여 복수의 게스트 머신 각각으로부터 요청된 수정 블록에 대해 서로 다른 저장 위치를 지정할 수 있다. 가상화 시스템은 서로 다른 저장 위치에 추가된 수정 블록의 아이노드를 이용하여 복수의 게스트 머신 각각에 따라 구분된 수정 블록을 포함하는 러닝 트랜잭션으로 관리할 수 있다.

[0059] 단계(302)에서 가상화 시스템은 복수의 게스트 머신에 의한 커밋 트리거가 발생하면, 커밋 트리거에 의한 수정 블록을 포함하는 러닝 트랜잭션을 커밋 트랜잭션으로 전환할 수 있다. 여기서, 가상화 시스템은 복수의 게스트 머신 중 어느 하나의 게스트 머신의 커밋 주기에 따른 커밋 트리거의 발생 여부를 감지할 수 있다. 가상화 시스템은 커밋 트리거를 감지하면, 커밋 트리거가 발생한 게스트 머신이 요청한 수정 블록을 포함하는 러닝 트랜잭션을 커밋 트랜잭션으로 전환할 수 있다.

[0060] 단계(303)에서 가상화 시스템은 전환된 커밋 트랜잭션에 포함되지 않은 나머지 수정 블록을 포함하는 러닝 트랜잭션을 유지할 수 있다. 가상화 시스템은 커밋 트랜잭션에 포함되지 않은 나머지 수정 블록 및/또는 호스트의 커밋 주기(t+1)에서 복수의 게스트 머신으로부터 요청된 수정 블록을 연결 리스트로 연결하여 러닝 트랜잭션으로 유지할 수 있다.

[0061] 또한, 가상화 시스템은 게스트 머신의 커밋 주기에 따라 발생된 커밋 트리거 이외에 호스트 머신의 커밋 주기가 도래하는 경우, 러닝 트랜잭션에 포함된 모든 수정 블록을 스토리지에 커밋할 수 있다. 결국, 가상화 시스템은 게스트 머신의 커밋 주기 및 호스트 머신의 커밋 주기에 따라 트랜잭션을 분할하여 수정 블록의 커밋을 관리할 수 있다.

[0063] 이상에서 설명된 실시예들은 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치, 방법 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 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 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0064] 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 장치, 또는 전송되는 신호 파(signal wave)에 영구적으로, 또는 일시적으로 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0065] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있으며 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.

[0066] 위에서 설명한 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 또는 복수의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0067] 이상과 같이 실시예들이 비록 한정된 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 이를 기초로 다양한 기술적 수정 및 변형을 적용할 수 있다. 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.

[0068] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

부호의 설명

[0069] 100: 가상화 시스템

101: 호스트 머신(Host Machine)

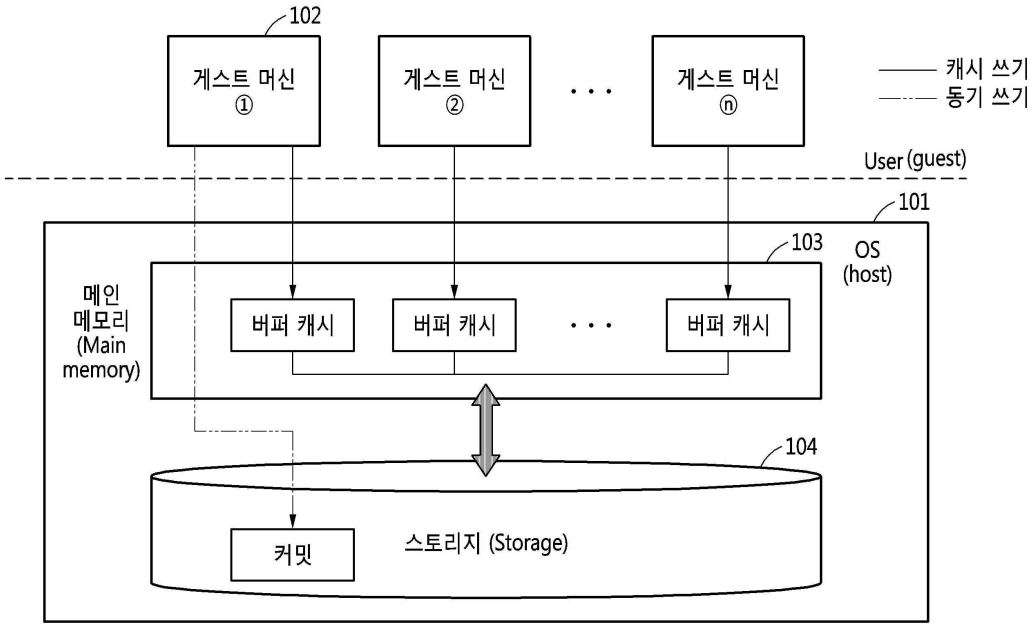
102: 게스트 머신(Guest Machine)

103: 메인 메모리(Main Memory)

104: 스토리지(Storage)

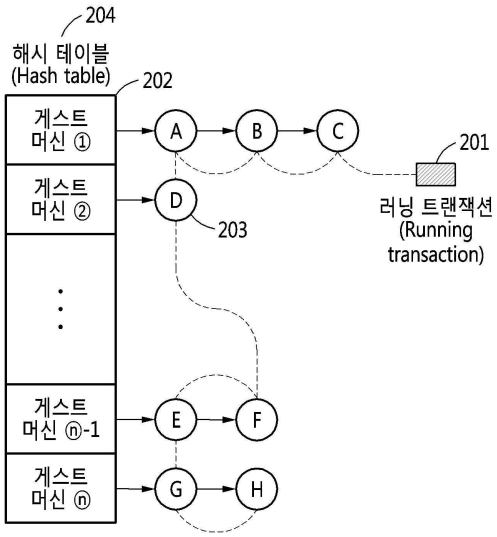
도면

도면1

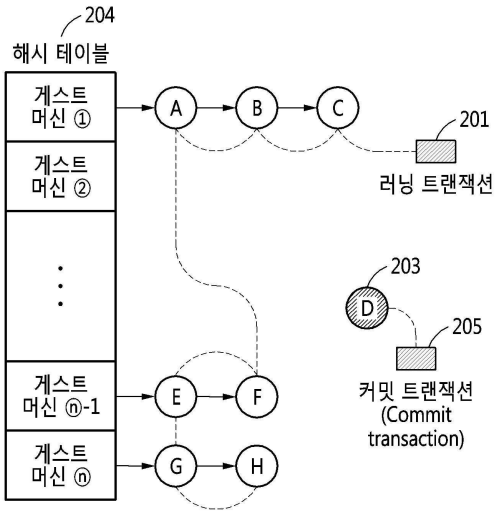


가상화 시스템 100

도면2

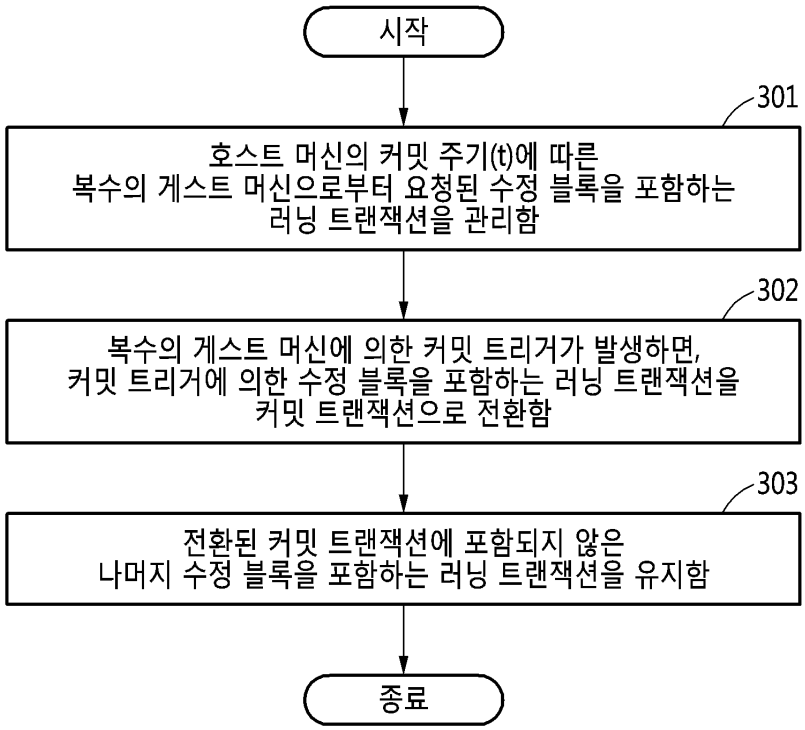


(a) 게스트 머신 ②의 커밋 전



(a) 게스트 머신 ②의 커밋 후

도면3


시작
301
호스트 머신의 커밋 주기(t)에 따른
복수의 게스트 머신으로부터 요청된 수정 블록을 포함하는
러닝 트랜잭션을 관리함
302
복수의 게스트 머신에 의한 커밋 트리거가 발생하면,
커밋 트리거에 의한 수정 블록을 포함하는 러닝 트랜잭션을
커밋 트랜잭션으로 전환함
303
전환된 커밋 트랜잭션에 포함되지 않은
나머지 수정 블록을 포함하는 러닝 트랜잭션을 유지함
종료