



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2014-0092493
(43) 공개일자 2014년07월24일

(51) 국제특허분류(Int. Cl.)
G06F 12/08 (2006.01) G06F 9/06 (2006.01)
(21) 출원번호 10-2012-0158026
(22) 출원일자 2012년12월31일
심사청구일자 2012년12월31일

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
유혁
서울 강남구 광평로39길 3, 801동 101호 (수서동, 동익아파트)
이치영
서울 성북구 안암로 145, 우정정보통신관 308호 (안암동5가, 고려대학교이과대학)
윤중희
경기도 수원시 영통구 매탄4동 원천아파트 103동 1506호
(74) 대리인
특허법인엠에이피에스

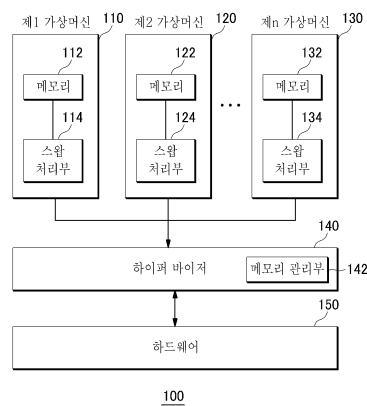
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 가상화 시스템에서의 메모리 관리 방법

(57) 요약

본 발명의 일 측면에 따른 가상화 시스템에서의 메모리 관리 방법은 제 1 페이지 데이터가 기록된 제 2 가상 머신의 메모리에 대하여 제 2 페이지 데이터의 쓰기 동작시에 메모리 부족이 발생한 경우, 제 1 가상 머신의 메모리로부터 제 1 스왑 캐시를 대여시키는 단계; 상기 제 1 페이지 데이터 중 일부 또는 전체를 상기 대여된 제 1 스왑 캐시로 스왑 아웃시키는 단계; 및 상기 제 2 가상 머신의 메모리의 여유 공간에 상기 제 2 페이지 데이터를 저장시키는 단계를 포함한다.

대표도 - 도1



특허청구의 범위

청구항 1

가상화 시스템에서의 메모리 관리 방법에 있어서,

제 1 페이지 데이터가 기록된 제 2 가상 머신의 메모리에 대하여 제 2 페이지 데이터의 쓰기 동작시에 메모리 부족이 발생한 경우, 제 1 가상 머신의 메모리로부터 제 1 스왑 캐시를 대여하는 단계;

상기 제 1 페이지 데이터 중 일부 또는 전체를 상기 대여한 제 1 스왑 캐시로 스왑 아웃하는 단계; 및

상기 제 2 가상 머신의 메모리의 여유 공간에 상기 제 2 페이지 데이터를 저장하는 단계를 포함하는 가상화 시스템에서의 메모리 관리 방법.

청구항 2

제 1 항에 있어서,

상기 제 1 페이지 데이터 중 일부 또는 전체를 상기 대여된 제 1 스왑 캐시로 스왑 아웃하는 단계는 상기 제 1 페이지 데이터 중 일부 또는 전체를 압축하여 상기 대여된 제 1 스왑 캐시로 스왑 아웃하는 것인 가상화 시스템에서의 메모리 관리 방법.

청구항 3

제 1 항에 있어서,

제 3 페이지 데이터가 기록된 제 3 가상 머신의 메모리에 대하여 제 4 페이지 데이터의 쓰기 동작시에 메모리 부족이 발생한 경우, 상기 제 1 가상 머신의 메모리로부터 제 2 스왑 캐시를 대여하는 단계;

상기 제 3 페이지 데이터 중 일부 또는 전체를 상기 대여된 제 2 스왑 캐시로 스왑 아웃하는 단계; 및

상기 제 3 가상 머신의 메모리의 여유 공간에 상기 제 4 페이지 데이터를 저장하는 단계를 더 포함하는 가상화 시스템에서의 메모리 관리 방법.

청구항 4

제 3 항에 있어서,

상기 제 3 페이지 데이터 중 일부 또는 전체를 상기 대여된 제 2 스왑 캐시로 스왑 아웃하는 단계는 상기 제 3 페이지 데이터 중 일부 또는 전체를 압축하여 상기 대여된 제 2 스왑 캐시로 스왑 아웃하는 것인 가상화 시스템에서의 메모리 관리 방법.

청구항 5

제 3 항에 있어서,

상기 제 1 스왑 캐시로 스왑 아웃된 데이터와 제 2 스왑 캐시로 스왑 아웃된 데이터가 동일한 경우, 상기 제 1 스왑 캐시 및 제 2 스왑 캐시 중 어느 하나에 저장된 데이터를 삭제하고, 나머지 스왑 캐시에 저장된 데이터를 상기 제 2 가상 머신 및 제 3 가상 머신의 메모리에 의하여 참조되도록 공유시키는 단계를 더 포함하는 가상화 시스템에서의 메모리 관리 방법.

청구항 6

제 1 항에 있어서,

제 3 페이지 데이터가 기록된 제 3 가상 머신의 메모리에 대하여 제 4 페이지 데이터의 쓰기 동작시에 메모리 부족이 발생한 경우, 상기 제 1 가상 머신의 메모리에 대하여 제 2 스왑 캐시의 대여 요청을 전달하는 단계;

상기 제 2 스왑 캐시의 대여 요청에 따라, 제 1 스왑 캐시에 저장된 데이터와 제 2 스왑 캐시에 저장할 데이터의 중복 여부를 판단하는 단계,

상기 제 1 스왑 캐시에 저장된 데이터와 제 2 스왑 캐시에 저장할 데이터가 동일한 경우, 제 1 스왑 캐시의 주소 정보를 제 2 스왑 캐시에 저장할 데이터와 매칭시키는 단계;

상기 제 1 스왑 캐시에 저장된 데이터와 제 2 스왑 캐시에 저장할 데이터가 상이한 경우, 제 2 스왑 캐시를 할당하고, 할당된 제 2 스왑 캐시로 상기 제 3 페이지 데이터 중 일부 또는 전체를 상기 대여된 제 2 스왑 캐시로 스왑 아웃하는 단계; 및

상기 제 3 가상 머신의 메모리의 여유 공간에 상기 제 4 페이지 데이터를 저장하는 단계를 더 포함하는 가상화 시스템에서의 메모리 관리 방법.

청구항 7

가상화 시스템에서의 메모리 관리 방법에 있어서,

제 1 가상 머신이 제 2 가상 머신으로부터 스왑 캐시 할당 요청을 수신하는 단계;

상기 스왑 캐시 할당 요청에 포함된 대상 페이지 데이터와 제 1 가상 머신의 스왑 캐시에 기 저장된 페이지 데이터의 중복 여부를 판단하는 단계;

중복되는 페이지 데이터의 존재시에 해당 페이지 데이터를 저장하고 있는 스왑 캐시의 주소 정보를 상기 제 2 가상 머신으로 전송하는 단계 및

중복되는 페이지 데이터의 미 존재시에 새로운 스왑 캐시를 할당하고, 상기 대상 페이지 데이터를 상기 스왑 캐시에 저장시키는 단계를 포함하는 가상화 시스템에서의 메모리 관리 방법.

청구항 8

제 7 항에 있어서,

상기 중복 여부를 판단하는 단계는

상기 대상 페이지 데이터에 대한 해쉬값과 상기 제 1 가상 머신의 스왑 캐시에 기 저장된 페이지 데이터들의 해쉬값을 비교하는 단계를 포함하는 가상화 시스템에서의 메모리 관리 방법.

청구항 9

제 8 항에 있어서,

상기 중복 여부를 판단하는 단계는

상기 해쉬값을 비교하는 단계의 수행 결과, 상기 대상 페이지 데이터의 해쉬값과 동일한 해쉬값을 가진 페이지 데이터가 상기 제 1 가상 머신의 스왑 캐시에 존재하는 경우, 상기 대상 페이지 데이터와 해당 페이지 데이터를 비트 레벨에서 비교하는 단계를 포함하는 가상화 시스템에서의 메모리 관리 방법.

청구항 10

제 7 항에 있어서,

상기 제 2 가상 머신으로 전송된 스왑 캐시의 주소 정보는 상기 대상 페이지 데이터와 매칭되어 주소 테이블에 저장되는 가상화 시스템에서의 메모리 관리 방법.

명세서

기술분야

본 발명은 가상화 시스템에서의 메모리 관리 방법에 관한 것이다.

배경기술

가상화 시스템은 제한된 물리 시스템에서 다양한 컴퓨팅 환경을 제공하기 때문에, 최근에 많은 관심을 받고 있다. 가상화 시스템에서는 여러 개의 가상 머신이 생성되고, 이러한 가상 머신은 한정된 컴퓨팅 자원을 다른 가상 머신과 공유하기 때문에, 가상 머신 간에 중재가 필요하다. 그러나, 이러한 중재 과정은 추가적인 프로세싱

[0001]

[0002]

이 필요하기 때문에 가상 머신의 성능 저하를 유발할 수 있다.

[0003] 특히, 가상화 시스템에서의 메모리는 가상 머신의 수가 증가함에 따라 각 가상 머신이 이용할 수 있는 메모리 용량의 크기는 작아진다. 이는 메모리를 분할하여 각 가상 머신에게 할당하기 때문이다. 메모리는 실행 중인 프로세스의 코드, 데이터 등이 적재되는 저장소이기 때문에, CPU처럼 스케줄링에 의해 시간적으로 나누어 사용할 수 없다. 서로 다른 프로세스를 실행 중인 각 가상 머신은 메모리에 적재되는 내용이 다르기 때문이다. 따라서 가상화 시스템은 메모리를 공간적으로 분할하여 각 가상 머신에게 할당한다. 이로 인해 각 가상 머신은 자신에게 할당된 작은 메모리만 사용할 수 있다. 또한, 가상 머신의 수가 증가하면 메모리를 더 작은 크기로 분할하기 때문에 사용할 수 있는 메모리의 크기가 작아지는 문제가 발생한다.

[0004] 따라서, 본 발명은 가상화 시스템에서의 메모리 사용의 효율성을 극대화하고 성능 저하를 최소화하고자 한다. 또한, 작은 메모리를 가진 가상 머신들의 추가적인 메모리 이용량을 보장함으로써, 해당 가상 머신들의 활용도를 상승시킨다.

[0005] 이와 관련하여, 한국등록특허(제1077782호, 발명의 명칭: 가상 머신의 메모리 관리 시스템 및 관리방법)는 게스트 메모리를 이용하는 복수의 가상 머신 중 어느 하나에서 게스트 메모리가 부족한 것을 감지하면, 나머지 가상 머신 중 어느 하나 이상에 할당된 게스트 메모리 중 일부를 게스트 메모리가 부족한 가상 머신에게 할당하는 가상 머신의 메모리 관리 시스템을 제공한다.

발명의 내용

해결하려는 과제

[0006] 본 발명은 전술한 종래 기술의 문제점을 해결하기 위한 것으로서, 본 발명의 일부 실시예는 가상화 시스템에 포함된 가상 머신간의 메모리 대역을 위한 스왑 캐시 할당 및 스왑 아웃 동작을 수행하도록 하는 메모리 관리 방법을 제공한다.

과제의 해결 수단

[0007] 상술한 기술적 과제를 달성하기 위한 기술적 수단으로서, 본 발명의 일 측면에 따른 가상화 시스템에서의 메모리 관리 방법은, 제 1 페이지 데이터가 기록된 제 2 가상 머신의 메모리에 대하여 제 2 페이지 데이터의 쓰기 동작시에 메모리 부족이 발생한 경우, 제 1 가상 머신의 메모리로부터 제 1 스왑 캐시를 대역시키는 단계; 상기 제 1 페이지 데이터 중 일부 또는 전체를 상기 대역된 제 1 스왑 캐시로 스왑 아웃시키는 단계; 및 상기 제 2 가상 머신의 메모리의 여유 공간에 상기 제 2 페이지 데이터를 저장시키는 단계를 포함한다.

[0008] 또한, 본 발명의 다른 측면에 따른 가상화 시스템에서의 메모리 관리 방법은 제 1 가상 머신이 제 2 가상 머신으로부터 스왑 캐시 할당 요청을 수신하는 단계; 상기 스왑 캐시 할당 요청에 포함된 대상 페이지 데이터와 제 1 가상 머신의 스왑 캐시에 기 저장된 페이지 데이터의 중복 여부를 판단하는 단계; 중복되는 페이지 데이터의 존재시에 해당 페이지 데이터를 저장하고 있는 스왑 캐시의 주소 정보를 상기 제 2 가상 머신으로 전송하는 단계 및 중복되는 페이지 데이터의 미 존재시에 새로운 스왑 캐시를 할당하고, 상기 대상 페이지 데이터를 상기 스왑 캐시에 저장시키는 단계를 포함한다.

발명의 효과

[0009] 전술한 본 발명의 과제 해결 수단에 의하면, 가상 머신 간의 스왑 캐시 대역을 통해 메모리 부족이 발생한 가상 머신에 대해서 메모리 부족을 효과적으로 해소할 수 있다. 특히, 디스크에 데이터를 저장하는 것이 아니고, 디스크 보다 읽기/쓰기 속도가 빠른 메모리에 데이터를 저장하기 때문에, 향후 스왑 캐시에 저장된 데이터를 다시 참조할 때에도 디스크에 저장하는 경우에 비하여 빠른 속도를 보장할 수 있는 효과가 있다.

도면의 간단한 설명

[0010] 도 1은 본원 발명의 일 실시예에 따른 가상화 시스템을 도시한 도면이다.

도 2는 본원 발명의 일 실시예에 따른 메모리 스왑 동작을 설명하기 위한 도면이다.

도 3은 본원 발명의 다른 실시예에 따른 메모리 스왑 동작을 설명하기 위한 도면이다.

도 4는 본원 발명의 다른 실시예에 따른 메모리 스왑 동작을 설명하기 위한 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0011] 아래에서는 첨부한 도면을 참조하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 본 발명의 실시예를 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.
- [0012] 명세서 전체에서, 어떤 부분이 다른 부분과 "연결"되어 있다고 할 때, 이는 "직접적으로 연결"되어 있는 경우뿐 아니라, 그 중간에 다른 소자를 사이에 두고 "전기적으로 연결"되어 있는 경우도 포함한다. 또한 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.
- [0013] 한편, 본 발명과 관련하여 다음과 같이 일부 용어를 정의한다.
- [0014] 페이지 (page)는 메모리를 일정한 크기로 나눈 후의 각 공간 조각을 의미하는 것으로, 예를 들어 일반적인 32 비트 컴퓨터에서는 4KB로 정의되어 있다.
- [0015] 가상 메모리 (virtual memory)는 물리적인 메모리의 제약을 벗어나 프로세스가 인식하고 있는 논리적인 메모리로서, 예를 들어 32 비트 컴퓨터에서는 하나의 프로세스가 4G의 메모리를 모두 사용하는 것으로 인식할 수 있다.
- [0016] 도 1은 본원 발명의 일 실시예에 따른 가상화 시스템을 도시한 도면이다.
- [0017] 가상화 시스템(100)은 복수의 가상 머신(110, 120, 130), 하이퍼바이저(140) 및 하드웨어(150)를 포함한다. 각각의 가상 머신(110, 120, 130)은 하이퍼바이저(140)에 의하여 생성 및 관리되는 것으로, 각 가상 머신에게 논리적으로 분리된 하드웨어 리소스를 제공한다. 예를 들면, 메모리, 하드 디스크, 그래픽 드라이버 등 각종 하드웨어 리소스를 논리적으로 분배하여 각 가상 머신에 할당할 수 있다. 또한, 하이퍼바이저(140)는 각 가상 머신(110, 120, 130)을 구동하는 운영체제 리소스를 제공할 수 있다. 이러한 하이퍼바이저(140)의 설정에 따라 각각의 가상 머신(110, 120, 130)의 상세 구성 옵션은 서로 상이할 수 있다.
- [0018] 하이퍼바이저(140)에 포함된 메모리 관리부(142)는 본원 발명의 내용에 따라 각 가상 머신(110, 120, 130)의 메모리 리소스를 관리하는 것으로, 특히 메모리 페이지 간의 스왑 동작시에 물리 메모리를 관리하는 동작을 수행한다. 일반적으로, 스왑(swap) 동작은 사용하지 않는 페이지를 디스크로 옮기고 사용할 페이지를 메모리로 가져오는 교체동작을 의미하는 것으로, 본 발명에서는 가상 머신의 메모리들 사이에서의 페이지를 교체하는 스왑 동작을 사용한다.
- [0019] 각각의 가상머신(110, 120, 130)은 각 가상머신에 할당된 메모리(112, 122, 132) 및 스왑 처리부(114, 124, 134)를 포함한다. 제 1 가상머신(110)에 포함된 스왑 처리부(114)를 기준으로 설명하면, 스왑 처리부(114)는 제 1 가상머신(110)의 메모리 상태를 모니터링하고, 여유 공간이 부족한 경우 다른 가상머신으로 스왑 캐시를 요청하는 동작을 수행한다. 아울러, 스왑 처리부(114)는 다른 가상머신으로부터 대여한 스왑 캐시를 관리하며, 특히 스왑 캐시의 주소 정보를 매칭하여 저장하도록 한다. 한편, 스왑 처리부(114)는 다른 가상머신으로부터 수신한 스왑 캐시 대여 요청에 따라 제 1 가상머신(110)의 메모리 상태를 확인하고, 메모리 여유 공간 존재 여부에 따라 스왑 캐시 대여 여부 판단, 스왑 캐시 대여 동작 등을 수행한다. 즉, 스왑 처리부(114, 124, 134)는 자신이 속한 가상 머신의 메모리를 모니터링 하여 타 가상 머신으로부터 스왑 캐시를 빌려오는 동작 뿐만 아니라, 타 가상 머신의 스왑 캐시 대여 요청에 응답하여 스왑 캐시를 빌려주는 동작을 처리할 수 있다.
- [0020] 하드웨어(150)는 가상화 시스템(100)에서 활용될 수 있는 각종 하드웨어 리소스를 포함하는 것으로, 예를 들면, 메모리, 하드 디스크, SSD(Solid state disk), 그래픽 드라이버, 각종 네트워크 드라이버를 포함할 수 있으나, 본 발명이 이에 한정되는 것은 아니다.
- [0021] 도 2는 본원 발명의 일 실시예에 따른 메모리 스왑 동작을 설명하기 위한 도면이다.
- [0022] 도 2의 (a)에 도시된 바와 같이, 제 2 가상 머신(VM2)의 메모리에는 운영체제(Kernel), 제 1 페이지 데이터(P1)가 저장되어 있고, 메모리의 상태에 따라 여유 공간이 존재할 수 있다. 다음으로, 도 2의 (b)에 도시된 바와 같이, 제 1 가상 머신(VM2)의 메모리에 추가적으로 제 2 페이지 데이터(P2)를 기록할 때 메모리의 여유 공간 부족이 부족한 경우가 발생할 수 있다.
- [0023] 이와 같은 경우에는, 도 2 (c), (b)에 도시된 바와 같이, 제 1 가상 머신(VM1)의 메모리로부터 스왑 캐시를 대

여한다. 그리고, 대여한 스왑 캐시에 제 1 페이지 데이터(P1) 전체 또는 일부를 스왑 아웃한다. 즉, 제 1 페이지 데이터(P1) 전체 또는 일부를 제 1 가상 머신(VM1)으로부터 대여한 스왑 캐시에 기록하면서, 해당 데이터는 제 2 가상 머신(VM2)의 메모리로부터 삭제하여, 제 2 가상 머신(VM2)의 메모리 여유 공간을 증가시킨다. 이때, 제 1 가상 머신(VM2)은 자신의 페이지 테이블에 대여한 스왑 캐시의 주소를 저장하여 마치 자신의 메모리인 것처럼 접근한다. 즉, 스왑 캐시의 소유권자는 제 1 가상 머신(VM1)이고, 제 2 가상 머신(VM2)이 이 공간을 대여하여 사용하는 것이다.

[0024] 한편, 제 1 페이지(P1) 중 스왑 아웃할 일부 데이터를 선택하는 경우에는, LRU(least recently used) 교체 방식에 따라 최근에 가장 사용 빈도가 낮은 데이터를 스왑 아웃할 데이터로 선택할 수 있다.

[0025] 한편, 스왑 아웃 동작시에 제 1 가상 머신(VM1)의 메모리에 기록된 데이터를 압축시켜, 스왑 캐시에 기록될 데이터의 용량을 축소시키는 동작을 수행할 수 있다. 예를 들면, LZ0(Lempel-Ziv-Oberhumer) 압축 알고리즘과 같은 무손실 압축 알고리즘을 사용할 수 있으나, 본원 발명이 이에 한정되는 것은 아니다.

[0026] 이와 같이, 다른 가상 머신의 메모리로부터 스왑 캐시를 대여하고, 기존에 기록된 데이터를 대여한 스왑 캐시로 스왑 아웃함으로써, 여유 메모리 공간을 확장시킨다.

[0027] 다음으로, 도 2 (e)에 도시된 바와 같이, 스왑 아웃 동작에 따라 발생한 여유 공간에 제 2 페이지 데이터(P2)를 기록한다.

[0028] 도 3은 본원 발명의 다른 실시예에 따른 메모리 스왑 동작을 설명하기 위한 도면이다.

[0029] 도 3은 총 2 개 이상의 가상 머신(VM2, VM3)이 하나의 가상 머신(VM1)의 메모리로부터 스왑 캐시를 대여하는 실시예를 도시하고 있다.

[0030] 먼저, 제 2 가상 머신(VM2)과 제 3 가상 머신(VM3)은 앞서 도 2에서 설명한 스왑 캐시 대여 동작 및 스왑 아웃 동작 등을 통해, 메모리 여유 공간을 확보하여 추가 데이터를 기록한 상태이다. 즉, 제 2 가상 머신(VM2)은 제 1 페이지 데이터(P1) 전체 또는 일부를 제 1 가상 머신(VM1)의 메모리로부터 대여한 제 1 스왑 캐시(310)로 스왑 아웃 시킨 후 생성된 여유 공간에 제 2 페이지 데이터(P2)를 기록하였다. 그리고, 제 3 가상 머신(VM3)은 제 3 페이지 데이터(P3) 전체 또는 일부를 제 1 가상 머신(VM1)의 메모리로부터 대여한 제 2 스왑 캐시(320)로 스왑 아웃 시킨 후 생성된 여유 공간에 제 4 페이지 데이터(P4)를 기록하였다.

[0031] 이때, 제 1 가상 머신(VM1)으로 스왑 아웃된 데이터가 동일한 경우, 즉, 제 1 스왑 캐시(310)에 저장된 데이터와 제 2 스왑 캐시(320)에 저장된 데이터가 동일한 경우에는 메모리 공간 사용의 최적화를 위하여 어느 하나의 스왑 캐시에 저장된 데이터만을 유지하도록 한다. 즉, 제 1 스왑 캐시(310) 및 제 2 스왑 캐시(320) 중 어느 하나에 저장된 데이터를 삭제하도록 한다. 그리고, 나머지 스왑 캐시를 제 2 가상 머신(VM2)의 메모리 및 제 3 가상 머신(VM3)의 메모리에 의하여 참조될 수 있도록 공유시킴으로써, 추후 각 가상 머신에 의한 참조 요청시에 해당 데이터가 참조될 수 있도록 한다.

[0032] 제 1 가상 머신(VM1)은 제 1 스왑 캐시(310)와 제 2 스왑 캐시(320)의 소유권자로서, 각의 스왑 캐시에 저장된 스왑 아웃 데이터들에 접근할 수 있다. 따라서, 제 1 가상 머신(VM1)은 스왑 캐시 내의 모든 페이지들의 내용을 비교하여 중복된 내용을 가진 스왑 아웃 데이터들을 제거할 수 있다.

[0033] 이때, 제 1 가상 머신(VM1)이 이미 저장된 스왑-아웃 페이지들을 제거하여 하나의 페이지로 공유하는 작업을 주기적으로 수행하는 것은 매우 큰 오버헤드를 유발할 수 있다. 이는 기존의 스왑-아웃 페이지의 저장소 주소를 변경하게 되어, 제 2 가상 머신(VM2)과 제 3 가상 머신(VM3)이 자체적으로 보유하고 있는 스왑 테이블 내의 주소 정보도 동시에 변경되어야 하기 때문이다. 이를 해결하기 위해, 본 발명은 제 2 가상 머신(VM2) 또는 제 3 가상 머신(VM3)이 새로운 스왑-아웃 페이지를 저장하기 위한 공간 요청을 보내면, 제 1 가상 머신(VM1)이 요청을 받은 후에 해당 페이지의 중복 여부를 판단하도록 한다. 만약 페이지의 중복이 있으면 공유할 페이지의 주소만 알려주고, 스왑 캐시 내의 새로운 공간을 할당하지 않도록 한다. 도면을 통해 더욱 구체적인 동작을 살펴볼 수 있다.

[0034] 도 4는 본원 발명의 다른 실시예에 따른 메모리 스왑 동작을 설명하기 위한 도면이다.

[0035] 먼저, 제 2 가상 머신(VM2)으로부터 제 1 가상 머신(VM1)이 스왑 아웃 요청을 수신한다(S410). 즉, 제 2 가상 머신(VM2)이 메모리 부족을 확인하고, 메모리에서 내보낼 페이지를 선택한다. 그리고, 선택한 페이지를 압축하고, 미리 설정된 키(key)값을 가지고 해쉬 알고리즘을 실행하여 해쉬값을 산출한다. 스왑 아웃 요청시에는 압

축된 페이지 데이터, 해쉬값, 키값등을 전송할 수 있다.

- [0036] 스왑 아웃 요청을 수신한 제 1 가상 머신(VM1)은 해쉬값 등에 기초하여 기존에 저장된 스왑 아웃 페이지와의 동일 여부를 판단한다(S420). 즉, 제 1 가상 머신(VM1)은 각 스왑 아웃 페이지의 해쉬값을 저장한 해쉬 테이블을 보유한다. 그리고, 해쉬 테이블에 저장된 해쉬값에 기초하여 새롭게 요청된 스왑 아웃 대상 데이터와 기존에 저장된 스왑 아웃 페이지의 중복 여부를 판단할 수 있다.
- [0037] 만약, 동일한 해쉬값을 가진 스왑 아웃 페이지가 있다면, 해당 스왑 아웃 페이지가 저장된 물리 주소를 제 2 가상 머신(VM2)에 전송한다(S440). 이때, 제 1 가상 머신(VM1)은 데이터의 정확한 비교를 위하여, 제 2 가상 머신(VM2)이 전송한 스왑 아웃 대상 페이지와 제 1 가상 머신(VM1)에 저장된 스왑 아웃 페이지의 내용을 비트 레벨에서 추가적으로 비교할 수 있다. 즉, 해쉬값이 동일한 것으로 판단된 스왑 아웃 페이지에 대하여, 추가적으로 비트 레벨의 비교를 수행하는 것이다. 이와 같은, 비교를 통해 중복 내용이 이미 제 1 가상 머신(VM1)의 스왑 캐시에 저장된 경우에는 제 2 가상 머신(VM2)에 대하여 해당 페이지의 물리 주소를 전송하고, 제 2 가상 머신(VM2)은 해당 물리 주소를 주소 테이블에 저장하도록 한다. 예를 들면, 제 2 가상 머신(VM2)은 스왑 아웃하려는 스왑 아웃 대상 페이지의 주소와 제 1 가상 머신(VM1)으로부터 수신한 물리 주소(즉 중복된 데이터를 포함한 스왑 아웃 페이지의 주소)를 매칭하여 주소 테이블에 저장한다. 그리고, 제 2 가상 머신(VM2)의 메모리로부터는 스왑 아웃 대상 페이지의 내용을 삭제하여, 메모리의 여유 공간을 확보한다. 이렇게 하여, 제 2 가상 머신(VM2)이 요청한 스왑 아웃 대상 페이지와 동일한 내용이 제 1 가상 머신(VM1)의 스왑 캐시에 이미 존재하는 경우에는 스왑 캐시를 추가적으로 할당하지 않고, 동일한 내용이 저장된 스왑 페이지의 주소만을 매칭시키는 형태로 스왑 아웃 동작이 완료된다.
- [0038] 다음으로, 비교 결과, 기존에 저장된 스왑 아웃 페이지와 동일하지 않은 것으로 판단된 경우에는, 도 2 를 통해 설명한 스왑 캐시 할당(S450) 및 스왑 아웃 동작(S460)을 수행한다.
- [0039] 즉, 앞서 산출한 해쉬값이 동일하지 않거나, 해쉬값이 동일하더라도 비트 레벨 비교 결과 내용이 상이한 것으로 판단되면, 새롭게 스왑 캐시를 할당하도록 한다. 제 1 가상 머신(VM1)의 메모리에 할당된 스왑 캐시의 주소 정보를 제 2 가상 머신(VM2)과 공유하도록 GRANT 메커니즘을 활용할 수 있다. 제 2 가상 머신(VM2)은 제 1 가상 머신(VM1)으로부터 수신한 주소 정보를 이용하여, 해당 메모리 공간을 하이퍼바이저(140)에게 요청하고, 이를 자신의 주소 공간에 추가한다. 그리고, 제 1 가상 머신(VM1)으로부터 할당된 스왑 캐시에 스왑 아웃 페이지 데이터를 압축하여 저장하고, 자신의 주소 테이블에 해당 페이지의 주소 정보를 기록한다. 이때, 압축 데이터의 저장시에 해쉬값을 산출하여 같이 저장되도록 함으로써, 추후 중복 데이터 검색시에 활용하도록 한다.
- [0040] 참고로, 본 발명의 실시예에 따른 도 1에 도시된 구성 요소들은 소프트웨어 또는 FPGA(Field Programmable Gate Array) 또는 ASIC(Application Specific Integrated Circuit)와 같은 하드웨어 구성 요소를 의미하며, 소정의 역할들을 수행한다.
- [0041] 그렇지만 '구성 요소들'은 소프트웨어 또는 하드웨어에 한정되는 의미는 아니며, 각 구성 요소는 어드레싱할 수 있는 저장 매체에 있도록 구성될 수도 있고 하나 또는 그 이상의 프로세서들을 재생시키도록 구성될 수도 있다.
- [0042] 따라서, 일 예로서 구성 요소는 소프트웨어 구성 요소들, 객체지향 소프트웨어 구성 요소들, 클래스 구성 요소들 및 태스크 구성 요소들과 같은 구성 요소들과, 프로세스들, 함수들, 속성들, 프로시저들, 서브루틴들, 프로그램 코드의 세그먼트들, 드라이버들, 펌웨어, 마이크로 코드, 회로, 데이터, 데이터베이스, 데이터 구조들, 테이블들, 어레이들 및 변수들을 포함한다.
- [0043] 구성 요소들과 해당 구성 요소들 안에서 제공되는 기능은 더 작은 수의 구성 요소들로 결합되거나 추가적인 구성 요소들로 더 분리될 수 있다.
- [0044] 본 발명의 일 실시예는 컴퓨터에 의해 실행되는 프로그램 모듈과 같은 컴퓨터에 의해 실행가능한 명령어를 포함하는 기록 매체의 형태로도 구현될 수 있다. 컴퓨터 판독 가능 매체는 컴퓨터에 의해 액세스될 수 있는 임의의 가용 매체일 수 있고, 휘발성 및 비휘발성 매체, 분리형 및 비분리형 매체를 모두 포함한다. 또한, 컴퓨터 판독 가능 매체는 컴퓨터 저장 매체 및 통신 매체를 모두 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함한다. 통신 매체는 전형적으로 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈, 또는 반송파와 같은 변조된 데이터 신호의 기타 데이터, 또는 기타 전송 메커니즘을 포함하며, 임의의 정보 전달 매체를 포함한다.
- [0045] 본 발명의 방법 및 시스템은 특정 실시예와 관련하여 설명되었지만, 그것들의 구성 요소 또는 동작의 일부 또는

전부는 범용 하드웨어 아키텍처를 갖는 컴퓨터 시스템을 사용하여 구현될 수 있다.

[0046] 전술한 본 발명의 설명은 예시를 위한 것이며, 본 발명이 속하는 기술분야의 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 쉽게 변형이 가능하다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다. 예를 들어, 단일형으로 설명되어 있는 각 구성 요소는 분산되어 실시될 수도 있으며, 마찬가지로 분산된 것으로 설명되어 있는 구성 요소들도 결합된 형태로 실시될 수 있다.

[0047] 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

부호의 설명

[0048] 100: 가상화 시스템

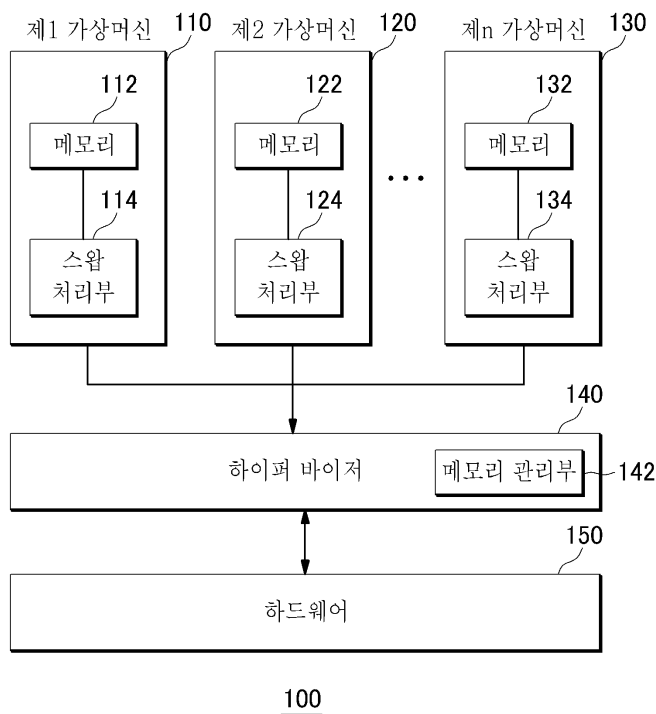
140: 하이퍼바이저

142: 메모리 관리부

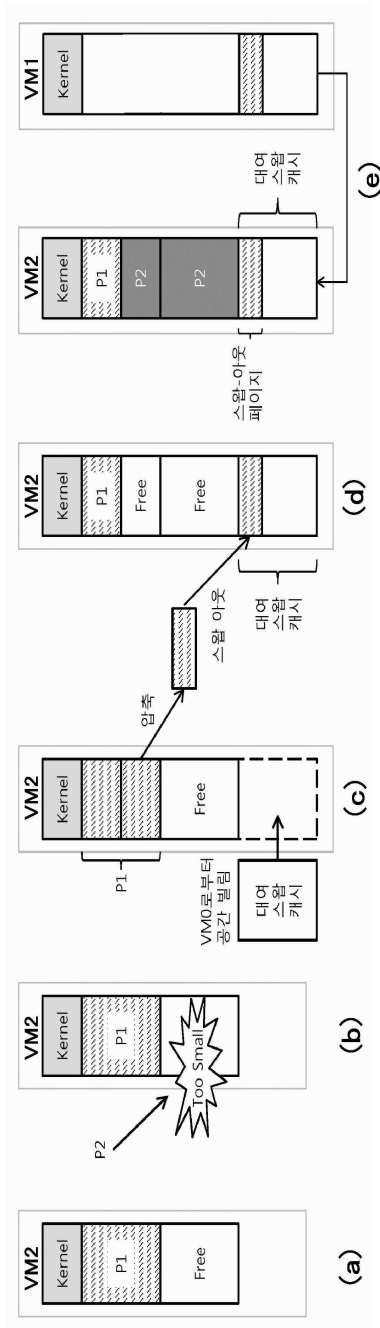
150: 하드웨어

도면

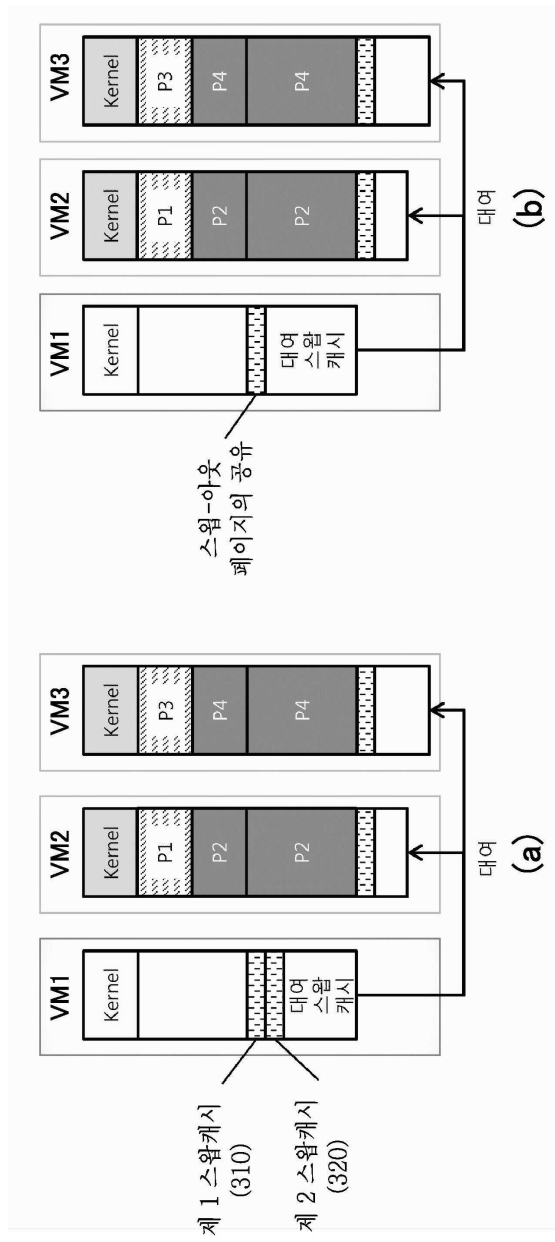
도면1



도면2



도면3



도면4

