



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2013년09월03일
(11) 등록번호 10-1303079
(24) 등록일자 2013년08월28일

(51) 국제특허분류(Int. Cl.)
G06F 9/06 (2006.01) G06F 9/46 (2006.01)
G06F 9/44 (2006.01)
(21) 출원번호 10-2011-0042577
(22) 출원일자 2011년05월04일
심사청구일자 2011년05월04일
(65) 공개번호 10-2012-0124743
(43) 공개일자 2012년11월14일
(56) 선행기술조사문헌
KR1019990026156 A
KR1020060099404 A
KR1020060051177 A

(73) 특허권자
한국과학기술원
대전광역시 유성구 대학로 291(구성동)
(72) 발명자
허재혁
대전광역시 유성구 대학로 291, 한국과학기술원 (구성동)
김대훈
서울특별시 은평구 연서로44길 7, 411동 901호 (진관동, 은평뉴타운 폭포동)
김환주
광주광역시 북구 서암대로 261-6 (충흥동)
(74) 대리인
특허법인충현

전체 청구항 수 : 총 17 항

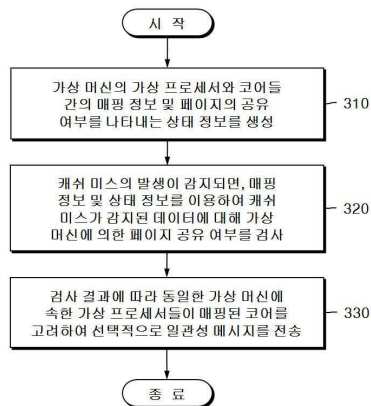
심사관 : 복진요

(54) 발명의 명칭 멀티-코어 기반의 가상화 환경에서 캐쉬 일관성을 제어하는 장치 및 방법

(57) 요약

본 발명은 멀티-코어 기반의 가상화 환경에서 캐쉬 일관성을 제어하는 장치, 방법 및 그 방법을 기록한 기록매체에 관한 것으로서, 본 발명에 따른 캐쉬 일관성 제어 방법은, 가상 머신의 가상 프로세서와 상기 코어들 간의 매핑 정보 및 페이지의 공유 여부를 나타내는 상태 정보를 생성하고, 캐쉬 미스의 발생이 감지되면 매핑 정보 및 상태 정보를 이용하여 캐쉬 미스가 감지된 데이터에 대해 가상 머신에 의한 페이지 공유 여부를 검사하며, 검사 결과에 따라 동일한 가상 머신에 속한 가상 프로세서들이 매핑된 코어를 고려하여 선택적으로 일관성 메시지를 전송한다.

대표도 - 도3



특허청구의 범위

청구항 1

복수 개의 가상 머신이 복수 개의 코어(core)를 공유하는 가상화 환경에서 캐쉬 일관성(cache coherence)을 제어하는 방법에 있어서,

상기 가상 머신의 가상 프로세서와 상기 코어들 간의 매핑(mapping) 정보 및 페이지의 공유 여부를 나타내는 상태 정보를 생성하는 단계;

캐쉬 미스의 발생이 감지되면, 상기 생성된 매핑 정보 및 상기 생성된 상태 정보를 이용하여 상기 캐쉬 미스가 감지된 데이터에 대해 상기 가상 머신에 의한 페이지(page) 공유 여부를 검사하는 단계; 및

상기 검사 결과에 따라 동일한 가상 머신에 속한 가상 프로세서들이 매핑된 코어를 고려하여 선택적으로 일관성 메시지를 전송하는 단계를 포함하되,

상기 일관성 메시지를 전송하는 단계는,

상기 캐쉬 미스의 발생을 감지하는 단계;

상기 생성된 상태 정보를 이용하여 상기 캐쉬 미스가 감지된 데이터에 대해 상기 가상 머신에 의한 페이지 공유 여부를 검사하는 단계;

상기 검사 결과에 기초하여 가상 머신 모니터(virtual machine monitor)가 상기 생성된 매핑 정보를 참조하여 상기 가상 머신에 매핑된 코어를 선택하는 단계; 및

상기 선택된 코어에 일관성 메시지를 전송하는 단계를 포함하는 방법.

청구항 2

삭제

청구항 3

제 1 항에 있어서,

상기 상태 정보는,

상기 페이지가 복수 개의 가상 머신에 의해 공유되지 않음을 나타내는 가상 머신 독점(virtual machine private) 상태,

상기 페이지가 복수 개의 가상 머신에 의해 공유되고 상기 가상 머신들 각각이 공유 페이지에 대해 읽기 및 쓰기가 가능함을 나타내는 읽기-쓰기 공유(read-write shared) 상태 및

상기 페이지가 복수 개의 가상 머신에 의해 공유되고 상기 가상 머신들 각각이 공유 페이지에 대해 읽기만이 가능함을 나타내는 읽기-전용 공유(read-only shared) 상태 중 어느 하나인 것을 특징으로 하는 방법.

청구항 4

제 1 항에 있어서,

상기 검사 결과, 상기 페이지가 복수 개의 가상 머신에 의해 공유되지 않음을 나타내는 가상 머신 독점 상태인 경우,

상기 일관성 메시지를 전송하는 단계는 상기 매핑 정보를 저장하는 가상 프로세서 맵 레지스터(virtual processor map register)를 참조함으로써 독출된 스누프 도메인(snoop domain)에 포함된 코어들에게만 일관성 메시지를 전송하는 것을 특징으로 하는 방법.

청구항 5

제 1 항에 있어서,

상기 검사 결과, 상기 페이지가 복수 개의 가상 머신에 의해 공유되고 상기 가상 머신들 각각이 공유 페이지에 대해 읽기 및 쓰기가 가능함을 나타내는 읽기-쓰기 공유 상태인 경우,

상기 일관성 메시지를 전송하는 단계는 모든 코어들에게 일관성 메시지를 브로드캐스팅(broadcasting)하는 것을 특징으로 하는 방법.

청구항 6

제 1 항에 있어서,

상기 검사 결과, 상기 페이지가 복수 개의 가상 머신에 의해 공유되고 상기 가상 머신들 각각이 공유 페이지에 대해 읽기만이 가능함을 나타내는 읽기-전용 공유 상태인 경우,

상기 일관성 메시지를 전송하는 단계는, 브로드캐스팅 방법, 메모리 직접 방법(memory-direct method), 내부 가상 머신 방법(intra-VM method), 프렌드 가상 머신 방법(friend-VM method) 중 적어도 하나를 이용하여 수행되는 것을 특징으로 하는 방법.

청구항 7

제 6 항에 있어서,

상기 메모리 직접 방법은, 상기 캐쉬 미스된 데이터를 메모리로부터 직접 독출하는 것을 특징으로 하는 방법.

청구항 8

제 6 항에 있어서,

상기 내부 가상 머신 방법은, 상기 매핑 정보를 이용하여 상기 캐쉬 미스가 발생한 가상 머신에 매핑된 코어 및 메모리에 동시에 캐쉬 미스된 데이터를 요청하는 것을 특징으로 하는 방법.

청구항 9

제 6 항에 있어서,

상기 프렌드 가상 머신 방법은,

상기 복수 개의 가상 머신 별로 공유가 빈번하게 발생하는 가상 머신들을 프렌드 가상 머신으로 미리 설정하고, 상기 매핑 정보를 이용하여 상기 캐쉬 미스가 발생한 가상 머신에 매핑된 코어, 상기 설정된 프렌드 가상 머신에 매핑된 코어 및 메모리에 동시에 캐쉬 미스된 데이터를 요청하는 것을 특징으로 하는 방법.

청구항 10

복수 개의 가상 머신이 복수 개의 코어를 공유하는 가상화 환경에서 캐쉬 일관성을 제어하는 방법에 있어서,

상기 가상 머신의 가상 프로세서와 상기 코어들 간의 매핑 정보를 생성하여 가상 프로세서 맵 레지스터에 저장하는 단계;

페이지의 공유 여부를 나타내는 상태 정보를 생성하여 새도우 페이지 테이블(shadow page table)에 저장하는 단계; 및

캐쉬 미스의 발생이 감지되면, 상기 저장된 매핑 정보 및 상기 저장된 상태 정보를 이용하여 상기 캐쉬 미스가 감지된 데이터에 대해 상기 가상 머신에 의한 페이지 공유 여부를 검사함으로써 상기 복수 개의 코어 중에서 선택된 코어에만 일관성 메시지를 전송하는 단계를 포함하는 방법.

청구항 11

제 10 항에 있어서,

상기 복수 개의 코어 별로 해당 코어의 로컬 캐쉬(local cache)에 적재된 가상 머신의 데이터의 수를 나타내는 레지던스 카운터(residence counter)를 산출하여 저장하는 단계를 더 포함하는 방법.

청구항 12

제 11 항에 있어서,

소정 코어에 대응하는 레지던스 카운터가 0인 경우, 상기 저장된 매핑 정보의 스놉 도메인에서 상기 소정 코어를 삭제하는 것을 특징으로 하는 방법.

청구항 13

제 1 항, 제 3 항 내지 제 12 항 중에 어느 한 항의 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

청구항 14

복수 개의 가상 머신이 복수 개의 코어를 공유하는 가상화 환경에서 캐쉬 일관성을 제어하는 장치에 있어서,

상기 가상 머신의 가상 프로세서와 상기 코어들 간의 매핑 정보를 생성하여 가상 프로세서 맵 레지스터에 저장하고, 페이지의 공유 여부를 나타내는 상태 정보를 생성하여 새도우 페이지 테이블에 저장하는 모니터링(monitoring)부;

캐쉬 미스의 발생이 감지되면, 상기 저장된 매핑 정보 및 상기 저장된 상태 정보를 이용하여 상기 캐쉬 미스가 감지된 데이터에 대해 상기 가상 머신에 의한 페이지 공유 여부를 검사하는 공유 검사부; 및

상기 검사 결과에 따라 동일한 가상 머신에 속한 가상 프로세서들이 매핑된 코어를 고려하여 선택적으로 일관성 메시지를 전송하는 처리부를 포함하되,

상기 처리부는,

상기 캐쉬 미스의 발생을 감지하고,

상기 생성된 상태 정보를 이용하여 상기 캐쉬 미스가 감지된 데이터에 대해 상기 가상 머신에 의한 페이지 공유 여부를 검사하고,

상기 검사 결과에 기초하여 가상 머신 모니터(virtual machine monitor)가 상기 생성된 매핑 정보를 참조하여 상기 가상 머신에 매핑된 코어를 선택하고,

상기 선택된 코어에 일관성 메시지를 전송하는 것을 특징으로 하는 장치.

청구항 15

제 14 항에 있어서,

상기 상태 정보는, 상기 페이지가 복수 개의 가상 머신에 의해 공유되지 않음을 나타내는 가상 머신 독점 상태, 상기 페이지가 복수 개의 가상 머신에 의해 공유되고 상기 가상 머신들 각각이 공유 페이지에 대해 읽기 및 쓰기가 가능함을 나타내는 읽기-쓰기 공유 상태 및 상기 페이지가 복수 개의 가상 머신에 의해 공유되고 상기 가상 머신들 각각이 공유 페이지에 대해 읽기만이 가능함을 나타내는 읽기-전용 공유 상태 중 어느 하나이고,

상기 모니터링부는 상기 페이지와 상기 페이지의 상태 정보를 매핑하여 상기 새도우 페이지 테이블에 저장하고,

상기 공유 검사부는 데이터 접근시 상기 새도우 페이지 테이블 및 변환 색인 버퍼(translation lookaside buffer, TLB)를 참조하여 상기 페이지 공유 여부를 검사하는 것을 특징으로 하는 장치.

청구항 16

복수 개의 가상 머신이 복수 개의 코어를 공유하는 가상화 환경에서 캐쉬 일관성을 제어하는 장치에 있어서,

상기 가상 머신의 가상 프로세서와 상기 코어들 간의 매핑 정보를 생성하여 가상 프로세서 맵 레지스터에 저장하고, 페이지의 공유 여부를 나타내는 상태 정보를 생성하여 새도우 페이지 테이블에 저장하며, 상기 복수 개의 코어 별로 해당 코어의 로컬 캐쉬에 적재된 가상 머신의 데이터의 수를 나타내는 레지던스 카운터를 산출하여 저장하는 모니터링부; 및

캐쉬 미스의 발생이 감지되면, 상기 저장된 매핑 정보 및 상기 저장된 상태 정보를 이용하여 상기 캐쉬 미스가 감지된 데이터에 대해 상기 가상 머신에 의한 페이지 공유 여부를 검사함으로써 상기 복수 개의 코어 중에서 선택된 코어에만 일관성 메시지를 전송하는 처리부를 포함하는 장치.

청구항 17

제 16 항에 있어서,

상기 모니터링부는 상기 복수 개의 가상 머신 별로 할당된 가상 프로세서를 매핑시켜 스눅 도메인으로서 상기 가상 프로세서 맵 레지스터에 저장하는 것을 특징으로 하는 장치.

청구항 18

제 16 항에 있어서,

소정 코어에 대응하는 레지던스 카운터가 0인 경우, 상기 모니터링부는 상기 저장된 매핑 정보의 스눅 도메인에서 상기 소정 코어를 삭제하는 것을 특징으로 하는 장치.

명세서

기술분야

[0001] 본 발명은 가상화 환경이 구현된 멀티-코어 시스템에 관한 것으로, 보다 상세하게는 복수 개의 코어들이 구비된 시스템에서 가상화 환경의 특성을 고려하여 캐쉬 일관성을 제어함으로써 코어들 간의 불필요한 일관성 메시지 전송에 따른 통신량을 감소시킬 수 있는 캐쉬 일관성 제어 장치, 방법 및 그 방법을 기록한 기록매체에 관한 것이다.

배경기술

[0002] 가상화(virtualization)는 컴퓨터의 자원을 추상화(abstraction)해 실제 물리적인 자원과 분리시키는 기술이다. 종래에는 중앙처리장치(CPU)나 메모리 같은 물리적인 컴퓨팅 자원은 운영체제(operating system: OS)와 연결되어 하나의 컴퓨팅 시스템을 구성하였다. 즉, 하나의 하드웨어 세트(set)에는 하나의 컴퓨팅 시스템만이 구성 가능하였다. 그러나, 가상화는 이러한 종래의 개념을 깨고 하나의 물리적인 하드웨어 상에 다수의 운영체제 및 이에 기반한 컴퓨팅 시스템을 구축하는 것을 가능하게 한다. 특히, 다양한 가상화 기술 중 서버 가상화 기술은 사용자나 기업의 입장에서는 하나의 PC나 서버를 가지고서도 여러 대의 PC를 가지고 있는 것처럼 사용할 수 있게 해 준다.

[0003] 한편, 이미 오래전부터 단일 스레드(thread) 기반의 싱글-코어(single-core) 기술은 과도한 전력 소모로 인하여 성능 향상의 한계에 도달하였다. 이를 극복하기 위하여 하나의 중앙처리장치(CPU)에 다수의 코어를 탑재한 멀티-코어(multi-core) 시스템 구축에 관한 연구가 활발하게 진행되어 왔다. 현재 대부분의 대용량 데이터 처리 시스템에서는 멀티-코어 시스템을 채택하여 사용 중에 있으며, 이로 인해 하나의 코어를 사용할 때에 비해 상대적으로 우수한 성능 향상을 이루었다. 특히, 이러한 멀티-코어 시스템은 병렬화된 작업을 처리할 때에 더욱 큰 이점과 성능 향상을 보여준다. 또한, 향후 반도체 직접도 향상에 따른 중앙처리장치당 코어의 수는 계속 증가할 것으로 예상된다.

[0004] 멀티-코어 시스템에는 각 코어 별로 보다 빠른 데이터 접근을 위한 로컬 캐쉬(local cache)가 존재하고, 이러한 코어들이 동일한 공유 메모리를 사용한다면, 캐쉬들 간의 일관성을 맞춰 주기 위해 코어들 간의 통신이 필요함을 알 수 있다. 예를 들어, 제 1 코어가 변수 x를 자신의 로컬 캐쉬에 적재(load)하였고, 제 2 코어 역시 변수 x를 자신의 로컬 캐쉬에 적재한 상황을 가정하자. 만약, 제 2 코어가 변수 x의 값을 변경하게 되면 쓰기에 의한 캐쉬 미스(cache miss)가 발생하게 되고, 동일한 변수 x에 대해 제 1 코어가 인지하는 값과 제 2 코어가 인지하는 값이 달라지는 문제가 발생한다. 이러한 상황을 캐쉬 일관성(cache coherence) 문제라고 하며, 이를 해결하기 위한 다양한 기술들이 소개되어 있다.

발명의 내용

해결하려는 과제

[0005] 본 발명이 해결하고자 하는 기술적 과제는 종래의 가상화 기술이 싱글-코어 기반에서 벗어나지 못하는 한계를 극복하고, 멀티-코어에 기반한 가상화 환경 하에서 캐쉬 일관성을 유지하기 위해 전송하는 일관성 메시지로 인해 불필요한 통신이 발생하는 불편함을 해소하며, 이러한 불필요한 통신으로 인해 시스템의 성능이 하락하고 전

력 소모가 증가하는 문제점을 해결할 수 있는 캐쉬 일관성 제어 장치 및 방법을 제공하고자 한다.

과제의 해결 수단

- [0006] 상기 기술적 과제를 해결하기 위하여, 본 발명에 따른 복수 개의 가상 머신이 복수 개의 코어를 공유하는 가상화 환경에서 캐쉬 일관성을 제어하는 방법은, 상기 가상 머신의 가상 프로세서와 상기 코어들 간의 매핑(mapping) 정보 및 페이지의 공유 여부를 나타내는 상태 정보를 생성하는 단계; 캐쉬 미스의 발생이 감지되면, 상기 생성된 매핑 정보 및 상기 생성된 상태 정보를 이용하여 상기 캐쉬 미스가 감지된 데이터에 대해 상기 가상 머신에 의한 페이지(page) 공유 여부를 검사하는 단계; 및 상기 검사 결과에 따라 동일한 가상 머신에 속한 가상 프로세서들이 매핑된 코어를 고려하여 선택적으로 일관성 메시지를 전송하는 단계를 포함한다.
- [0007] 이 때, 상기 상태 정보는, 상기 페이지가 복수 개의 가상 머신에 의해 공유되지 않음을 나타내는 가상 머신 독립점(virtual machine private) 상태, 상기 페이지가 복수 개의 가상 머신에 의해 공유되고 상기 가상 머신들 각각이 공유 페이지에 대해 읽기 및 쓰기가 가능함을 나타내는 읽기-쓰기 공유(read-write shared) 상태 및 상기 페이지가 복수 개의 가상 머신에 의해 공유되고 상기 가상 머신들 각각이 공유 페이지에 대해 읽기만이 가능함을 나타내는 읽기-전용 공유(read-only shared) 상태 중 어느 하나이다.
- [0008] 상기 기술적 과제를 해결하기 위하여, 본 발명에 따른 복수 개의 가상 머신이 복수 개의 코어를 공유하는 가상화 환경에서 캐쉬 일관성을 제어하는 방법은, 상기 가상 머신의 가상 프로세서와 상기 코어들 간의 매핑 정보를 생성하여 가상 프로세서 맵 레지스터에 저장하는 단계; 페이지의 공유 여부를 나타내는 상태 정보를 생성하여 새도우 페이지 테이블(shadow page table)에 저장하는 단계; 및 캐쉬 미스의 발생이 감지되면, 상기 저장된 매핑 정보 및 상기 저장된 상태 정보를 이용하여 상기 캐쉬 미스가 감지된 데이터에 대해 상기 가상 머신에 의한 페이지 공유 여부를 검사함으로써 상기 복수 개의 코어 중에서 선택된 코어에만 일관성 메시지를 전송하는 단계를 포함한다.
- [0009] 또한, 상기된 캐쉬 일관성을 제어하는 방법은, 상기 복수 개의 코어 별로 해당 코어의 로컬 캐쉬(local cache)에 적재된 가상 머신의 데이터의 수를 나타내는 레지던스 카운터(residence counter)를 산출하여 저장하는 단계를 더 포함한다.
- [0010] 나아가, 이하에서는 상기 기재된 복수 개의 가상 머신이 복수 개의 코어를 공유하는 가상화 환경에서 캐쉬 일관성을 제어하는 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공한다.
- [0011] 상기 기술적 과제를 해결하기 위하여, 본 발명에 따른 복수 개의 가상 머신이 복수 개의 코어를 공유하는 가상화 환경에서 캐쉬 일관성을 제어하는 장치는, 상기 가상 머신의 가상 프로세서와 상기 코어들 간의 매핑 정보를 생성하여 가상 프로세서 맵 레지스터에 저장하고, 페이지의 공유 여부를 나타내는 상태 정보를 생성하여 새도우 페이지 테이블에 저장하는 모니터링(monitring)부; 캐쉬 미스의 발생이 감지되면, 상기 저장된 매핑 정보 및 상기 저장된 상태 정보를 이용하여 상기 캐쉬 미스가 감지된 데이터에 대해 상기 가상 머신에 의한 페이지 공유 여부를 검사하는 공유 검사부; 및 상기 검사 결과에 따라 동일한 가상 머신에 속한 가상 프로세서들이 매핑된 코어를 고려하여 선택적으로 일관성 메시지를 전송하는 처리부를 포함한다.
- [0012] 이 때, 상기 상태 정보는, 상기 페이지가 복수 개의 가상 머신에 의해 공유되지 않음을 나타내는 가상 머신 독립점 상태, 상기 페이지가 복수 개의 가상 머신에 의해 공유되고 상기 가상 머신들 각각이 공유 페이지에 대해 읽기 및 쓰기가 가능함을 나타내는 읽기-쓰기 공유 상태 및 상기 페이지가 복수 개의 가상 머신에 의해 공유되고 상기 가상 머신들 각각이 공유 페이지에 대해 읽기만이 가능함을 나타내는 읽기-전용 공유 상태 중 어느 하나이고, 상기 모니터링부는 상기 페이지와 상기 페이지의 상태 정보를 매핑하여 상기 새도우 페이지 테이블에 저장하고, 상기 공유 검사부는 데이터 접근시 상기 새도우 페이지 테이블 및 변환 색인 버퍼(translation lookaside buffer, TLB)를 참조하여 상기 페이지 공유 여부를 검사한다.
- [0013] 상기 기술적 과제를 해결하기 위하여, 본 발명에 따른 복수 개의 가상 머신이 복수 개의 코어를 공유하는 가상화 환경에서 캐쉬 일관성을 제어하는 장치는, 상기 가상 머신의 가상 프로세서와 상기 코어들 간의 매핑 정보를 생성하여 가상 프로세서 맵 레지스터에 저장하고, 페이지의 공유 여부를 나타내는 상태 정보를 생성하여 새도우 페이지 테이블에 저장하며, 상기 복수 개의 코어 별로 해당 코어의 로컬 캐쉬에 적재된 가상 머신의 데이터의 수를 나타내는 레지던스 카운터를 산출하여 저장하는 모니터링부; 및 캐쉬 미스의 발생이 감지되면, 상기 저장된 매핑 정보 및 상기 저장된 상태 정보를 이용하여 상기 캐쉬 미스가 감지된 데이터에 대해 상기 가상 머신에 의한 페이지 공유 여부를 검사함으로써 상기 복수 개의 코어 중에서 선택된 코어에만 일관성 메시지를 전송하는

처리부를 포함한다.

[0014] 또한, 상기 기재된 캐쉬 일관성 제어 장치에서, 상기 모니터링부는 상기 복수 개의 가상 머신 별로 할당된 가상 프로세서를 매핑시켜 스누프 도메인으로서 상기 가상 프로세서 맵 레지스터에 저장한다.

발명의 효과

[0015] 본 발명은 가상 프로세서와 코어들 간의 매핑 정보 및 페이지의 공유 여부를 나타내는 상태 정보를 이용하여 코어들에 선택적으로 일관성 메시지를 전송하는 기술을 제안함으로써 가상화 기술을 멀티-코어 기반의 시스템에 확장하여 적용할 수 있고, 멀티-코어에 기반한 가상화 환경 하에서 캐쉬 일관성을 유지하기 위해 전송하는 일관성 메시지에 따른 통신량을 감소시킴으로써 시스템의 성능을 향상시키며, 이러한 통신량 감소로 인해 전력 소모를 최소화할 수 있다.

도면의 간단한 설명

[0016] 도 1은 복수 개의 가상 머신이 복수 개의 코어를 공유하는 가상화 환경에서 캐쉬 일관성을 제어하기 위한 본 발명의 기본 아이디어를 도시한 도면이다.

도 2는 본 발명의 일 실시예에 따른 가상화 환경에서 가상 머신의 재배치가 이루어지는 과정을 예시한 도면이다.

도 3은 본 발명의 일 실시예에 따른 가상화 환경에서 캐쉬 일관성을 제어하는 방법을 도시한 흐름도이다.

도 4는 본 발명의 일 실시예에 따른 도 3의 캐쉬 일관성 제어 방법을 보다 구체적으로 도시한 흐름도이다.

도 5는 본 발명의 일 실시예에 따른 가상화 환경에서 캐쉬 미스가 발생한 페이지가 읽기-전용 공유 상태인 경우 캐쉬 일관성을 제어하는 방법의 일례를 도시한 도면이다.

도 6은 본 발명의 다른 실시예에 따라 레지던스 카운터를 이용하여 가상 프로세서 맵 레지스터의 스누프 도메인을 갱신하는 과정을 예시한 도면이다.

도 7은 본 발명의 일 실시예에 따른 가상화 환경에서 캐쉬 일관성을 제어하는 장치를 도시한 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0017] 본 발명의 실시예들을 설명하기에 앞서, 캐쉬 일관성을 제어하는 관련 기술과 본 발명의 실시예들이 구현되는 가상화 환경에서 복수 개의 가상 머신이 복수 개의 코어를 공유할 때의 문제 상황에 대해 개괄적으로 검토하도록 하겠다.

[0018] 일반적으로 캐쉬 일관성(cache coherence) 문제를 해결하기 위해서는 모든 로컬 캐쉬가 항상 최신의 값을 반환해줄 수 있도록 프로토콜에 의해 캐쉬 간의 일관성을 유지해야 한다. 또한, 로컬 캐쉬에서 읽기에 의한 캐쉬 미스가 발생했을 때 역시 해당 데이터가 만약 다른 코어의 로컬 캐쉬에 존재할 경우, 해당 데이터를 지연시간(latency time)이 큰 램(RAM)과 같은 메모리에서 가져오지 않고, 다른 코어의 로컬 캐쉬에서 데이터를 가져오는 것이 바람직하다. 이러한 처리를 위해서도 캐쉬 일관성 프로토콜이 사용된다. 캐쉬 일관성을 위한 기본적인 두 가지 방법으로 스누프(snoop) 방식과 디렉토리(directory) 방식이 알려져 있다.

[0019] 첫째, 스누프 방식은 캐쉬 일관성 문제를 해결하기 위해 캐쉬 미스가 발생한 코어에서 요청하는 메시지를 모든 코어에 직접적으로 전송하는 방식이다. 이러한 스누프 방식은 별도의 지연 없이 빠른 캐쉬 간의 데이터 전달을 제공하지만, 만약 코어의 수가 많아진다면 모든 코어로 메시지를 보내야 하기 때문에 통신량이 급격하게 늘어나는 문제점을 갖는다.

[0020] 둘째, 디렉토리 방식은 별도의 디렉토리를 가지고 해당 데이터가 어느 코어의 로컬 캐쉬에 적재(load)가 되어 있는지를 추적하여 해당 코어에만 메시지를 보내는 방식이다. 이를 위해 각각의 데이터마다 물리적 주소에 따라 다른 홈 노드(home node)라고 불리우는 코어가 존재하고, 홈 노드는 해당 데이터가 어느 코어의 로컬 캐쉬에 로드되어 있는지에 대한 정보를 유지하고 제공하게 된다. 따라서, 캐쉬 미스가 발생하게 되면 우선 캐쉬 미스가 발생한 코어에서 홈 노드로 메시지를 발생시키고, 홈 노드에서 제공하는 공유 코어들에 대한 정보를 바탕으로 해당 코어들에게만 메시지를 보내게 된다. 이상과 같은 과정을 통해서 디렉토리 방식은 통신량은 줄일 수 있지만 홈 노드로 가기 위한 추가적인 과정이 필요하고, 공유 코어들에 대한 정보를 유지하기 위한 별도의 공간과 과정이 필요하게 되므로 복잡도가 높아지고 비용이 증가하게 된다.

- [0021] 이러한 복잡성과 비용 때문에 현재 많은 시스템에서는 스눕 방식을 채택하여 사용하여 왔으나, 문제는 멀티-코어의 개수가 증가하는 경우에 발생할 수 있다. 스눕 방식의 경우 4개 정도의 코어를 갖는 멀티-코어 시스템에서는 일정 수준의 성능을 보장할 수 있으나, 멀티-코어의 개수가 8, 16, 32, 64, 128 개와 같이 증가할 경우, 일관성 메시지를 전송으로 인해 통신량이 급격하게 증가하게 된다. 이러한 통신량의 증가는 시스템의 성능 저하와 과도한 전력 소모를 야기할 수 밖에 없으므로, 이러한 불필요한 통신량을 감소시키기 위한 노력이 필요하다.
- [0022] 한편, 최근 클라우드 컴퓨팅(cloud computing)이 이슈가 되면서 하나의 물리적 머신 상에서 다수의 가상 머신(virtual machine)을 구현함으로써, 각각의 가상 머신을 통해 마치 별개의 하드웨어 자원이 존재하는 것처럼 동작하게 해주는 가상화(virtualization) 기술이 주목받고 있다. 이러한 가상화 기술은 높은 자원 이용률(resource utilization), 높은 결함 내성(fault tolerance)과 이식성(portability), 그리고 뛰어난 관리성(manageability)의 장점을 갖는다.
- [0023] 그런데, 이러한 가상화 기술을 멀티-코어 시스템 환경에 적용시키게 되면, 물리적인 머신(core를 의미한다.)이 복수 개 존재할 뿐만 아니라, 가상 머신 역시 복수 개 존재하는 상황에 놓여진다. 즉, 각각의 가상 머신은 가상 프로세서(virtual processor, vCPU)를 가지고 있고, 또한 가상 프로세서들은 실제 복수 개의 물리적 코어에 매핑(mapping)이 되어 코어를 이용하게 된다. 이러한 환경 하에서 가상 머신들은 각각 서로 다른 주소 메모리 영역을 사용하기 때문에, 가상 머신 간의 데이터 공유나 가상 머신 모니터 영역 같은 부분을 제외하고는 가상 머신들 간의 통신은 불필요하다. 그러나, 현재의 멀티-코어 시스템에서는 코어 간의 통신에 있어서 가상 머신에 대한 고려가 전혀 없기 때문에 상당량의 불필요한 통신이 발생하게 된다. 따라서, 멀티-코어 시스템에서 가상화 환경을 고려하여 이에 적합한 캐쉬 일관성 유지를 위한 통신량 감소 기법이 요구된다.
- [0024] 이하에서, 관련 도면들을 참조하여 본 발명의 실시예들을 보다 구체적으로 설명한다. 도면들에서 동일한 참조 번호들은 동일한 구성 요소를 지칭한다.
- [0025] 도 1은 복수 개의 가상 머신이 복수 개의 코어를 공유하는 가상화 환경에서 캐쉬 일관성을 제어하기 위한 본 발명의 기본 아이디어를 도시한 도면이다. 캐쉬 일관성 제어 장치(100)는 복수 개의 코어(core)(50)와 이에 따른 캐쉬(L2 캐쉬가 예시되었다.)뿐만 아니라, 가상 프로세서 맵 레지스터(virtual processor map register)(13)와 섀도우 페이지 테이블(shadow page table)(15)을 포함한다.
- [0026] 가상 프로세서 맵 레지스터(13)는 가상 머신과 이에 할당된 코어를 매핑시켜 관리한다. 즉, 가상 프로세서 맵 레지스터(13)는 가상 머신 식별자(VMID)와 코어 리스트를 연결하여 관리하는 것이 바람직하다. 도 1에는 1개의 가상 머신(VM1)에 4개의 코어(P0, P1, P2, P4)가 매핑되어 있음이 예시되어 있다.
- [0027] 섀도우 페이지 테이블(15)은 페이지와 각각의 페이지의 상태를 매핑시켜 관리한다. 즉, 섀도우 페이지 테이블(15)에는 페이지 주소(paddr)와 페이지의 상태(status)를 연결하여 테이블의 형태로 관리하는 것이 바람직하다. 도 1에는 3개의 페이지에 대해 각각 다른 3개의 상태(VM-private, RW-shared, RO-shared)가 기록되어 있음이 예시되어 있다. 각각의 상태 정보에 대해서는 이후 도 3을 통해 구체적으로 설명한다.
- [0028] 가상화 기술은 각 가상 머신 별로 독립적인 컴퓨팅 환경을 제공한다. 기본적으로 각 가상 머신은 개별적인 가상 프로세서와 메모리 영역을 사용하게 된다. 각 가상 머신의 가상 프로세서(vCPU)가 서로 다른 물리적 코어에 매핑이 되어 실행이 된다. 이 때, 가상 머신 간의 메모리 공유가 없다면 서로 다른 가상 머신들이 매핑된 물리적 코어의 로컬 캐쉬 간의 일관성을 맞출 필요가 없게 되고, 같은 가상 머신에 속한 가상 프로세서들이 매핑된 코어들 간의 로컬 캐쉬들 간에만 캐쉬 일관성을 맞춰 주면 된다.
- [0029] 따라서, 이하에서 기술될 본 발명의 실시예들은 이러한 가상화의 특징을 이용하여 캐쉬 간의 통신량을 줄이기 위해 각 가상 머신의 가상 프로세서가 어떤 코어에 매핑되어 있는지에 대한 정보를 추적하고, 이 정보를 이용하여 현재 같은 가상 머신에 속한 가상 프로세서들이 매핑되어 있는 코어들에게만 캐쉬 일관성을 위한 메시지를 전송하도록 한다. 즉, 본 발명의 실시예들의 기본 아이디어는 가상화 환경을 갖춘 멀티-코어 시스템에서 가상화의 특성을 고려하여 코어들 간의 일관성을 유지하기 위해 발생하는 통신량을 감소시킬 수 있는 일관성 제어 기법을 제시하고자 한다.
- [0030] 다만, 가상 머신들 간에는 공유되는 영역이 존재할 수도 있고, 또한 가상 머신 모니터 영역은 모든 가상 머신에 의해서 호출되어 어느 물리적 코어의 로컬 캐쉬에도 데이터가 적재될 수 있기 때문에, 이러한 데이터의 성격에 따라 그 유형을 구분하여 각각의 유형에 따라 캐쉬 미스에 따른 일관성 처리를 다르게 할 필요가 있다. 이를 위해, 본 발명의 다양한 실시예들은 페이지 테이블(도 1에서는 섀도우 페이지 테이블(15)이 될 수 있다.) 엔트리에 각각의 페이지에 대한 공유 상태 정보를 저장함으로써 현재 캐쉬 미스가 발생한 페이지가 다른 가상 머신에

의해 공유되는 페이지인지 여부를 검사한다.

- [0031] 보다 구체적으로, 이상과 같은 일관성 유지를 수행하기 위해 각각의 가상 머신의 데이터가 현재 어느 코어의 로컬 캐쉬에 존재하고 있는지가 추적되어야 하고, 추적된 코어들에 대한 정보가 자료구조에 저장되어야 한다. 이러한 정보는 어떤 가상 머신이 어떤 코어에 새롭게 스케줄링될 때 가상 프로세서 맵 레지스터(13)에 적재된다. 이렇게 적재된 가상 프로세서 맵 레지스터(13)는 해당 코어에서 캐쉬 미스가 발생하여 일관성 메시지를 보내야 할 때 항상 참조된다.
- [0032] 이제, 일관성 메시지를 전송할 대상을 선택하는 방법을 설명한다. 일관성 메시지는 캐쉬 미스가 발생한 시점에 해당 가상 머신의 데이터를 보유하고 있는 코어들에게만 전송되어야 한다. 이를 위해 먼저 캐쉬 미스가 감지된 데이터에 대해 가상 머신에 의한 페이지 공유 여부를 검사할 필요가 있다. 도 1에는 가상화 환경이 구현된 멀티-코어 시스템(100)에서 가상 주소를 실제 물리 주소로 바꾸어 주는 새도우 페이지 테이블(15)에 각각의 페이지들에 대한 공유 정보를 기록함으로써, 캐쉬 미스가 발생한 경우 일관성 메시지를 전송하기 전에 새도우 페이지 테이블(15)에 기록된 공유 정보를 참조한다.
- [0033] 만약, 현재 캐쉬 미스가 발생한 데이터가 공유된 페이지에 포함되어 있지 않다면, 가상 프로세서 맵 레지스터(13)를 통해 참조된 코어들에게만 일관성 메시지를 전송하면 된다. 이에 반해, 현재 캐쉬 미스가 발생한 데이터가 다른 가상 머신이나 다른 가상 머신 모니터에 의해 공유된 페이지에 포함되어 있다면, 다른 가상 머신의 데이터를 보유하고 있는 코어도 동일한 데이터를 보유하고 있을 가능성이 높으므로, 해당 가상머신의 데이터를 가지고 있는 코어들에게만 일관성 메시지를 보내는 것만으로는 캐쉬 일관성 유지에 불충하다. 이러한 경우의 일관성 유지 방법은 이후에 도 4를 통해 구체적으로 설명하도록 한다.
- [0034] 도 2는 본 발명의 일 실시예에 따른 가상화 환경에서 가상 머신의 재배치가 이루어지는 과정을 예시한 도면으로서, 가상 머신의 가상 프로세서가 고정되어 있지 않고 코어의 이용률을 높이기 위해 옮겨다니는 상황을 예시하고 있다.
- [0035] 도 2에서 캐쉬 미스가 발생한 경우, 해당 가상 머신의 데이터를 가지고 있는 모든 코어에게 일관성 메시지를 올바르게 보내려면 새롭게 이용되는 코어를 스눅 도메인에 적절히 추가해주는 것이 필요하다. 해당 가상 머신이 가상 프로세서 맵 레지스터에 없는 코어를 새롭게 사용하게 된다면, 새로 사용되는 코어를 새로운 로컬 캐쉬에 데이터를 로드하기 전에 가상 프로세서 맵 레지스터에 추가하여야만 한다. 또한, 현재 새롭게 사용되고 있는 코어의 가상 프로세서 맵 레지스터뿐만 아니라 해당 가상 머신의 매핑 정보를 로드하고 있는 모든 코어의 가상 프로세서 맵 레지스터에 업데이트 메시지를 전송함으로써 각각의 가상 프로세서 맵 레지스터를 업데이트해야 한다. 이렇게 업데이트된 가상 프로세서 맵 레지스터의 값은 이후 각 가상 머신의 스눅 도메인을 나타내는 자료구조에 저장된다. 이상과 같은 과정을 통해 해당 가상 머신의 데이터를 가지고 있는 모든 코어들에게 일관성 메시지를 보낼 수 있지만, 가상 프로세서 재배치로 인하여 계속해서 가상 프로세서 맵 레지스터의 엔트리가 증가하게 된다면, 가상 프로세서의 개수가 실제 코어의 수보다 적다고 하더라도 실제 코어의 수와 같거나 이에 준하는 코어의 수가 스눅 도메인에 포함될 수 있다.
- [0036] 도 3은 본 발명의 일 실시예에 따른 가상화 환경에서 캐쉬 일관성을 제어하는 방법을 도시한 흐름도로서, 다음과 같은 단계들을 포함한다. 이 때, 도 3의 가상화 환경은 복수 개의 가상 머신이 복수 개의 코어를 공유하는 환경을 말한다.
- [0037] 310 단계에서 가상 머신의 가상 프로세서와 코어들 간의 매핑(mapping) 정보 및 페이지의 공유 여부를 나타내는 상태 정보를 생성한다. 여기서, 매핑 정보는 가상 머신의 가상 프로세서가 사용하는 페이지 단위의 데이터가 코어들 중 어느 코어의 로컬 캐쉬에 존재하는지를 추적하여 추적된 코어를 기록함으로써 생성된 정보를 말한다.
- [0038] 320 단계에서 캐쉬 미스의 발생이 감지되면, 310 단계를 통해 생성된 매핑 정보 및 상태 정보를 이용하여 캐쉬 미스가 감지된 데이터에 대해 가상 머신에 의한 페이지(page) 공유 여부를 검사한다.
- [0039] 330 단계에서는 320 단계의 공유 여부 검사 결과에 따라 동일한 가상 머신에 속한 가상 프로세서들이 매핑된 코어를 고려하여 선택적으로 일관성 메시지를 전송한다. 보다 구체적으로 일관성 메시지를 전송하는 과정은 다음과 같다.
- [0040] 우선, 캐쉬 미스의 발생을 감지한다. 이러한 감지 과정은 통상적인 멀티-코어 시스템이 구현되는 과정에서 필수적으로 활용되고 있는 구성이므로 여기서는 자세한 설명을 생략한다. 다음으로, 310 단계를 통해 생성된 상태 정보를 이용하여 캐쉬 미스가 감지된 데이터에 대해 가상 머신에 의한 페이지 공유 여부를 검사한다. 페이지 공유 여부는 새도우 페이지 테이블에 기록된 페이지 별 상태 정보를 참조함으로써 수행될 수 있다. 이러한 검사

결과에 기초하여 가상 머신 모니터(virtual machine monitor)가 310 단계를 통해 생성된 매핑 정보를 참조하여 가상 머신에 매핑된 코어를 선택한다. 이렇게 선택된 코어는 페이지 공유에 관한 유형에 따라 결정될 수 있으며, 보다 구체적인 유형과 선택 방법은 이후 도 4를 통해 설명한다. 이제, 이상에서 선택된 코어에 일관성 메시지를 전송한다.

- [0041] 도 4는 본 발명의 일 실시예에 따른 도 3의 캐쉬 일관성 제어 방법을 보다 구체적으로 도시한 흐름도로서, 페이지 공유 여부 유형에 따른 3 가지 일관성 메시지 전송 방법을 제시한다.
- [0042] 410 단계에서 우선 매핑 정보 및 상태 정보를 생성하여 저장한다. 즉, 가상 머신의 가상 프로세서와 코어들 간의 매핑 정보를 생성하여 가상 프로세서 맵 레지스터에 저장하고, 또한 페이지의 공유 여부를 나타내는 상태 정보를 생성하여 새도우 페이지 테이블에 저장한다.
- [0043] 420 단계에서는 캐쉬 미스가 발생했는지 여부를 감지한다. 이러한 감지는 각 코어에서 데이터 접근이 일어났을 때 수행되며, 캐쉬 미스가 발생하지 않았다면 절차는 종료된다. 반면, 캐쉬 미스가 발생했다면 430 단계로 진행하여 캐쉬 일관성을 유지하기 위한 일련의 절차가 수행된다.
- [0044] 430 단계에서 페이지 공유 여부를 검사한다. 이 때, 페이지의 공유 여부 판단을 위해 새도우 페이지 테이블과 변환 색인 버퍼(translation lookaside buffer, TLB)를 참조하여 페이지 공유 여부를 판단한다. 이를 위해 본 발명의 실시예들은 페이지 공유 상태를 적어도 다음과 같은 3 가지 유형으로 구분하였다.
- [0045] 첫째, 페이지가 복수 개의 가상 머신에 의해 공유되지 않음을 나타내는 가상 머신 독점(virtual machine private) 상태가 분류 가능하다.
- [0046] 둘째, 페이지가 복수 개의 가상 머신에 의해 공유되고 가상 머신들 각각이 공유 페이지에 대해 읽기 및 쓰기가 가능함을 나타내는 읽기-쓰기 공유(read-write shared) 상태가 분류 가능하다.
- [0047] 셋째, 페이지가 복수 개의 가상 머신에 의해 공유되고 가상 머신들 각각이 공유 페이지에 대해 읽기만이 가능함을 나타내는 읽기-전용 공유(read-only shared) 상태가 분류 가능하다.
- [0048] 본 발명의 실시예들에서는 공유되지 않은 가상 머신 독점의 경우와 현재 해당 가상 머신의 스누프 도메인에 속하지 않으면서 다른 가상 머신의 스누프 도메인에 포함된 로컬 캐쉬에 공유된 데이터가 존재할 수 있는 읽기-쓰기 공유 및 읽기-전용 공유의 유형을 포함하여 총 3 가지 유형을 구분하기 위한 구분자(페이지 공유 상태 정보를 의미한다.)를 사용한다. 구현의 측면에서 이러한 상태 정보는 새도우 페이지 테이블 엔트리와 변환 색인 버퍼의 엔트리에 추가적인 2 비트(bit)를 사용하여 표시 가능하다.
- [0049] 430 단계의 검사 결과, 만약 해당 페이지가 공유된 페이지가 아니라면(즉, 가상 머신 독점 상태(440 단계)인 경우를 의미한다.) 445 단계로 진행한다. 445 단계에서는 가상 프로세서 맵 레지스터(virtual processor map register)를 참조함으로써 독출된 스누프 도메인(snoop domain)에 포함된 코어만을 선택하고, 선택된 코어들에게만 일관성 메시지를 전송한다.
- [0050] 반면, 해당 페이지가 공유된 페이지이고, 읽기-쓰기 공유 상태라면(450 단계) 455 단계로 진행한다. 455 단계에서는 모든 코어들에게 일관성 메시지를 브로드캐스팅(broadcasting)한다. 예를 들어, 대표적인 읽기-쓰기 가능 공유 데이터인 가상 머신 모니터의 데이터에 캐쉬 미스가 발생한 경우를 가정하자. 가상 머신 모니터 영역은 어느 가상 머신에 의해서도 호출될 수 있기 때문에, 가상 머신 모니터의 데이터는 어떠한 물리적 코어의 로컬 캐쉬에도 쉽게 적재될 수 있다는 특징을 갖는다. 따라서, 해당 페이지가 어느 코어에 존재하는지를 추적하는 것보다 브로드캐스팅하는 것이 보다 효과적이다.
- [0051] 한편, 해당 페이지가 공유된 페이지이고, 읽기-전용 공유 상태라면 465 단계로 진행한다. 465 단계에서는 읽기-쓰기 공유 상태와 같이 브로드캐스팅할 수도 있으나, 보다 효율적인 메시지 전송을 위해 다른 접근 방법이 활용될 수 있다. 물론 읽기-전용 페이지를 위한 일관성 메시지도, 읽기-쓰기 가능한 페이지처럼 브로드캐스팅을 이용하면 간단히 해결되겠지만, 가상 머신 간의 공유되는 페이지가 많아진다면 이 역시 각 로컬 캐쉬들의 일관성을 맞추주기 위해 많은 통신량을 야기할 우려가 있다. 따라서, 본 발명의 실시예들에서는 다음과 같은 적어도 3 가지 최적화 기법을 소개하고 있는데, 각각은 메모리 직접 방법(memory-direct method)과 내부 가상 머신 방법(intra-VM method), 그리고 프렌드 가상 머신 방법(friend-VM method)이다.
- [0052] 읽기-전용 공유 데이터의 처리에 보다 다른 접근 방법이 필요한 이유는 다음과 같다. 읽기-전용 공유 페이지의 대표적인 경우는 가상화 환경에서 메모리 절약을 위해 사용되는 기법인 내용기반공유(content-based sharing) 기법에 의해 공유 되는 페이지들을 예로 들 수 있다. 이러한 읽기-전용 공유는 먼저 쓰기 연산이 일어나면 캐피

온 라이트(copy on write)가 일어나서 새로운 물리 주소에 새로운 페이지가 할당되고, 기존 공유 페이지에 대해 별도의 캐쉬 일관성을 맞춰줄 필요가 없게 된다. 따라서, 읽기-전용 공유 페이지의 일관성을 위한 통신의 경우 읽기 연산에 대한 고려만 하면 되는데, 이는 쓰기 연산처럼 반드시 데이터를 로드하고 있는 모든 로컬 캐쉬에 일관성 메시지를 보낼 필요가 없다. 그 이유는 메모리로부터 언제든지 올바른 데이터를 가져올 수 있기 때문이다. 만약, 다른 코어의 로컬 캐쉬에 데이터가 존재하고 있었는데 그 캐쉬에 일관성 메시지를 못 보내게 된 경우, 해당 캐쉬로부터 데이터를 가져오지 못하더라도 메모리로부터 원하는 데이터를 가져올 수 있다. 따라서, 일관성 메시지의 수를 줄이기 위해 추가적인 개선이 가능하다. 이하에서는 읽기-전용 페이지의 일관성 메시지를 감소시키기 위한 3 가지 개선 방법을 제시한다.

[0053] 첫째, 메모리 직접 방법은 캐쉬 미스된 데이터를 메모리로부터 직접 독출한다. 보다 구체적으로, 메모리 직접 방법은 읽기-전용 페이지에 대한 읽기 연산에 의해서 캐쉬 미스가 발생할 경우, 일관성 메시지를 발생시키지 않고 무조건 메모리로 요청 메시지를 전송한다. 다만, 이 방법은 캐쉬 일관성에 의한 통신량은 줄일 수 있겠지만, 만약 그 데이터가 다른 코어의 로컬 캐쉬에 존재한다면, 캐쉬에서 데이터를 가져올 수 있음에도 불구하고 메모리에서 데이터를 가져 오게 되므로, 데이터를 가져오는 시간의 측면에서 손해를 볼 우려가 있다. 따라서, 캐쉬에서 데이터를 가져올 수 있는 기회를 좀 더 얻기 위해 다음의 다른 방법들이 활용 가능하다.

[0054] 둘째, 내부 가상 머신 방법은 매핑 정보를 이용하여 캐쉬 미스가 발생한 가상 머신에 매핑된 코어 및 메모리에 동시에 캐쉬 미스된 데이터를 요청한다. 즉, 내부 가상 머신 방법은 현재 캐쉬 미스를 일으킨 가상 머신이 가상 프로세서 맵 레지스터의 엔트리에 기록되어 있는 코어들과 메모리로 동시에 데이터 요청을 보내는 방법이다. 이 방법을 이용하면 일관성 메시지에 의한 데이터 통신량은 메모리 직접 방법보다는 상대적으로 증가하지만, 관련이 있는 코어에 데이터 요청을 보내므로 증가하는 통신량에 비해서 데이터를 캐쉬에서 가져올 확률을 높일 수 있다.

[0055] 셋째, 프렌드 가상 머신 방법은 복수 개의 가상 머신 별로 공유가 빈번하게 발생하는 가상 머신들을 프렌드 가상 머신으로 미리 설정하고, 매핑 정보를 이용하여 캐쉬 미스가 발생한 가상 머신에 매핑된 코어, 프렌드 가상 머신에 매핑된 코어 및 메모리에 동시에 캐쉬 미스된 데이터를 요청한다. 즉, 프렌드 가상 머신 방법은 가상 머신 별로 공유가 많이 일어나는 가상 머신들을 프렌드 가상 머신으로 설정하여, 데이터 요청을 할 때 현재 캐쉬 미스를 일으킨 가상 머신의 스눅 도메인, 프렌드 가상 머신의 스눅 도메인, 및 메모리에 동시에 데이터 요청을 보내는 방법이다. 이 방법은 이상에서 소개한 방법들에 비해 상대적으로 조금 더 통신량을 증가시키지만, 데이터를 캐쉬에서 가져올 수 있는 확률을 보다 높일 수 있다. 이를 위해, 가상 머신 모니터는 각 가상 머신들 간의 공유 정도를 파악해야 하며, 공유 정도에 따라 적절하게 프렌드 가상 머신을 설정해 주어야 한다. 프렌드 가상 머신을 설정함으로써 캐쉬 일관성을 제어하는 방법은 다음의 도 5의 예시도를 통해 다시 설명한다.

[0056] 도 5는 본 발명의 일 실시예에 따른 가상화 환경에서 캐쉬 미스가 발생한 페이지가 읽기-전용 공유 상태인 경우 캐쉬 일관성을 제어하는 방법의 일례를 도시한 도면으로서, 각각의 가상 머신들(510, 520, 530)은 4 개의 코어를 스눅 도메인으로 할당받았음을 예시하고 있다.

[0057] 또한, 도 5는 각각의 가상 머신들(510, 520, 530) 별로 프렌드 가상 머신의 설정 정보(550)가 유지되고 있음을 예시하고 있다. 도시된 바에 따르면, 가상 머신 VM1은 가상 머신 VM2와 40%의 페이지를 공유하고 있으므로 가상 머신 모니터에 의해 VM1과 VM2는 프렌드 가상 머신으로 설정이 되었다. 반면, 가상 머신 VM1은 가상 머신 VM3와 1%의 페이지를 공유하고 있으므로 프렌드 가상 머신으로 설정되지 않는다.

[0058] 이제, 가상 머신 VM1이 자신의 코어인 P0에서 읽기-전용 공유 페이지에 대해 읽기 연산에 의한 캐쉬 미스를 일으켰다고 가정하자. 이러한 상황에서 프렌드 가상 머신 방법에 따르면, 가상 머신 VM1과 가상 머신 VM2의 스눅 도메인에 대해서는 일관성 메시지가 전송이 되고, 이와 동시에 메모리로 데이터 요청이 전송된다. 반면, 가상 머신 VM3는 가상 머신 VM1의 프렌드 가상 머신이 아니므로 일관성 메시지가 전송되지 않는다.

[0059] 도 6은 본 발명의 다른 실시예에 따라 레지던스 카운터(17)를 이용하여 가상 프로세서 맵 레지스터(13)의 스눅 도메인을 갱신하는 과정을 예시한 도면으로서, 본 실시예에서는 복수 개의 코어(50) 별로 해당 코어(50)의 로컬 캐쉬(local cache)(55)에 적재된 가상 머신의 데이터의 수를 나타내는 레지던스 카운터(residence counter)(17)를 산출하여 저장한다.

[0060] 보다 구체적으로, 각 코어(50) 별로 레지던스 카운터(17)를 이용하여 해당 코어(50)의 로컬 캐쉬(55)에 어떤 가상 머신의 데이터가 새롭게 로드가 되면 카운터를 1 증가시켜 저장하고, 가상 머신의 데이터의 교체(replacement)가 발생하면 해당 카운터를 1 감소시켜 저장한다. 즉, 현재 코어(50)의 로컬 캐쉬(55)에 로드되어

있는 데이터의 수를 각 가상 머신 별 카운터를 통해서 나타낼 수 있다. 따라서, 레지던스 카운터(17)가 0이 되는 순간, 해당 가상 머신의 스눅 도메인에서 해당 코어를 제거해주면, 스눅 도메인으로부터 더 이상 모니터링할 필요 없는 불필요한 코어들을 제거할 수 있다. 이를 위해 본 발명의 실시예는 어떠한 코어(50)에 대응하는 레지던스 카운터(17)가 0인 경우, 가상 프로세서 맵 레지스터(13)에 저장된 매핑 정보의 스눅 도메인에서 이에 해당하는 코어를 삭제한다.

[0061] 도 6에서는 코어 P0(50)에 대응하는 레지던스 카운터(17)에서 가상 머신 VM1의 데이터가 교체되면서 가상 머신 VM1의 카운터 값이 0이 되는 경우를 예시하였다. 그러면, 가상 프로세서 맵 레지스터(13)는 가상 머신 VM1의 스눅 도메인으로부터 P0를 제거하게 된다.

[0062] 상기 기재된 본 발명의 다양한 실시예들에 따르면, 가상 프로세서와 코어들 간의 매핑 정보 및 페이지의 공유 여부를 나타내는 상태 정보를 이용하여 코어들에 선택적으로 일관성 메시지를 전송하는 기술을 제안함으로써 가상화 기술을 멀티-코어 기반의 시스템에 확장하여 적용할 수 있고, 멀티-코어에 기반한 가상화 환경 하에서 캐쉬 일관성을 유지하기 위해 전송하는 일관성 메시지에 따른 통신량을 감소시킴으로써 시스템의 성능을 향상시키며, 이러한 통신량 감소로 인해 전력 소모를 최소화할 수 있다.

[0063] 한편, 본 발명은 컴퓨터로 읽을 수 있는 기록 매체에 컴퓨터가 읽을 수 있는 코드로 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록 매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록 장치를 포함한다.

[0064] 컴퓨터가 읽을 수 있는 기록 매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광 데이터 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구현하는 것을 포함한다. 또한, 컴퓨터가 읽을 수 있는 기록 매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산 방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다. 그리고 본 발명을 구현하기 위한 기능적인(functional) 프로그램, 코드 및 코드 세그먼트들은 본 발명이 속하는 기술 분야의 프로그래머들에 의하여 용이하게 추론될 수 있다.

[0065] 도 7은 본 발명의 일 실시예에 따른 가상화 환경에서 캐쉬 일관성을 제어하는 장치(100)를 도시한 블록도로서, 모니터링부(10), 공유 검사부(20) 및 처리부(30)를 포함한다. 또한, 본 실시예가 가정하고 있는 바에 따라 복수 개의 가상 머신이 복수 개의 코어(50)를 공유하고 있으며, 각각의 코어(50)는 캐쉬를 구비한다. 도 7에 도시된 각각의 구성은 앞서 도 3 및 도 4를 통해 구체적으로 소개한 캐쉬 일관성 제어 방법의 각 단계에 대응하는 것으로서, 여기서는 구체적인 내용을 생략하고 그 개요만을 설명한다.

[0066] 모니터링부(10)는 가상 머신의 가상 프로세서와 코어들 간의 매핑 정보를 생성하여 가상 프로세서 맵 레지스터(13)에 저장하고, 페이지의 공유 여부를 나타내는 상태 정보를 생성하여 새도우 페이지 테이블(15)에 저장한다. 여기서 상태 정보는, 가상 머신 독점 상태, 읽기-쓰기 공유 상태 및 읽기-전용 공유 상태 중 어느 하나를 의미한다. 또한, 모니터링부(10)는 복수 개의 가상 머신 별로 할당된 가상 프로세서를 매핑시켜 스눅 도메인으로서 가상 프로세서 맵 레지스터(13)에 저장한다.

[0067] 한편, 모니터링부(10)는 복수 개의 코어 별로 해당 코어의 로컬 캐쉬에 적재된 가상 머신의 데이터의 수를 나타내는 레지던스 카운터(17)를 산출하여 저장할 수 있으며, 어떤 코어에 대응하는 레지던스 카운터가 0인 경우, 모니터링부(10)는 매핑 정보의 스눅 도메인에서 해당 코어를 삭제함으로써 불필요한 코어를 모니터링 대상에서 배제시킨다.

[0068] 공유 검사부(20)는 캐쉬 미스의 발생이 감지되면, 가상 프로세서 맵 레지스터(13)에 저장된 매핑 정보 및 새도우 페이지 테이블(15)에 저장된 상태 정보를 이용하여 캐쉬 미스가 감지된 데이터에 대해 가상 머신에 의한 페이지 공유 여부를 검사한다. 이러한 공유 검사부(20)는 데이터 접근시 새도우 페이지 테이블(15) 및 변환 색인 버퍼(25)를 참조하여 페이지 공유 여부를 검사한다.

[0069] 처리부(30)는 공유 검사부(20)에 의한 검사 결과에 따라 동일한 가상 머신에 속한 가상 프로세서들이 매핑된 코어를 고려하여 선택적으로 일관성 메시지를 전송한다.

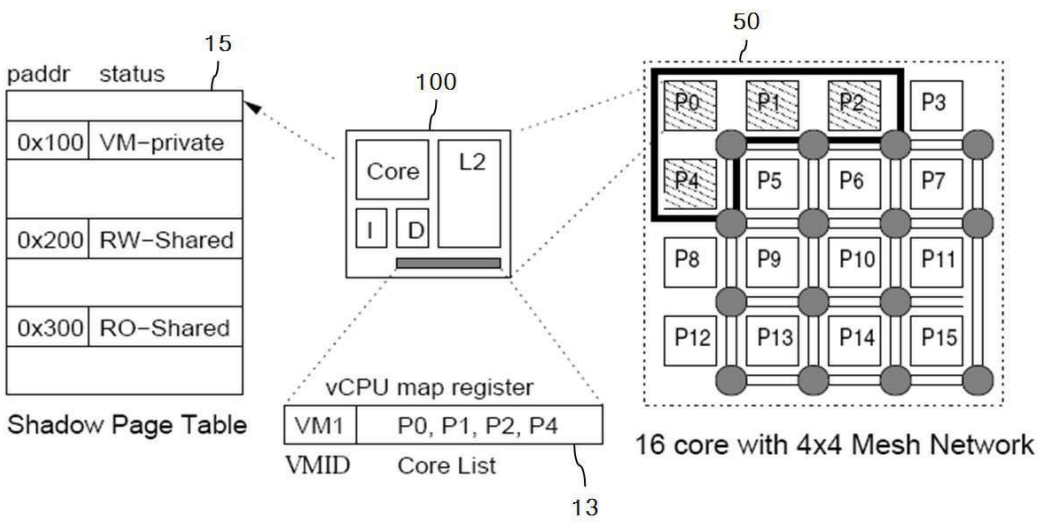
[0070] 이상에서 본 발명에 대하여 그 다양한 실시예들을 중심으로 살펴보았다. 본 발명에 속하는 기술 분야에서 통상의 지식을 가진 자는 본 발명이 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 본 발명의 범위는 전술한 설명이 아니라 특허청구범위에 나타나 있으며, 그와 동등한 범위 내에 있는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이다.

부호의 설명

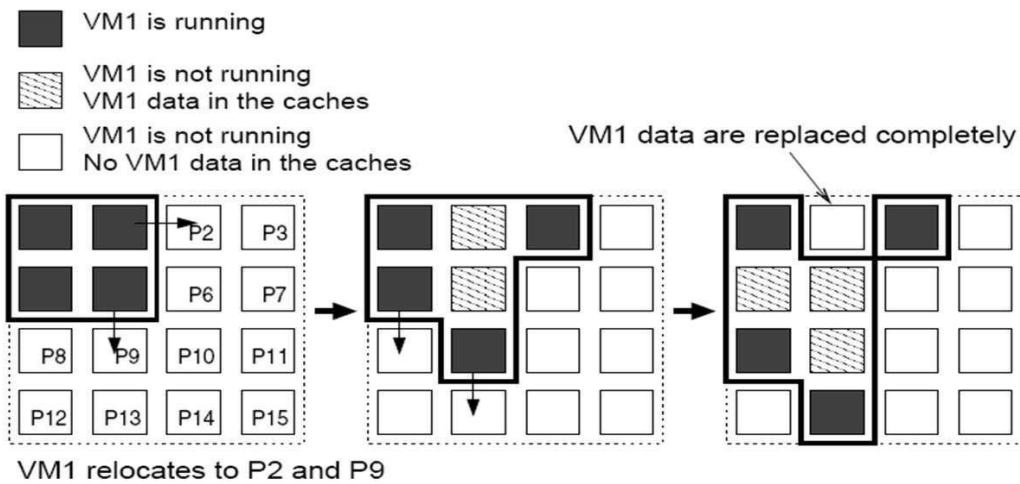
- [0071] 100 : 캐쉬 일관성 제어 장치
- 10 : 모니터링부
- 15 : 새도우 페이지 테이블
- 20 : 공유 검사부
- 30 : 처리부
- 50 : 코어
- 510, 520, 530 : 가상 머신
- 13 : 가상 프로세서 맵 레지스터
- 17 : 레지던스 카운터
- 25 : 변환 색인 버퍼
- 55 : 캐쉬
- 550 : 프렌드 머신 설정 정보

도면

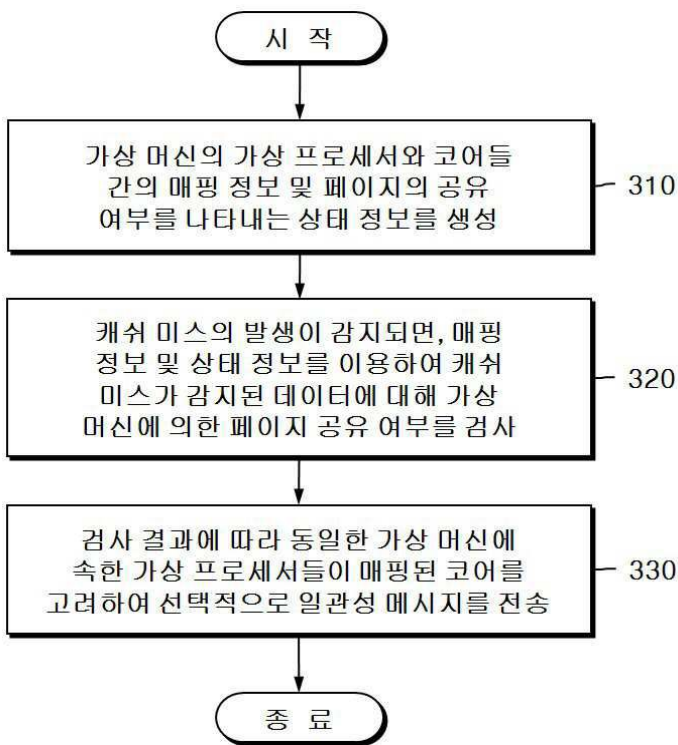
도면1



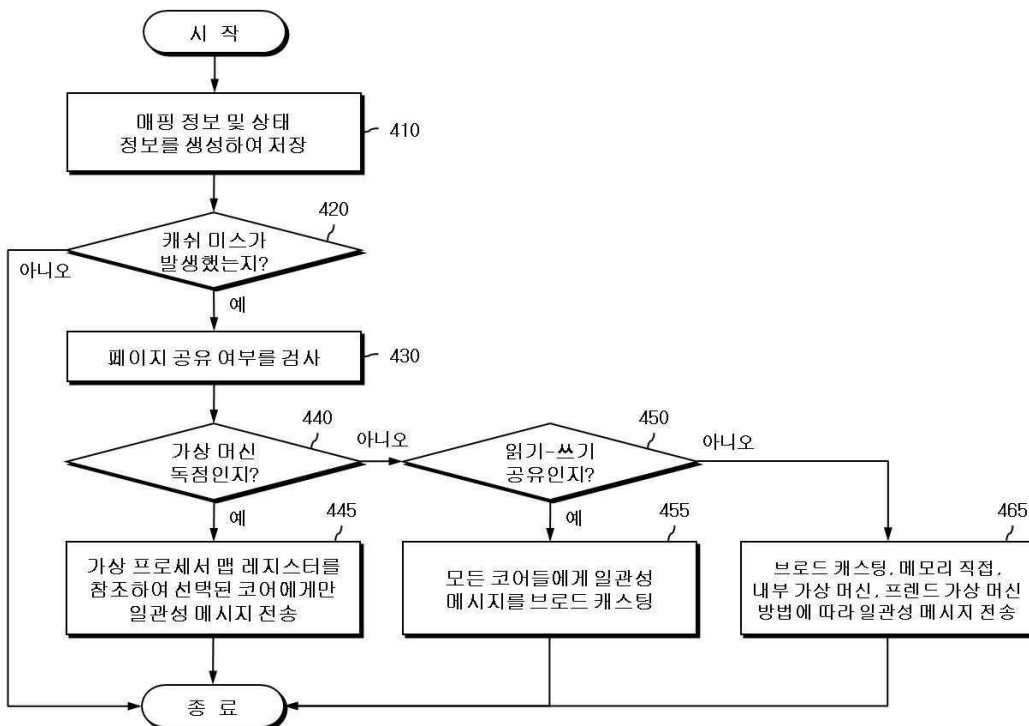
도면2



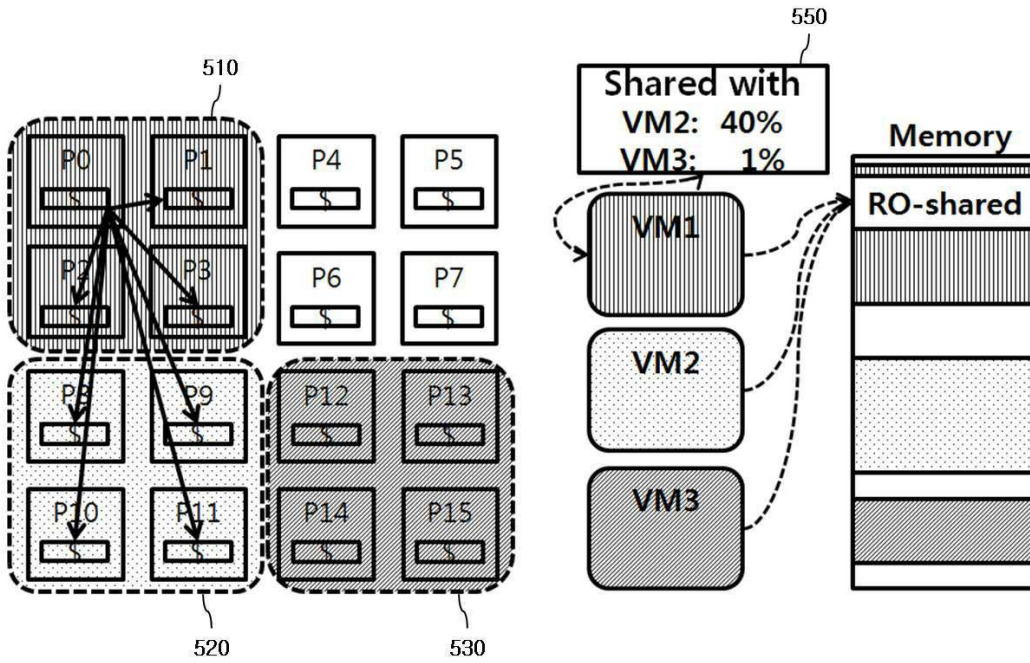
도면3



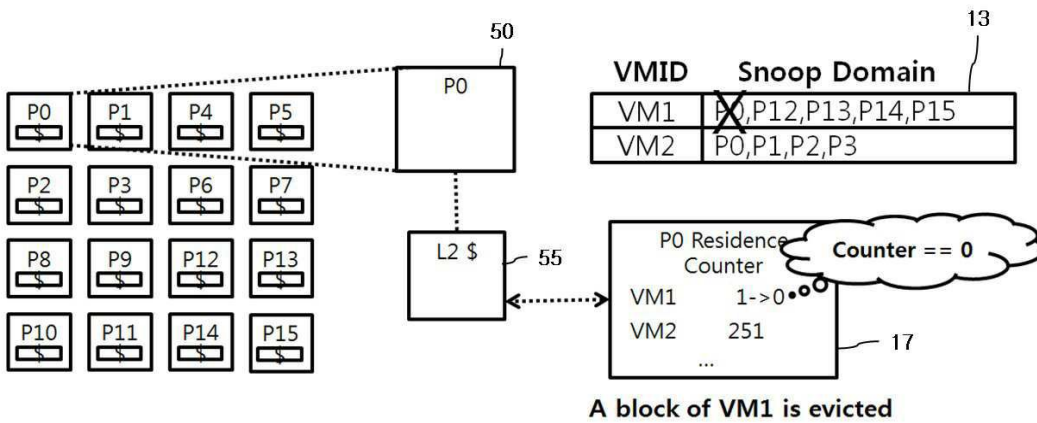
도면4



도면5



도면6



도면7

