



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2007-0100367
 (43) 공개일자 2007년10월10일

- | | |
|---|--|
| (51) Int. Cl.
G06F 9/50 (2006.01) G06F 15/167 (2006.01)
G06F 9/455 (2006.01)
(21) 출원번호 10-2007-7018903
(22) 출원일자 2007년08월17일
심사청구일자 2007년08월17일
번역문제출일자 2007년08월17일
(86) 국제출원번호 PCT/US2006/003320
국제출원일자 2006년01월30일
(87) 국제공개번호 WO 2006/088637
국제공개일자 2006년08월24일
(30) 우선권주장
11/062,202 2005년02월17일 미국(US) | (71) 출원인
인텔 코오퍼레이션
미합중국 캘리포니아 산타클라라 미션 칼리지 블러바드 2200
(72) 발명자
맨골드, 리차드
미국 97116 오레곤주 포레스트 그로브 노쓰웨스트 캔자스 시티로드 7155
(74) 대리인
양영준, 백만기 |
|---|--|

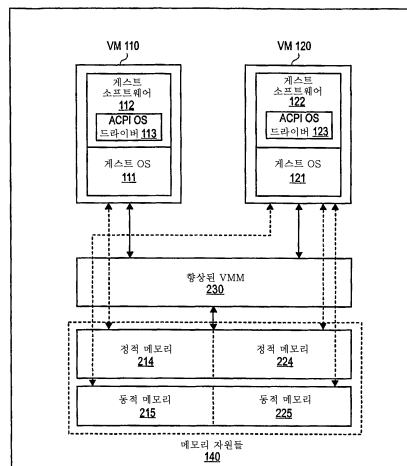
전체 청구항 수 : 총 20 항

(54) 하나의 가상 머신에서 다른 가상 머신으로 메모리를동적으로 재할당하기 위한 방법, 장치 및 시스템

(57) 요약

VMM(virtual machine manager)가 메모리를 하나의 VM(virtual machin)에서 다른 VM으로 동적으로 재할당할 수 있게 하는 방법, 장치 및 시스템이 제공된다. VMM은 현재 메모리가 할당된 VM에 대해 메시지를 생성하고 그 디바이스에 메모리가 중단(shut down)될 것임을 알릴 수 있다. 그후에 현재의 VM은 메모리의 내용들을 호스트 하드 디스크에 복사하고 그 메모리를 추출할 수 있다. 그 후 VMM은 다른 VM에 메모리가 사용가능함을 알릴 수 있고, 제2 VM은 그 후 그 메모리를 그의 사용가능한 메모리 자원들에 추가할 수 있다.

대표도 - 도3



특허청구의 범위

청구항 1

메모리를 제1 가상 머신(virtual machine)("VM")에서 제2 VM으로 동적으로 재할당하기 위한 방법으로서,
 상기 제1 VM에 상기 메모리가 제거되었음을 통지하는 단계;
 상기 제1 VM이 상기 메모리에 중단(shutdown) 명령을 발행하도록 하는 단계;
 상기 중단 명령을 가로채는 단계; 및
 제2 VM에 상기 메모리가 사용가능함을 통지하는 단계
 를 포함하는 방법.

청구항 2

제1항에 있어서,
 상기 제1 VM에 상기 메모리가 제거되었음을 통지하는 단계는,
 상기 메모리 대신 상기 제1 VM에 대해 제1 메시지를 생성하는 단계;
 상기 제1 메시지의 원인과 관련해서 상기 제1 VM으로부터 제1 조회(inquiry)를 가로채는 단계; 및
 제1 조회에 응답하여 상기 제1 VM에 할당된 상기 메모리가 중단되는 것을 상기 제1 VM에 알리는 단계
 를 더 포함하는 방법.

청구항 3

제2항에 있어서,
 상기 제1 VM이 상기 메모리를 추출하는 명령을 발행하도록 하는 단계를 더 포함하는 방법.

청구항 4

제1항에 있어서,
 상기 제1 VM에 상기 메모리가 제거되었음을 통지하는 단계는,
 ACPI(Advanced Configuration and Power Interface) 프로토콜에 따라 상기 제1 VM에 상기 메모리가 제거되었음을 통지하는 단계를 더 포함하는 방법.

청구항 5

제1항에 있어서,
 상기 제2 VM에 상기 메모리가 사용가능함을 통지하는 단계는,
 상기 제2 VM에 상기 메모리를 할당하는 단계;
 상기 제2 VM에 대해 제2 메시지를 생성하는 단계;
 상기 제2 메시지의 원인과 관련해서 상기 제2 VM으로부터 제2 조회를 가로채는 단계; 및
 상기 제2 조회에 응답하여 상기 제2 VM에 상기 메모리가 사용가능함을 알리는 단계
 를 더 포함하는 방법.

청구항 6

제5항에 있어서,
 상기 제2 VM에 상기 메모리가 사용가능함을 통지하는 단계는 ACPI 프로토콜에 따라 상기 제2 VM에 통지하는 단계를 더 포함하는 방법.

청구항 7

제5항에 있어서,

상기 제2 VM에 의해 발행된 구성 조회(configuration inquiry)들을 가로채는 단계를 더 포함하는 방법.

청구항 8

제1항에 있어서,

상기 메모리를 상기 제1 VM에서 상기 제2 VM으로 재할당하라는 사용자 요구를 수신하는 단계를 더 포함하는 방법.

청구항 9

제1항에 있어서,

상기 메모리를 상기 제1 VM에서 상기 제2 VM으로 재할당하는 단계는 소정의 할당 정책에 근거하는 방법.

청구항 10

메모리를 동적으로 재할당할 수 있는 호스트 컴퓨터 시스템으로서,

모니터링 모듈;

상기 모니터링 모듈에 연결된 제1 컴퓨터 시스템;

상기 모니터링 모듈에 연결된 제2 컴퓨터 시스템; 및

상기 모니터링 모듈에 연결된 물리적 디바이스

를 포함하고,

상기 모니터링 모듈은 상기 제1 컴퓨터 시스템에 상기 메모리가 제거되었음을 알림으로써 상기 메모리를 상기 제1 컴퓨터 시스템에서 상기 제2 컴퓨터 시스템으로 동적으로 재할당할 수 있는 호스트 컴퓨터 시스템.

청구항 11

제10항에 있어서,

상기 모니터링 모듈은,

상기 제1 컴퓨터 시스템에 대해 메시지를 생성함으로써 상기 제1 컴퓨터 시스템에 상기 메모리가 제거되었음을 알릴 수 있는 호스트 컴퓨터 시스템.

청구항 12

제11항에 있어서,

상기 모니터링 모듈은 상기 제1 컴퓨터 시스템에 의해 상기 메모리로 발행된 메시지들을 가로챌 수 있는 호스트 컴퓨터 시스템.

청구항 13

제10항에 있어서,

상기 모니터링 모듈은 또한 상기 제2 컴퓨터 시스템에 상기 메모리를 재할당하고 상기 제2 컴퓨터 시스템에 상기 메모리가 사용가능함을 알릴 수 있는 호스트 컴퓨터 시스템.

청구항 14

제10항에 있어서,

상기 제1 컴퓨터 시스템 및 상기 제2 컴퓨터 시스템은 호스트 컴퓨터 상의 VM들인 호스트 컴퓨터 시스템.

청구항 15

명령어들을 저장한 머신 액세스 가능한 매체를 포함하는 물품으로서,
 상기 명령어들은 머신에 의해 실행되는 경우,
 제1 VM에 메모리가 제거되었음을 통지하고;
 상기 제1 VM이 중단 명령어를 상기 메모리에 발행하도록 하고;
 상기 중단 명령어를 가로채고;
 제2 VM에 상기 메모리가 사용가능함을 통지하는 것
 에 의해,
 상기 머신이 상기 메모리를 상기 제1 VM에서 상기 제2 VM으로 동적으로 재할당하도록 하는 물품.

청구항 16

제15항에 있어서,
 상기 명령어들은 상기 머신에 의해 실행되는 경우, 또한
 상기 메모리 대신 상기 제1 VM에 대해 제1 메시지를 생성하고;
 상기 제1 메시지의 원인과 관련해서 상기 제1 VM으로부터 제1 조회를 가로채고;
 상기 제1 조회에 응답하여 상기 제1 VM에 할당된 상기 메모리가 중단되는 것을 상기 제1 VM에 알리는 것
 에 의해,
 상기 머신이 상기 제1 VM에 상기 메모리가 제거되었음을 통지하도록 하는 물품.

청구항 17

제16항에 있어서,
 상기 명령어들은 상기 머신에 의해 실행되는 경우, 또한 상기 머신이 상기 제1 VM에게 상기 메모리를 추출하는
 명령어를 발행하도록 하게 하는 물품.

청구항 18

제15항에 있어서,
 상기 명령어들은 상기 머신에 의해 실행되는 경우, 또한 상기 머신이 ACPI 프로토콜에 따라 상기 제1 VM에 상기
 메모리가 제거되었음을 통지하도록 하는 물품.

청구항 19

제15항에 있어서,
 상기 명령어들은 상기 머신에 의해 실행되는 경우, 또한
 상기 제2 VM에 상기 디바이스를 할당하고;
 상기 제2 VM에 대해 제2 메시지를 생성하고;
 상기 제2 메시지의 원인과 관련해서 상기 제2 VM으로부터 제2 조회를 가로채고;
 상기 제2 조회에 응답하여 상기 제2 VM에 상기 메모리가 사용가능함을 알리는 것
 에 의해,
 상기 머신이 상기 제2 VM에 상기 메모리가 사용가능함을 통지하도록 하는 물품.

청구항 20

제19항에 있어서,

상기 명령어들은 상기 머신에 의해 실행되는 경우, 또한 상기 머신이 ACPI 프로토콜에 따라 상기 제2 VM에 상기 메모리가 사용가능함을 통지하도록 하는 물품.

명세서

배경 기술

<1> 프로세서 기술이 발전함에 따라 가상화 기술(virtualization technology)에 대한 관심이 꾸준히 증가하고 있다. 가상화 기술의 하나의 양상은 가상 머신 모니터(virtual machine monitor)("VMM")를 실행하는 단일 호스트 컴퓨터가 복수의 호스트의 뷰 및/또는 추상화를 제공하여 호스트의 기반 하드웨어가 독립적으로 동작하는 하나 또는 그보다 많은 가상 머신(virtual machines)("VMs")처럼 보이게 하는 것을 가능하게 한다. 각 VM은 자신의 운영 체제("OS") 및/또는 소프트웨어 어플리케이션(들)을 실행하는 일체 완비된(self-contained) 플랫폼으로서 기능할 수 있다. VMM은 호스트 상의 자원들의 할당을 관리하고 필요에 따라 콘텍스트 전환(context switching)을 수행하여 라운드-로빈 또는 다른 소정의 방식에 따라 다양한 가상 머신 사이를 순환한다.

발명의 상세한 설명

<7> 본 발명의 실시예들은 가상 머신(들) 상의 운영 체제들을 재부팅시키지 않고 하나의 가상 머신에서 다른 가상 머신으로 자원들을 동적으로 재할당하기 위한 방법, 장치, 및 시스템을 제공한다. 명세서에서 본 발명의 "하나의 실시예" 또는 "일 실시예"에 대한 언급은 그 실시예와 관련하여 기술된 특정한 특징, 구조 또는 특성이 본 발명의 적어도 하나의 실시예에 포함된다는 것을 의미한다. 그리하여, 명세서 전반에 걸친 다양한 장소에서의 "하나의 실시예에서", "하나의 실시예에 따라" 또는 그와 같은 표현의 출현은 모두 반드시 동일한 실시예를 가리키는 것은 아니다.

<8> 도 1은 통상적인 가상 머신 호스트 플랫폼의 예("호스트(100)")를 예시한다. 전술된 바와 같이, 가상 머신 모니터("VMM(130)")는 통상적으로 호스트 플랫폼 상에서 실행되고 다른 소프트웨어에 대해 ("가상 머신들" 또는 "VM들"이라고도 하는) 플랫폼의 추상화(들) 및/또는 뷰(들)을 제공한다. 단지 두 개의 VM 파티션이 예시되었지만("VM(100)" 및 "VM(120)", 이후 집합적으로 "VM들"이라고 함), 이러한 VM들은 단순히 예시적인 것이고 추가적인 가상 머신들이 호스트에 추가될 수 있다. VMM(130)은 소프트웨어(예를 들면, 독립형 프로그램 및/또는 호스트 운영 체제의 컴포넌트), 하드웨어, 펌웨어 및/또는 그들의 임의의 조합으로 구현될 수 있다.

<9> VM(110) 및 VM(120)은 각각 그들 자신의 "게스트 운영 체제"(즉, "게스트 OS(111)" 및 "게스트 OS(121)"로서 예시된, VMM(130)에 의해 호스팅된 운영 체제들, 이후 집합적으로 "게스트 OS"라고 함) 및 다른 소프트웨어("게스트 소프트웨어(112)" 및 "게스트 소프트웨어(122)"로서 예시됨. 이후 집합적으로 "게스트 소프트웨어"라고 함)를 실행하는 일체 완비된 플랫폼들로서 기능할 수 있다. 각 게스트 OS 및/또는 게스트 소프트웨어는 가상 머신보다는 오히려 전용 컴퓨터 상에서 실행되는 것처럼 동작한다. 즉, 각 게스트 OS 및/또는 게스트 소프트웨어는 다양한 이벤트를 제어하고 호스트(100) 상의 하드웨어 자원들에 액세스하도록 기대될 수 있다. 실제로, VMM(130)은 이벤트들 및 하드웨어 자원들을 궁극적으로 제어하고 자원들을 그 자신의 정책들에 따라 가상 머신들에 할당한다.

<10> 도 1의 각 VM은 통상적으로 메모리를 모니터링하고 및/또는 동적으로 재할당하기 위한 ACPI(Advanced Configuration & Power Interface) 드라이버("ACPI OS 드라이버(113)" 및 "ACPI OS 드라이버(123)")를 포함한다. ACPI(예를 들면, 2002년 10월 11일 개정판 2.0b)는 플랫폼 구성 및 전원 관리 방식에 대한 공개된 공업 표준 규격이다. ACPI 드라이버들은 현재 존재하고 본 기술분야의 보통의 숙련자들에게 공지되어 있다. 이러한 드라이버들은 가상 호스트들 상의 VMM 및 VM들 사이에서 통상적인 ACPI 상호작용을 가능하게 하는 데 사용된다. 이하의 기술은 ACPI 프로토콜의 사용을 가정하고 있지만, 본 발명의 실시예들의 사상에서 벗어나지 않고 다른 구성 프로토콜들이 이용될 수도 있다.

<11> 다양한 메모리 자원들이 호스트(100)에 대해 사용가능할 수 있다(도 1에서 메모리 자원들(140)로서 집합적으로 예시되었고, 메모리 자원들의 한 부분이 VM(110)에 할당되는 한편 다른 부분은 VM(120)에 할당될 수 있다). 호스트(100) 상의 다양한 VM들에 대한 메모리 자원들의 할당은 VMM(130)에 의해 관리된다. 통상적으로, VMM(130)은 VM들이 인스턴스화될(instantiated) 때 VM들에게 메모리 자원들을 할당한다. 이러한 자원들을 재할당하여 새로운 VM을 추가하는 기존의 방식들은 통상적으로 번거롭다. 예를 들면, VMM(130)은 호스트(100) 상의 VM들을

중단(shutdown)할 수 있고, 그 후 모든 VM(원래의 VM 및 새로운 VM)을 재할당된 자원들과 함께 재시작할 수 있다. 이러한 방식은 다양한 VM의 게스트 OS가 VM 초기화 프로세스의 일부로서 메모리 자원들에서의 변경을 검출할 수 있도록 한다. 그러나 그 방식은 자원들의 임의의 유형의 동적 재할당을 가능하게 하고 새로운 VM의 인스턴스화를 가능하게 하기 위해 호스트(100) 상의 활성 VM들이 필수적으로 "재부팅되도록" 요구한다.

- <12> 대안적으로, 독점 소프트웨어(예를 들면, 도 1의 VM(110)에서 "소프트웨어 드라이버(150)"로서 개념적으로 예시된 소프트웨어 드라이버)가 메모리 자원들(140)의 재할당을 다루기 위해 호스트(100) 상의 각각의 VM들에 추가될 수 있다. 소프트웨어 드라이버(150)는 하나의 VM으로부터 메모리 자원들을 효과적으로 제거하고 VMM(130)이 이러한 자원들을 다른 VM에 재할당할 수 있게 함으로써 메모리 자원들(140)을 재할당할 수 있다. 상이한 유형 및/또는 버전의 운영 체제를 위해 복수의 소프트웨어 드라이버가 생성되고 유지되어야 할 수 있다. VM들에 소프트웨어 드라이버를 추가하는 것은 통상적으로 상당한 양의 코드를 VMM(130)에 추가하는 것을 필요로 한다. 게다가, 이러한 드라이버들은 소프트웨어 드라이버와 VMM(130) 사이에 독점 인터페이스를 필요로 할 수도 있다. 마침내, 이러한 방식은 유지하기 어렵고 VMM(130)에 대한 안정성 문제들을 초래하여 호스트(100)의 성능에 악영향을 줄 수 있다.
- <13> 본 발명의 실시예들은 가상화된 호스트 상의 메모리 자원들의 동적 재할당을 가능하게 한다. 더 구체적으로, 본 발명의 일 실시예에서, 메모리 자원들이 호스트(100) 상의 VM들을 "재부팅"할 필요없이, 및 추가적인 소프트웨어 없이 재할당될 수 있다. 도 2는 본 발명의 일 실시예를 더 자세하게 예시한다. 예시된 바와 같이, 향상된 VMM(230)은 다양한 VM 상의 ACPI OS 드라이버(113) 및 ACPI OS 드라이버(123)와 상호작용하여 VM들에 소프트웨어를 추가할 필요를 회피하면서 메모리를 모니터링 및/또는 동적으로 재할당할 수 있다. 본 발명의 실시예들의 향상된 VMM(230)은 아래에서 훨씬 자세하게 기술된 것과 같이 ACPI 드라이버들을 이용하여 호스트(100) 상의 메모리를 동적으로 재할당할 수 있다. 향상된 VMM(230)이 기존의 VMM 및/또는 기존의 VMM과 함께 동작할 수 있는 다른 요소들에 만들어진 증진들을 포함할 수 있다는 것이 본 기술분야의 보통의 숙련자들에게 곧 명백할 것이다. 따라서 향상된 VMM(230)은 소프트웨어(예를 들면, 독립형 프로그램 및/또는 호스트 운영 체제의 컴포넌트), 하드웨어, 펌웨어 및/또는 그들의 임의의 조합으로 구현될 수 있다.
- <14> 메모리 자원들(140)은 "정적" 부분 및 "동적" 부분을 포함할 수 있다. 하나의 실시예에서, 도 2에서 예시된 바와 같이, 메모리 자원들("정적 메모리(214)" 및 "정적 메모리(224)")(140)의 한 부분이 각 VM에 대해 전용으로 되는 한편 메모리 자원들(140)의 다른 부분은 VM(110)과 VM(120) 사이에서 동적으로 할당 및/또는 공유될 수 있다. 대안적인 실시예들에서, 모든 메모리 자원(140)이 VM(110)과 VM(120)에 의해 공유될 수 있다. 즉, VM들이 각자에게 전용으로 된 메모리의 정적 부분을 갖지 않고, 대신 적당한 메모리량이 각각에 동적으로 할당될 수 있다. 설명의 목적들을 위해, 앞의 가정(즉, 메모리의 정적 부분 및 동적 부분)이 아래에서 사용된다. 이 실시예에서, (도 2에서 동적 메모리(215)가 VM(110)에 할당되고 동적 메모리(225)가 VM(120)에 할당된 것과 같이) 동적 메모리의 일부분이 초기에 각 VM에 할당될 수 있지만, 이러한 부분들은 언제라도 동적으로 제거 및/또는 추가될 수 있다. 본 발명의 일 실시예에 따르면, 향상된 VMM(230)이 메모리 자원들이 재할당되어야 한다고 결정할 수 있다. 이 결정은 향상된 VMM(230)에 제공된 기준에 근거하여 자동적으로 이루어지고, 및/또는 VM으로부터의 추가적인 자원들에 대한 요구에 응답하여 이루어질 수 있다. 이 예의 목적들을 위해, 자원들이 VM(110)으로부터 제거되어 VM(120)으로 재할당된다고 가정한다.
- <15> 자원들을 재할당한다는 결정을 할 때, 향상된 VMM(230)은 VM(110)에 대해 ACPI 범용 이벤트("GPE")를 생성할 수 있다. 하나의 실시예에서, 향상된 VMM(230)에 의해 생성된 ACPI 이벤트는 호스트(100)의 하드웨어에 의해 생성되고 및/또는 다루어지기 보다는 오히려 소프트웨어로 에뮬레이트될 수 있다. GPE의 수신시, VM(110)의 게스트 OS(111)는 ACPI 이벤트 상태 레지스터를 판독하고 및/또는 다른 동작들(예를 들면, 호스트 버스의 구성 레지스터들에 속하는 조회들(이후, "구성 조회들")을 생성함)을 수행하여 GPE의 목적을 판단할 수 있다. 향상된 VMM(130)은 이러한 동작들을 가로채고 VM(110)에 동적 메모리(215)가 제거됨을 알릴 수 있다. 결국, 메모리가 실제로 "제거"되지 않더라도, VM(110)에는 그렇게 보일 것이다. 이러한 정보의 수신시, 게스트 OS(111)는 메모리의 임의의 현재의 정보를 호스트(100)의 하드디스크로 교체(swap)할 수 있고 그 후 동적 메모리(215)를 추출할 수 있다. 즉, 게스트 OS(111)가 동적 메모리(215)에 메시지를 보내 그것이 중단 및/또는 제거될 것임을 알릴 수 있다.
- <16> 실제로 동적 메모리(215)가 사실상 중단되지 않기 때문에, 향상된 VMM(230)은 VM(110)로부터 동적 메모리(215)로의 메시지를 가로챈다. 그 후, 동적 메모리(215)는 다른 VM에 재할당되는 데 사용가능할 수 있다. 향상된 VMM(230)은 이제 동적 메모리(215)를 호스트(100) 상의 다른 VM, 예를 들면 (도 3에 도시된 바와 같이) VM(120)에 재할당할 수 있다. 구체적으로, 하나의 실시예에서, 향상된 VMM(230)은 에뮬레이트된 ACPI GPE를 다

시, 이번에는 VM(120)에 대해 생성할 수 있다. VM(120)의 게스트 OS(121)는 ACPI 이벤트 상태 레지스터를 관독할 수 있고 및/또는 다른 동작들을 수행하여 GPE에 대한 이유를 판단할 수 있다. 다시, 향상된 VMM(230)은 이러한 동작들을 가로챌 수 있고 VM(120)에 동적 메모리(215)가 사용가능함을 알릴 수 있다. 일 실시예에서, 향상된 VMM(230)은 게스트 VM(120)의 메모리 공간에 (ACPI 규격에 의해 규정된 바와 같이) 디바이스 표들을 만듦으로써 VM(120)에 알릴 수 있다. 이 정보의 수신시, ACPI OS 드라이버(123)과 함께 게스트 OS(121)는 VM(120)에 대해 사용가능한 메모리 자원들에 동적 메모리(215)를 추가할 수 있고(예를 들면, 메모리를 페이지 테이블들 등에 추가한다) 그 후 그 디바이스가 다른 VM에 의해 요구되고 및/또는 향상된 VMM(230)이 동적 메모리(215)를 재할당하기로 결정할 때까지 그 메모리에 대한 배타적 액세스를 갖는다. 게스트 OS(121) 및 ACPI OS 드라이버(123)가 VM(120)에 메모리를 추가하는 방법의 상세는 본 기술 분야의 보통의 숙련자들에게 공지되어 있으므로 그것의 더 자세한 설명은 본 명세서에서 생략되었다.

<17> 그리하여 본 발명의 실시예들은 게스트 OS(111) 및 게스트 OS(121)를 재부팅할 필요없이, 및 추가적인 소프트웨어를 필요로 하지 않고 향상된 VMM(230)이 하나의 VM으로부터 다른 VM으로 메모리를 동적으로 재할당하는 것을 가능하게 한다. 필요에 따라 메모리 자원들을 동적으로 재할당하는 능력은 VMM(230)이 (예를 들면, 메모리 자원들이 효과적으로 재할당되는 것을 보증함에 의해) 각 VM의 성능을 최적화할 수 있게 하기 때문에 호스트(100) 상에 더 많은 VM이 인스턴스화됨에 따라 이러한 유연성은 점점 가치있게 된다. 도 4는 본 발명의 일 실시예의 개요를 도시하는 흐름도이다. 이하의 동작들이 순차적인 프로세스로서 기술되었지만, 사실상 동작들 중 다수가 병렬적으로 및/또는 동시발생적으로 수행될 수 있다. 더욱이, 동작들의 순서는 본 발명의 실시예들의 사상에서 벗어나지 않고 재배열될 수 있다. 401에서, 향상된 VMM(230)이 요구들을 수신하고 그리고/또는 동적 메모리(215)를 재할당하도록 결정할 수 있다. 402에서 향상된 VMM(230)은 현재 전용의 동적 메모리(215)를 갖는 VM(110)에 대해 ACPI GPE를 생성할 수 있다. 상술된 바와 같이, 본 발명의 실시예들이 본 명세서에서 ACPI에 관해 기술되었지만, 다른 인터페이스들 및/또는 프로토콜들이 동일한 효과를 달성하기 위해 본 발명의 실시예들의 사상에서 벗어나지 않고 사용될 수 있다. 403에서, VM(110)의 게스트 OS(111)가 ACPI 이벤트 상태 레지스터를 관독하고 및/또는 다른 동작들을 수행하여 GPE의 이유를 판단할 수 있다. 404에서 이러한 동작들은 향상된 VMM(230)에 의해 가로채질 수 있고, 향상된 VMM(230)은 VM(110)에 동적 메모리(215)가 중단될 것임을 알릴 수 있다. 그 후 405에서 게스트 OS(111)가 동적 메모리(215)의 정보를 호스트(100)의 하드 디스크로 교체하고 동적 메모리(215)를 추출할 수 있다. 406에서, 향상된 VMM(230)은 VM(120)에 제2 ACPI GPE를 전송할 수 있다. 407에서, VM(120)의 게스트 OS(121)가 ACPI 이벤트 상태 레지스터를 관독하고 및/또는 다른 동작들을 수행하여 GPE의 이유를 판단할 수 있다. 408에서, 이러한 동작들은 향상된 VMM(230)에 의해 가로채질 수 있고, 향상된 VMM(230)은 VM(120)에 동적 메모리(215)가 사용가능함을 알릴 수 있다. 이후, 409에서, (ACPI OS 드라이버(123)와 함께) 게스트 OS(121)가 사용가능한 자원들에 동적 메모리(215)를 맵핑할 수 있고 그 후 동적 메모리(215)에 대한 배타적인 액세스를 가질 수 있다.

<18> 위의 설명이 복수의 VM를 실행하는 호스트들에 초점을 맞추지만, 본 발명의 실시예들은 그렇게 제한되지 않는다. 대신, 본 발명의 실시예들은 버스를 공유하는 복수의 독립적인 (가상 또는 가상이 아닌) 컴퓨터 시스템을 갖는 임의의 플랫폼 상에서 구현될 수 있다. 그리하여, 예를 들면, 독립적인 컴퓨터 시스템들을 갖는 서버 시스템에서, 컴퓨터 시스템들 중 하나는 고장들을 위한 백업 시스템으로서 사용될 수 있다. 메인 컴퓨터 시스템의 고장시, 본 발명의 실시예들은 컴퓨터 시스템을 백업하기 위해 모든 메모리 자원들을 동적으로 재할당하기 위한 모니터링 및/또는 컴포넌트 관리에 사용될 수 있고 그리하여 서버 시스템이 임의의 운영 체제들을 재부팅할 필요 없이 실행을 계속할 수 있도록 할 수 있다. 본 발명의 다른 실시예들로부터도 다양한 다른 유형의 시스템들이 이익을 얻을 수 있다.

<19> 본 발명의 실시예들에 따른 호스트들은 다양한 컴퓨팅 디바이스 상에서 구현될 수 있다. 본 발명의 일 실시예에 따르면, 컴퓨팅 디바이스들은 본 발명의 일 실시예를 이루기 위한 명령어들을 실행할 수 있는 다양한 컴포넌트들을 포함할 수 있다. 예를 들면, 컴퓨팅 디바이스들은 적어도 하나의 머신 액세스 가능한 매체를 포함하고 및/또는 적어도 하나의 머신 액세스 가능한 매체에 연결될 수 있다. 본 명세서에서 사용될 때, "머신"은 하나 이상의 프로세서를 갖는 임의의 컴퓨팅 디바이스를 포함하지만, 거기에 제한되지는 않는다. 본 명세서에서 사용되었을 때, 머신 액세스 가능 매체는 컴퓨팅 디바이스에 의해 액세스 가능한 임의의 형식으로 정보를 저장하고 및/또는 송신하는 임의의 기구를 포함하고, 머신 액세스 가능 매체는 (ROM, RAM, 자기 디스크 저장 매체, 광학 저장 매체 및 플래시 메모리 디바이스들과 같은) 기록 가능한/기록 가능하지 않은 매체뿐만 아니라, 전기적, 광학적, 음향학적 또는 (반송파들, 적외선 신호들 및 디지털 신호들과 같은) 전파되는 신호들의 다른 형태를 포함하지만, 그것에 제한되지는 않는다.

<20> 일 실시예에 따르면, 컴퓨팅 디바이스는 하나 이상의 프로세서와 같은 다양한 다른 공지의 컴포넌트들을 포함할 수 있다. 프로세서(들) 및 머신 액세스 가능 매체는 브리지/메모리 컨트롤러를 이용하여 통신적으로 연결될 수 있고, 그 프로세서는 그 머신 액세스 가능한 매체에 저장된 명령어들을 실행할 수 있을 수 있다. 브리지/메모리 컨트롤러는 그래픽 컨트롤러에 연결될 수 있고, 그 그래픽 컨트롤러는 디스플레이 디바이스 상의 디스플레이 데이터의 출력을 제어할 수 있다. 브리지/메모리 컨트롤러는 하나 이상의 버스에 연결될 수 있다. 이러한 요소들 중 하나 이상은 프로세서와 함께 단일 패키지 상에 집적될 수 있거나 복수의 패키지 또는 다이를 사용할 수 있다. USB(Universal Serial Bus) 호스트 컨트롤러와 같은 호스트 버스 컨트롤러는 버스(들)에 연결될 수 있고 복수의 디바이스가, 그 USB에 연결될 수 있다. 예를 들면, 키보드 및 마우스와 같은 사용자 입력 디바이스들이 입력 데이터를 제공하기 위한 컴퓨팅 디바이스에 포함될 수 있다. 대안적인 실시예들에서, 호스트 버스 컨트롤러는 PCI, PCI 익스프레스(PCI Express), 파이어 와이어(Fire Wire) 및 그와 같은 다른 기존 표준들 및 미래의 표준들을 포함하는 다양한 다른 상호접속 표준들과 호환될 수 있다.

<21> 앞의 명세서에서, 본 발명은 특정한 예시적인 실시예들을 참조하여 기술되었다. 그러나 첨부된 청구범위에서 개시된 본 발명의 더 넓은 사상 및 범위에서 벗어나지 않고 다양한 수정들 및 변경들이 만들어질 수 있다는 것이 이해될 것이다. 따라서, 본 명세서 및 도면은 한정적인 의미라기보다는 오히려 예시적인 의미로 여겨져야 한다.

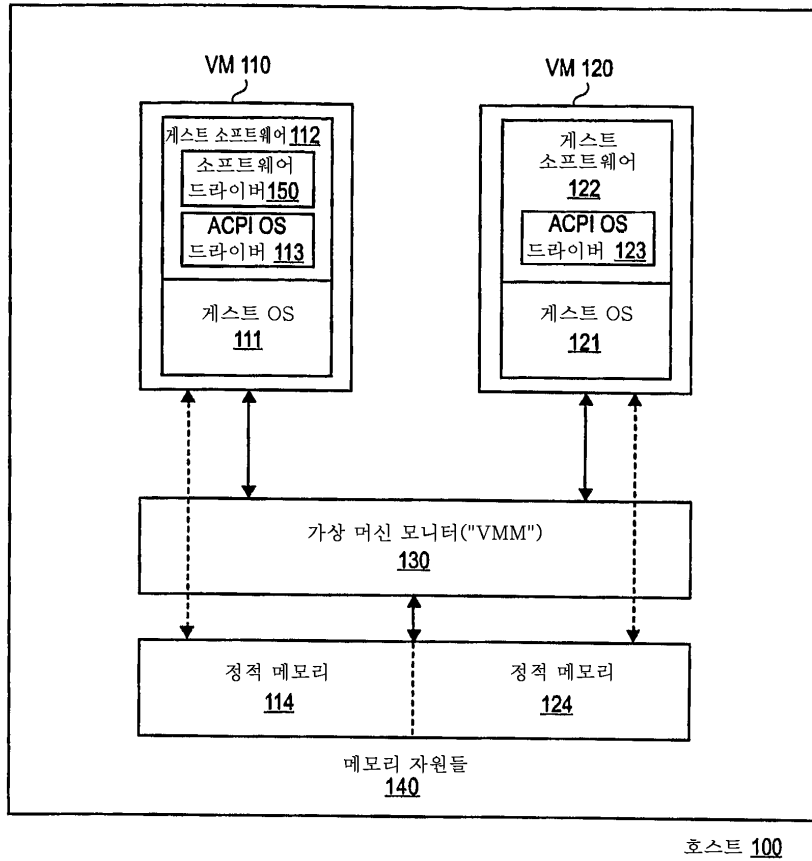
도면의 간단한 설명

- <2> 본 발명은 비슷한 참조번호들이 유사한 요소들을 가리키는 첨부 도면에서 제한이 아닌 예로서 예시된다.
- <3> 도 1은 통상적인 가상 머신 호스트의 예를 예시한다.
- <4> 도 2는 본 발명의 일 실시예의 개요를 예시한다.
- <5> 도 3은 본 발명의 하나의 실시예에 따라 도 2에서 "축출된" 메모리를 새로운 VM에 할당하는 것의 개략을 예시한다.
- <6> 도 4는 본 발명의 일 실시예를 예시하는 흐름도이다.

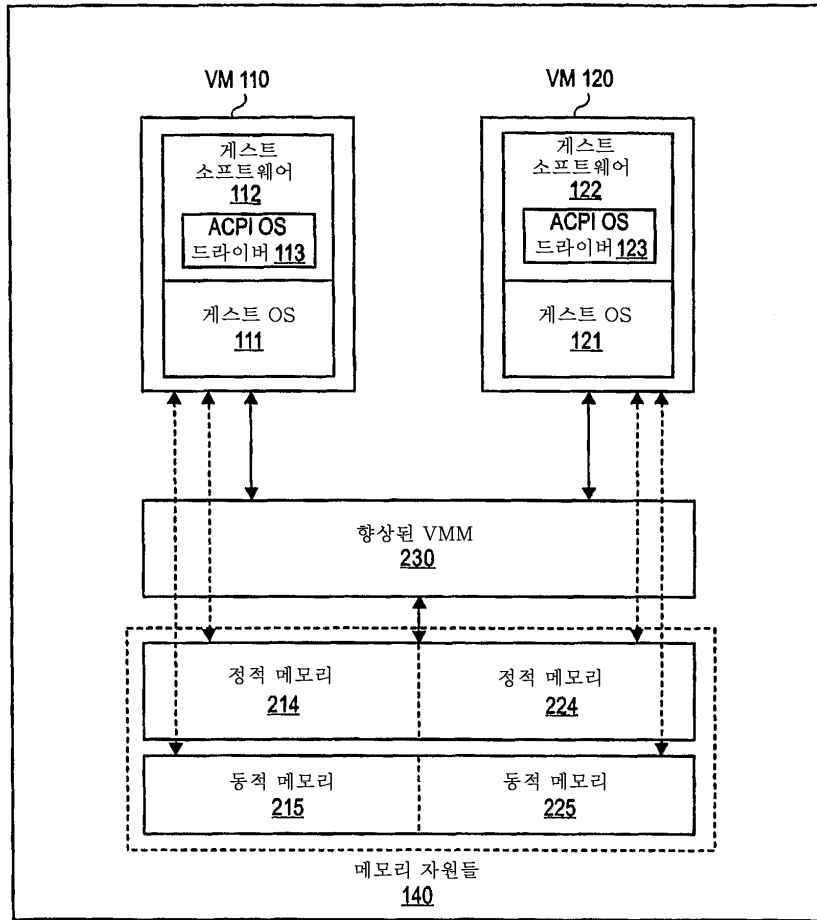
도면

도면1

(종래 기술)

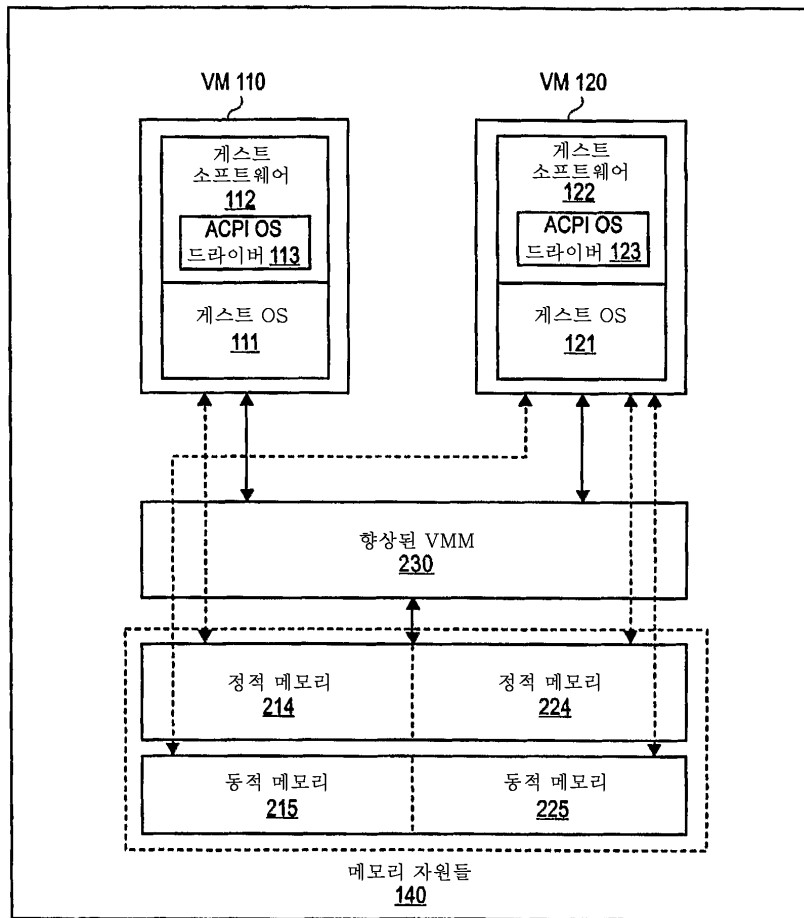


도면2



호스트 100

도면3



호스트 100

도면4

