



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2015년12월03일

(11) 등록번호 10-1574403

(24) 등록일자 2015년11월27일

- (51) 국제특허분류(Int. Cl.)
G06F 9/455 (2006.01) G06F 9/44 (2006.01)
- (21) 출원번호 10-2013-7030130
- (22) 출원일자(국제) 2011년11월23일
심사청구일자 2013년11월13일
- (85) 번역문제출일자 2013년11월13일
- (65) 공개번호 10-2014-0002048
- (43) 공개일자 2014년01월07일
- (86) 국제출원번호 PCT/US2011/062022
- (87) 국제공개번호 WO 2012/145024
국제공개일자 2012년10월26일
- (30) 우선권주장
13/089,066 2011년04월18일 미국(US)
- (56) 선행기술조사문헌
US20080307213 A1*
US20090313391 A1*
US20110087822 A1*
US20050081212 A1*
*는 심사관에 의하여 인용된 문헌

- (73) 특허권자
인텔 코퍼레이션
미합중국 캘리포니아 95054 산타클라라 미션 칼리지 블러바드 2200
- (72) 발명자
첸, 산웨이
미국 97229 오레곤주 포틀랜드 노쓰웨스트 트윈플라워 드라이브 14742
스리니바산, 라만
미국 97229 오레곤주 포틀랜드 노쓰웨스트 린미어 드라이브 3355
코퍼스웨이트, 데이비드 제이.
미국 97214 오레곤주 힐스보로 진 플레이스 노쓰 이스트 스트리트 355
- (74) 대리인
양영준, 백만기

전체 청구항 수 : 총 16 항

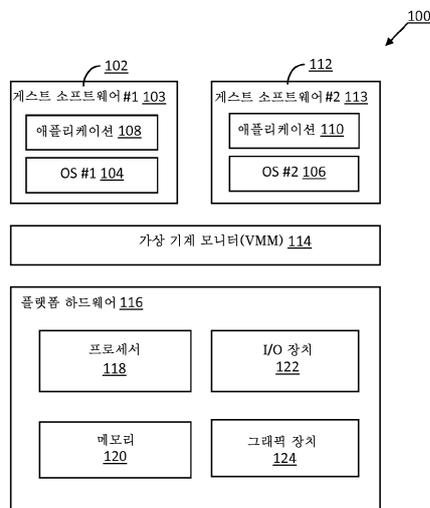
심사관 : 유진태

(54) 발명의 명칭 결합된 가상 그래픽 장치

(57) 요약

시스템은 가상 기계 이동을 지원하기 위해 기본 에뮬레이트된 가상 그래픽 장치 및 물리적 그래픽 장치의 가상 기능을 결합하는 결합된 가상 그래픽 장치를 가진 게스트 그래픽 서브시스템을 포함한다. 시스템의 VMM은 가상 그래픽 장치 및 가상 기능에 대한 액세스를 결합하고, 게스트 OS에서 PnP 이벤트를 트리거하지 않고 그래픽 가속을 위해 가상 그래픽 장치와 가상 기능 간에 전환하는 단일 결합된 가상 PCIe 그래픽 장치를 게스트에 노출시킬 수 있다. 전환에 응답해서, 게스트 그래픽 스택 및 애플리케이션들은 일관된 사용자 경험을 제공하도록 그들의 윈도들을 리드로우할 수 있다.

대표도 - 도1



명세서

청구범위

청구항 1

결합된 가상 장치를 가능하게 하기 위한 시스템으로서,

플랫폼 하드웨어;

게스트 가상 기계;

가상 기계 모니터에서 에뮬레이트된 가상 장치 및 상기 플랫폼 하드웨어의 물리적 장치의 가상 기능 중 하나를 상기 게스트 가상 기계에 할당하고 할당된 장치를 통해 상기 게스트 가상 기계에 결합된 가상 장치를 노출시키기 위한 가상 기계 모니터; 및

상기 에뮬레이트된 가상 장치 및 상기 가상 기능 중 어느 것이 상기 결합된 가상 장치에 대해 활성 상태인지를 검출하고 활성 장치에 정합하는 플러그인을 선택하기 위한 게스트 드라이버

를 포함하고,

상기 게스트 드라이버는 래퍼-플러그인 구조(wrapper-plugin structure)를 포함하고, 상기 래퍼는 상기 활성 장치에 대응하는 플러그인을 결정하는 시스템.

청구항 2

삭제

청구항 3

제1항에 있어서,

상기 게스트 드라이버에 가상 기계 이동을 통지하기 위한 가상 기계 툴 모듈을 더 포함하는 시스템.

청구항 4

제1항에 있어서,

상기 가상 기계 모니터는 또한 할당된 장치 인터페이스와 무관하게 상기 결합된 가상 장치에 대한 구성 공간을 상기 게스트 가상 기계에 노출시키는 시스템.

청구항 5

제4항에 있어서,

상기 가상 기계 모니터는 또한 상기 구성 공간으로부터 활성 장치로의 매핑을 유지하는 시스템.

청구항 6

제4항에 있어서,

상기 가상 기계 모니터는 또한 상기 에뮬레이트된 가상 장치 및 상기 가상 기능을 커버하기 위해 상기 구성 공간의 하나의 또는 그 이상의 필드들을 제공하는 시스템.

청구항 7

제1항에 있어서,

상기 에뮬레이트된 가상 장치는 에뮬레이트된 그래픽 처리 기능을 구현하기 위한 에뮬레이트된 가상 그래픽 장치를 포함하는 시스템.

청구항 8

삭제

청구항 9

제1항에 있어서,

상기 가상 기계 모니터는, 또한, 상기 게스트 가상 기계의 게스트 운영 체제의 플러그-앤-플레이 이벤트(plug-and-play event)를 트리거하지 않고, 상기 결합된 가상 장치 내에서 상기 에뮬레이트된 가상 장치와 상기 물리적 장치의 가상 기능 간에 전환하는 시스템.

청구항 10

컴퓨팅 시스템에 의해 수행되는, 결합된 가상 장치를 가능하게 하기 위한 방법으로서,

에뮬레이트된 가상 그래픽 장치 및 호스트의 물리적 그래픽 장치의 가상 기능 중 하나를 통해 결합된 가상 그래픽 장치를 제공하는 단계;

상기 결합된 가상 그래픽 장치를 게스트에게 노출시키는 단계;

상기 에뮬레이트된 가상 그래픽 장치와 상기 가상 기능 간의 전환에 응답해서, 전환되는 장치(device that is switched to)에 대응하는 플러그인 드라이버를 결정하는 단계; 및

상기 에뮬레이트된 가상 그래픽 장치와 상기 가상 기능 중 전환되는 것을 초기화하는 단계를 포함하는 방법.

청구항 11

제10항에 있어서,

상기 결합된 가상 그래픽 장치를 노출시키는 단계 이후에, 상기 게스트의 게스트 운영 체제로부터의 핫-플러그/언플러그(hot-plug/unplug) 없이 상기 에뮬레이트된 가상 그래픽 장치들과 가상 기능 장치 간에 전환하는 단계를 더 포함하는 방법.

청구항 12

제10항에 있어서,

상기 에뮬레이트된 가상 그래픽 장치와 상기 가상 기능 중 어느 것이 상기 결합된 가상 그래픽 장치에 대해 활성 상태인지와 무관한 상기 결합된 가상 그래픽 장치에 대한 PCIe 구성 공간을 노출시키는 단계를 더 포함하는 방법.

청구항 13

제12항에 있어서,

상기 PCIe 구성 공간은 상기 에뮬레이트된 가상 그래픽 장치와 상기 가상 기능 중 활성인 것에 매핑하는 하나의 또는 그 이상의 필드들을 포함하는 방법.

청구항 14

삭제

청구항 15

제10항에 있어서,

상기 결합된 가상 그래픽 장치를 노출시키는 단계 이후에, 상기 게스트 가상 기계의 게스트 운영 체제의 플러그-앤-플레이 이벤트(plug-and-play event)를 트리거하지 않고, 상기 결합된 가상 장치 내에서 상기 에뮬레이트된 가상 그래픽 장치와 상기 가상 기능 간의 전환을 실행하는 단계를 더 포함하는 방법.

청구항 16

명령들을 포함하는 기계 판독 가능 매체로서,
 상기 명령들은, 처리 시스템에 의해 실행될 때, 상기 처리 시스템이
 호스트의 물리적 그래픽 장치의 가상 기능 및 에뮬레이트된 가상 그래픽 장치 중 하나를 게스트에 할당하고;
 PCIe 그래픽 장치의 가상 모델을 상기 게스트에 노출시키고 - 상기 PCIe 장치는 할당된 장치를 통해 그래픽 기능을 가짐 -;
 상기 가상 기능과 상기 에뮬레이트된 가상 그래픽 장치 중 어느 것이 가상 PCIe 그래픽 장치에 대해 활성 상태 인지를 검출하고;
 검출된 활성 장치에 정합하는 플러그인을 선택하게 하는 기계 판독 가능 매체.

청구항 17

삭제

청구항 18

제16항에 있어서,
 실행됨에 따라, 컴퓨팅 장치가
 가상 PCIe 그래픽 장치에 대한 PCIe 구성 공간을 상기 게스트에 노출시키게 하는 복수의 명령을 더 포함하고,
 상기 PCIe 구성 공간은 상기 가상 기능 및 상기 에뮬레이트된 가상 그래픽 장치의 구성들을 커버하기 위해 상기 할당된 장치와 무관한 기계 판독 가능 매체.

청구항 19

제16항에 있어서,
 실행됨에 따라, 컴퓨팅 장치가
 상기 게스트에서의 가상 기계 이동에 응답해서, 상기 가상 기능 및 상기 에뮬레이트된 가상 그래픽 장치에 대한 액세스를 무시하기 위해 가상 PCIe 그래픽 장치에서 PCIe 기능 레벨 리셋을 수행하게 하는 복수의 명령을 더 포함하는 기계 판독 가능 매체.

청구항 20

제19항에 있어서,
 실행됨에 따라, 컴퓨팅 장치가
 상기 이동에 기초하여 상기 가상 PCIe 그래픽 장치의 기본이 되는 상기 가상 기능과 상기 물리적 그래픽 장치 간에 전환하게 하는 복수의 명령을 더 포함하는 기계 판독 가능 매체.

발명의 설명

배경 기술

[0001] 가상화는 가상화를 위한 하드웨어 및 소프트웨어 지원을 가진 단일 호스트 기계가 호스트의 추상화를 제시할 수 있게 해주어서, 호스트 기계의 기본 하드웨어가 하나의 또는 그 이상의 독립적으로 동작하는 가상 기계들로서 나타난다. 따라서, 각각의 가상 기계는 자립적인 플랫폼으로서 작용할 수 있다. 가상화 기술은 다수의 게스트 운영 체제들 및/또는 다른 게스트 소프트웨어가, 실제로는 동일한 하드웨어 플랫폼에서 물리적으로 실행하면서, 외관상으로는 동시에 그리고 외관상으로는 독립적으로 다수의 가상 기계들에서 공존 및 실행할 수 있게 하는데 사용될 수 있다. 가상 기계는 호스트 기계의 하드웨어를 모방하거나 또는 대안으로 상이한 하드웨어 추상화를 함께 제시할 수 있다.

[0002] 가상화 시스템들은 호스트 기계에 대한 액세스를 제어하는 가상 기계 모니터(VMM)를 포함할 수 있다. 일부 실시예들에서, 임의의 다른 VM 제어 로직이 사용될 수 있다. VMM은 리소스들(예를 들어, 프로세서들, 메모리, IO 장치들)의 집합을 가진 가상 기계에서 동작하는 게스트 소프트웨어를 제공할 수 있다. VMM은 물리적 호스트 기

계의 일부의 또는 모든 컴포넌트들을 가상 기계에 매핑할 수 있으며, 가상 기계(예를 들어, 가상 IO 장치들)가 이용할 수 있는, VMM 내의 소프트웨어로 에뮬레이트된 가상 컴포넌트들을 생성할 수 있다. VMM은 가상 기계에서 서비스들을 제공하고 호스트 기계에서 실행중인 다수의 가상 기계들로부터의 또한 다수의 가상 기계들 간의 보호를 제공하기 위해 하드웨어 가상화 아키텍처의 특징들을 사용할 수 있다.

[0003]

가상화된 컴퓨팅 환경에서, 가상 기계는, "라이브 이동(live migration)"을 통해 또는 VM의 상태의 저장 및 복원을 통해, 하나의 물리적 플랫폼으로부터 다른 물리적 플랫폼으로 이동할 수 있다. VM에 제시된 모든 플랫폼 리소스들이 가상인 경우에, VMM은 이러한 모든 리소스들의 상태를 전송할 수 있다. 물리적 그래픽 장치 기능(예를 들어, 단일-루트 I/O 가상화(SR-IOV) 가능 장치 또는 입력/출력 메모리 관리 모듈(IOMMU)의 지원 하에 매핑된 전용 장치의 가상 기능)에 대한 인터페이스가 VM에 제공된 경우에, 이동 프로세스는 금지될 수 있다. 한 경우에, 이동을 위한 목표 플랫폼은 게스트 OS를 위해 이용 가능한 하드웨어 그래픽 장치들을 가지고 있지 않을 수 있다. 다른 경우에, 목표 플랫폼의 그래픽 하드웨어는 소스와 상이할 수 있다. 호환성 하드웨어가 목표에서 이용 가능한 경우에, VMM은 모든 관련 장치 상태 정보를 전송하지 않을 수 있다. 그래픽 가상화 및 VM 이동의 경우, 게스트 OS는 하드웨어-기반 그래픽 장치들이 이용 가능하게 되고 가상 기계 모니터(VMM)에 의해 VM에 할당됨에 따라 하드웨어-기반 그래픽 장치들의 동적 변경들을 지원할 필요가 있을 수 있다. 일부 운영 체제들은 그래픽 장치들의 동적 플러그-앤-플레이(plug-and-play)를 지원하지 않을 수 있다.

도면의 간단한 설명

[0004]

본 명세서에 기술된 본 발명은 첨부 도면들에서 제한이 아니라 일례로서 도시된다. 설명의 단순성 및 명료성을 위해, 도면들에 도시된 요소들은 반드시 일정한 비율로 그려진 것이 아니다. 예를 들어, 일부 요소들의 치수들은 명료성을 위해 다른 요소들에 비해 과장될 수 있다. 또한, 적합하다고 생각되는 경우에, 참조 부호들은 대응하거나 또는 유사한 요소들을 나타내기 위해 도면들에서 반복되었다.

도 1은 본 발명의 일 실시예에 따른 일례의 가상 기계 환경의 하이 레벨 구조의 개략도이다.

도 2는 한 장치의 일 실시예의 개략도이다.

도 3은 본 발명의 일부 실시예들에 따른 흐름도이다.

도 4a 및 도 4b는 본 발명의 일부 실시예들에 따른 개략적인 흐름도들이다.

발명을 실시하기 위한 구체적인 내용

[0005]

이하의 설명은 결합된 가상 그래픽 장치를 가능케 하는 사용 모델을 제공하는 기술을 설명한다. 일 실시예에서, 결합된 가상 그래픽 장치는 가상 기계 환경에서 구현될 수 있다. 다른 실시예에서, 결합된 가상 그래픽 기능들은 프로세서 기반 컴퓨팅 시스템에서 구현될 수 있다. 본 기술의 구현은 컴퓨팅 시스템들에 제한되지 않으며; 유사한 목적들을 위한 임의의 실행 환경, 예를 들어, 임의의 다른 디지털/전자 장치 또는 시스템에 의해 사용될 수 있다. 이하의 설명에서, 논리 구현들, 연산 코드들, 피연산자를 명시하는 수단, 리소스 분할/공유/복제 구현들, 시스템 컴포넌트들의 타입들 및 상관 관계들, 및 논리 분할/통합 선택 사항들 등의 다수의 특정 세부 사항들이 본 발명의 더 철저한 이해를 제공하기 위해 기재된다. 그러나, 본 발명은 이러한 특정 세부 사항들 없이 실시될 수 있다. 다른 실례들에서, 제어 구조들 및 완전한 소프트웨어 명령 시퀀스들은 본 발명을 모호하게 하지 않기 위해 상세히 도시되지 않았다.

[0006]

본 명세서에서 "one embodiment(일 실시예)", "an embodiment(일 실시예)", "an example embodiment(일례의 실시예)" 등에 대한 언급들은, 기술된 실시예가 특정 특징, 구조, 또는 특성을 포함할 수 있지만, 모든 실시예가 반드시 그 특정 특징, 구조, 또는 특성을 포함하는 것은 아닐 수 있음을 나타낸다. 더욱이, 이러한 구절들은 반드시 동일한 실시예와 관련되는 것이 아니다. 또한, 특정 특징, 구조, 또는 특성이 일 실시예와 관련해서 기술될 때, 명백히 기술되지 않은 기술되지 않은 간에, 당업자가 아는 범위 내에서 다른 실시예들과 관련해서 이러한 특징, 구조, 또는 특성을 달성한다고 제안된다.

[0007]

본 발명의 실시예들은 하드웨어, 펌웨어, 소프트웨어, 또는 임의의 조합으로 구현될 수 있다. 본 발명의 실시예들은, 하나의 또는 그 이상의 프로세서들에 의해 관독 및 실행될 수 있는, 기계-관독 가능 매체에 저장된 명령들로서 구현될 수도 있다. 기계-관독 가능 매체는 기계(예를 들어, 컴퓨팅 장치)에 의해 관독 가능한 형태로 정보를 저장 또는 송신하는 임의의 매커니즘을 포함할 수 있다. 예를 들어, 기계-관독 가능 매체는 관독 전용 메모리(ROM); 랜덤 액세스 메모리(RAM); 자기 디스크 기억 매체; 광 기억 매체; 플래시 메모리 장치들; 전기,

광, 음향 또는 다른 형태들의 전파 신호들(예를 들어, 반송파들, 적외선 신호들, 디지털 신호들 등) 등을 포함할 수 있다.

[0008] 이하의 설명은 오직 설명을 위해 사용되며 제한으로 해석되지 않는 용어들, 예를 들어, 제1, 제2 등을 포함할 수 있다.

[0009] 도 1은 가상 기계 환경(100)의 일 실시예의 블록도이다. 본 실시예에서, 프로세서-기반 플랫폼(116)은 VMM(114)을 실행할 수 있다. VMM은, 소프트웨어로 구현되더라도, 더 높은 레벨의 소프트웨어에 대한 가상 기계 인터페이스를 에뮬레이트하고 익스포트(export)할 수 있다. 이러한 더 높은 레벨의 소프트웨어는 표준 OS, 실시간 OS를 포함할 수 있으며, 또는 제한된 운영 체제 기능을 가진 기본만 남긴(stripped-down) 환경일 수 있으며 일부 실시예들에서 표준 OS에서 이용 가능한 OS 설비들을 포함하지 않을 수 있다. 대안으로, 예를 들어, VMM(114)은 다른 VMM의 서비스들 내에서 또는 그 서비스들을 사용해서 실행될 수 있다. VMM들은, 예를 들어, 일부 실시예들에서 하드웨어, 소프트웨어, 펌웨어 또는 각종 기술들의 조합으로 구현될 수 있다. 적어도 하나의 실시예에서, VMM의 하나의 또는 그 이상의 컴포넌트들은 하나의 또는 그 이상의 가상 기계들에서 실행할 수 있으며, VMM의 하나의 또는 그 이상의 컴포넌트들은 도 1에 도시된 것과 같은 플랫폼 하드웨어에서 실행할 수 있다. 플랫폼 하드웨어에서 직접 실행중인 VMM의 컴포넌트들은 본 명세서에서 VMM의 호스트 컴포넌트들로서 언급된다. 다른 실시예에서, VMM(114)의 일례들은 하이브리드 가상 기계 모니터, 호스트 가상 기계 모니터 또는 하이퍼바이저 가상 기계 모니터를 포함할 수 있다.

[0010] 플랫폼 하드웨어(116)는 퍼스널 컴퓨터(PC), 서버, 메인프레임, 개인 휴대 정보 단말기(PDA), 태블릿들, 스마트폰 또는 임의의 다른 스마트 장치들 등의 핸드헬드형 장치들, 인터넷 프로토콜 장치, 디지털 카메라, 휴대용 컴퓨터, 넷북 또는 노트북 등의 핸드헬드형 PC, 또는 마이크로 컨트롤러, 디지털 신호 프로세서(DSP), 시스템 온 칩(SoC), 네트워크 컴퓨터들(NetPC), 셋톱 박스들, 네트워크 허브들, 광역 통신망(WAN) 스위치들, 또는 다른 프로세서-기반 시스템 등의 내장형 애플리케이션들일 수 있다.

[0011] 플랫폼 하드웨어(116)는 적어도 하나의 프로세서(118) 및 메모리(120)를 포함한다. 프로세서(118)는, 마이크로 프로세서, 디지털 신호 프로세서, 마이크로컨트롤러 등의, 프로그램들을 실행할 수 있는 임의의 타입의 프로세서일 수 있다. 프로세서는 실시예들에서 실행을 위한 마이크로코드, 프로그래밍 가능 로직 또는 하드 코드 로직을 포함할 수 있다. 도 1이 오직 하나의 프로세서(118)를 도시하지만, 일 실시예에서 시스템에 하나의 또는 그 이상의 프로세서들이 있을 수 있다. 또한, 프로세서(118)는 다수의 코어들, 다수의 스레드들을 위한 지원 등을 포함할 수 있다. 프로세서(118)는 본 명세서에 기술된 각종 실시예들과 연관된 동작들을 실행하기 위한 마이크로코드, 프로그래밍 가능 로직 또는 하드 코드 로직을 포함할 수 있다.

[0012] 메모리(120)는 각종 실시예들에서 하드 디스크, 플로피 디스크, 랜덤 액세스 메모리(RAM), 판독 전용 메모리(ROM), 플래시 메모리, 임의의 다른 타입의 휘발성 메모리 장치들 또는 비휘발성 메모리 장치들, 또는 상기 장치들의 조합, 또는 프로세서(118)에 의해 판독 가능한 임의의 다른 타입의 기계 매체를 포함할 수 있다. 메모리(120)는 프로그램 실행 및 다른 방법 실시예들을 실행하기 위한 명령들 및/또는 데이터를 저장할 수 있다. 일부 실시예들에서, 본 발명의 일부 요소들은 다른 시스템 컴포넌트들에서, 예를 들어, 플랫폼 칩셋에서 또는 시스템의 하나의 또는 그 이상의 메모리 컨트롤러들에서 구현될 수 있다.

[0013] VMM(114)은 하나의 또는 그 이상의 가상 기계들의 추상화를 게스트 소프트웨어에 제시할 수 있다. VMM(114)은 상이한 VM들에 물리적 플랫폼의 동일한 또는 상이한 추상화들을 제시할 수 있다. 도 1의 실시예는 2개의 가상 기계들(102 및 112)을 도시한다. 각각의 가상 기계에서 실행중인 게스트 소프트웨어(103 및 113) 등의 게스트 소프트웨어는 게스트 OS(104 또는 106) 및 각종 게스트 소프트웨어 애플리케이션들(108 및 110) 등의 게스트 OS를 포함할 수 있다. 게스트 소프트웨어(103 및 113)는 게스트 소프트웨어(103 및 113)가 실행중인 가상 기계들 내에서 또한 다른 기능들을 실행하기 위해 물리적 리소스들(예를 들어, 프로세서 레지스터들, 메모리 및 I/O 장치들)에 액세스할 수 있다. 예를 들어, 게스트 소프트웨어(103 및 113)는, 가상 기계(102 및 112)에서 제시된 프로세서 및 플랫폼의 아키텍처에 따라, 모든 레지스터들, 캐시들, 구조들, I/O 장치들, 메모리 등에 액세스할 수 있다.

[0014] 일 실시예에서, 프로세서(118)는 가상 기계들(102 및 112)의 동작을 제어할 수 있다. 일 실시예에서, 가상 기계(102)가 그의 가상 어드레스 공간 내의 메모리 위치를 참조함에 응답해서, 호스트 기계(116)의 물리적 메모리(기계 물리적 메모리)의 실제 어드레스에 대한 참조가, 하드웨어(때때로 프로세서(118)에 통합됨) 및 소프트웨어(예를 들어, 호스트 기계의 운영 체제)로 구현될 수 있는, VMM(114)의 메모리 관리 모듈(도시되지 않음)에 의해 생성될 수 있다. 도 1의 실시예에서, 플랫폼 하드웨어(116)는 하나의 또는 그 이상의 I/O 장치들(122)을 또

한 포함할 수 있다. 플랫폼 하드웨어(116)는 하나의 또는 그 이상의 그래픽 기능들을 실행하는데 사용될 수 있는 하나의 또는 그 이상의 그래픽 제어 또는 장치들(124)을 더 포함할 수 있다; 일부 실시예들에서, 플랫폼 하드웨어(116)는 임의의 다른 물리적 그래픽 장치들을 포함할 수 있다.

[0015] 도 2는 일 실시예에 따라 가상 기계 환경에서 구현될 수 있는 결합된 가상 그래픽 장치의 일례를 도시한다. 도 2를 참조하면, 가상 기계 환경(200)은 가상 기계 모니터(VMM)(220) 및 플랫폼(210)을 포함할 수 있다. 플랫폼(210)은 프로세서(118), 메모리(120), 입력/출력 장치(122), 그래픽 장치(124)를 포함할 수 있다; 그러나, 일부 실시예들에서, 플랫폼(210)은 임의의 다른 플랫폼 하드웨어 장치들을 포함할 수 있다. 일 실시예에서, 플랫폼(210)의 프로세서는 소프트웨어 명령들을 실행할 수 있는 임의의 타입의 프로세서, 예를 들어, 복합 명령 집합 컴퓨터(CISC) 마이크로프로세서, 감소된 명령 집합 컴퓨팅(RISC) 마이크로프로세서, 매우 긴 명령어 워드(VLIW) 마이크로프로세서, 명령 집합들의 조합을 구현하는 프로세서, 또는 디지털 신호 프로세서, 마이크로프로세서, 디지털 신호 프로세서 또는 마이크로컨트롤러 등의 임의의 다른 프로세서 장치 동일 수 있다. 플랫폼(210)의 프로세서는 본 명세서에 기술된 각종 실시예들과 연관된 동작들을 실행하기 위한 마이크로코드, 프로그래밍 가능 로직 또는 하드 코드 로직을 포함할 수 있다. 일 실시예에서, 플랫폼(210)에 하나의 또는 그 이상의 프로세서들이 있을 수 있고, 하나의 또는 그 이상의 프로세서들은 다수의 스레드들, 다수의 코어들 등을 포함할 수 있다.

[0016] 본 실시예는 컴퓨팅 시스템들로 제한되지 않는다. 본 발명의 다른 실시예들이 핸드헬드형 장치들 및 내장형 애플리케이션들 등의 임의의 형태 인자 장치들에서 사용될 수 있다. 플랫폼(210)은 핸드헬드형 장치, 휴대형 컴퓨터, 셋톱 박스, 또는 임의의 다른 프로세서 기반 시스템의 하드웨어 플랫폼을 포함할 수 있다. 핸드헬드형 장치들의 일부 일례들은 휴대 전화들, 스마트폰, 태블릿들, 인터넷 프로토콜 장치들, 디지털 카메라, 개인 휴대 정보 단말기(PDA), 또는 넷북 또는 노트북 등의 핸드헬드형 PC들을 포함할 수 있다. 내장형 애플리케이션들은 마이크로 컨트롤러, 디지털 신호 프로세서(DSP), 시스템 온 칩(SoC), 네트워크 컴퓨터들(NetPC), 셋톱 박스들, 네트워크 허브들, 광역 통신망(WAN) 스위치들, 또는 임의의 다른 시스템을 포함할 수 있다.

[0017] 도 2를 참조하면, 일 실시예에서, VMM(220)은 하나의 또는 그 이상의 VM들 또는 가상 시스템들(도시되지 않음)의 추상화를 다른 소프트웨어에 제시할 수 있다. 일 실시예에서, VMM(220)은 환경(200)의 하드웨어 리소스들이 VM들 또는 가상 시스템들 내에서 실행중인 하나의 또는 그 이상의 동시 운영 체제 세션들에 걸쳐 분배될 수 있게 할 수 있다. 게스트 플랫폼(250)의 각각의 VM은, 총괄하여 게스트 소프트웨어라고 하는, 자신의 "게스트 운영 체제"(VMM에 의해 호스팅되는 운영 체제(OS)) 및 다른 소프트웨어를 실행하는, 자립적인 플랫폼으로서 작용할 수 있다. 일 실시예에서, 게스트 플랫폼(250)은 각종 이벤트들을 제어할 수 있으며, 각종 하드웨어 리소스들에 액세스할 수 있다. 게스트 플랫폼(250)의 각각의 가상 기계에서 실행중인 게스트 소프트웨어는 게스트 OS 또는 게스트 커널(230) 및 게스트 사용자 인터페이스(240)의 각종 게스트 소프트웨어 애플리케이션들(242)을 포함할 수 있다. VMM(220)은 하드웨어, 소프트웨어, 펌웨어, 또는 조합들로 구현될 수 있다. 일 실시예에서, 게스트 플랫폼(250)은 게스트 그래픽 서브시스템으로서 작용할 수 있다.

[0018] 도 2를 참조하면, 일 실시예에서, 플랫폼(210)은 물리적 그래픽 장치(212)를 포함할 수 있다. 일 실시예에서, 물리적 그래픽 장치(212)는 SR-IOV 또는 임의의 다른 표준 등의 하드웨어 가상화 표준을 지원할 수 있으며, VMM(220)은 물리적 그래픽 장치(212)의 하나의 또는 그 이상의 가상 기능들(214)을 게스트 OS(230)에 할당할 수 있다. 도 2의 실시예에서, VMM(220)은 결합된 가상 그래픽 장치(CVGD)(222)를 게스트 OS(230)에 노출시킬 수 있다. 일 실시예에서, 결합된 가상 그래픽 장치(222)는 기본 에플레이트된 비디오 그래픽 어레이(VGA) 장치(224), 에플레이트된 가상 그래픽 처리 유닛(VGPU)(226) 등의 가상 그래픽 장치, 및 물리적 그래픽 장치(212)의 가상 기능(VF)(214)을 결합하여, VGA(224)를 통해 소프트웨어-에플레이트된 VGA 기능을 제공하고 장치(212)의 VGPU(226) 및 VF(214) 중 하나를 통해 그래픽 기능들을 제공할 수 있다. 다른 실시예에서, 결합된 가상 그래픽 장치(222)는 VGPU(226) 및 VF(214)에 대한 액세스를 결합할 수 있는 단일 PCIe(peripheral component interconnect express) 그래픽 장치로서 VM에 제시될 수 있다. 결합된 가상 그래픽 장치(222)는, 예를 들어, VM에 제공된 그래픽 가속 및/또는 다른 그래픽 기능들을 위해, VGPU(226)와 VF(214) 간에 전환할 수 있다.

[0019] VMM(220)에 의해 제공된 VGA(224)는 그래픽 장치(212)의 가상 기능들(214)을 보장할 수 있다. 일 실시예에서, VGA(224)는, 예를 들어, (VF 드라이버가 로드되기 전에) 부팅 프로세스 중에 및/또는 진단 출력 동안 사용될 수 있다. 일부 실시예들에서, VMM(220)은, 예를 들어, 향상된 그래픽 기능을 지원하기 위해 게스트 OS(230)에 결합된 가상 그래픽 장치(222)의 슈퍼 VGA를 제공할 수 있다. 일 실시예에서, VF(214)가 없는 플랫폼의 경우, VMM(220)은 소프트웨어 에플레이션(예를 들어, VGPU(226))을 이용해 또는 호스트 그래픽 하드웨어에 의한 가속화를 이용해 그래픽 및/또는 미디어 렌더링 기능 등의 에플레이트된 그래픽 처리 기능들을 구현할 수 있다.

VMM(220)은, 예를 들어, 게스트 OS(230)의 플러그-앤-플레이(PnP) 이벤트를 트리거하지 않고, 동일한 결합된 가상 그래픽 장치(222) 내에서 기본 에뮬레이트된 VGPU(226)와 VF(214) 간에 동적으로 전환할 수 있다.

[0020]

일 실시예에서, VMM(220)은 결합된 가상 그래픽 장치(222)에 대한 PCIe 구성 공간을 게스트 OS(230)에 노출시킬 수 있다. PCIe 구성 공간은 기본 그래픽 장치들 중 어느 것이 사용되는지와 무관할 수 있다. 예를 들어, VMM(220)은, 예를 들어, 에뮬레이트된 그래픽 장치 VGPU(226)가 사용되는지 VF(214)가 사용되는지 또는 가상 기능이 어떤 그래픽 하드웨어로부터의 것인지와 무관하게 결합된 가상 그래픽 장치(222)에 대한 동일한 PCIe 구성 공간을 유지할 수 있다. VMM(220)은 결합된 가상 그래픽 PCIe 구성 공간(예를 들어, 게스트 OS에 가시적임)으로부터 활성 기본 장치로의 매핑을 또한 유지할 수 있다. 구성 공간은 하나의 또는 그 이상의 기본 장치들의 구성들을 커버하기 위해 베이스 어드레스 레지스터들(BAR들)의 과도 제공(over-provision) 수들 및/또는 크기들을 가질 수 있다.

[0021]

일 실시예에서, PCIe 구성 공간은 하나의 또는 그 이상의 레지스터들 및/또는 필드를 포함할 수 있다. 예를 들어, 가상 그래픽 PCIe 구성 공간은 벤더 아이덴티티(VID), 장치 ID(DID) 등의 데이터 필드들을 포함할 수 있다. 다른 실시예에서, 가상 그래픽 PCIe 구성 공간은 구성 공간의 제어 레지스터 집합의 일부일 수 있는 BAR0 및 BAR1 내지 BAR5 등의 베이스 어드레스 레지스터들(BAR들)을 포함할 수 있다. 예를 들어, BAR들은 메모리-맵 I/O(MMIO) 레지스터들, 또는 I/O 레지스터들, 또는 메모리 영역들을 포함할 수 있다. 다른 실시예에서, 가상 장치의 경우, VMM은 PCIe 구성 레지스터들의 가상 벤더 및/또는 장치 ID 등의 정보를 저장할 수 있고/있거나; BAR들 중 한 BAR 내의 MMIO 레지스터들의 현재 상황에 대한 구성을 위해 드라이버에 의해 사용될 수 있는 임의의 정보를 저장할 수 있다. 일 실시예에서, VID/DID는 구성 공간 내에 있을 수 있지만, BAR0 또는 임의의 다른 BAR 내에는 있지 않을 수 있다; 일부 실시예들에서, VID 및/또는 DID는 임의의 다른 레지스터 내에 있을 수 있다.

[0022]

일 실시예에서, VID는 기본 하드웨어와 무관할 수 있는 벤더 ID일 수 있으며, VF(214)(또는 그래픽 장치(212)) 등의 하드웨어의 삽입/취소에 걸쳐 변하지 않을 수 있다. DID는 기본 하드웨어와 무관한 ID 값을 가질 수 있는 가상 장치 ID(DID)일 수 있으며, VF(214) 등의 하드웨어의 삽입/취소에 걸쳐 지속적일 수 있다. 다른 실시예에서, BAR0은 기본 그래픽 장치를 식별하는데 사용될 수 있는 하나의 또는 그 이상의 VMM 에뮬레이트된 메모리-맵 I/O(MMIO) 레지스터들에 매핑할 수 있다. 일 실시예에서, BAR0은 2개의 레지스터들 HW VID 및 HW DID를 포함할 수 있다; 일부 실시예들에서, BAR0은 하나의 또는 그 이상의 레지스터들을 포함할 수 있다. HW VID 레지스터는, 에뮬레이트된 그래픽 장치(226) 또는 VF(214)일 수 있는, 기본 그래픽 장치의 벤더 ID를 저장할 수 있다. HW DID 레지스터는, 에뮬레이트된 그래픽 장치(226) 또는 VF(214)일 수 있는, 기본 그래픽 장치의 장치 ID를 저장할 수 있다. 일부 실시예들에서, 벤더 ID, 장치 ID 및 BAR0은 물리적 기능들(PF) 및/또는 가상 기능들(VF) 등의 싱글-루트 I/O 가상화(SR-IOV) 그래픽 하드웨어에 매핑하지 않을 수 있다. 다른 실시예에서, 가상 그래픽 PCIe 구성 공간은 하나의 또는 그 이상의 에뮬레이트된 그래픽 장치들 및/또는 하나의 또는 그 이상의 지원 VF 구현들을 커버하기 위해 수 및/또는 크기 면에서 과도하게 제공될 수 있는 BAR1 내지 BAR5 등의 하나의 또는 그 이상의 레지스터들을 가질 수 있다; 일부 실시예들에서, 상이한 수의 리턴던트 레지스터들이 제공될 수 있다. VMM(220)은 활성 에뮬레이트된 그래픽 장치(226) 또는 VF(214) 등의 현재 활성 기본 그래픽 장치의 레지스터들에 리턴던트 BAR1 내지 BAR5를 동적으로 매핑할 수 있다.

[0023]

일 실시예에서, VMM(220)은 에뮬레이트된 그래픽 장치(226)가 사용되는지 VF(214)가 사용되는지와 무관하게, 또는 가상 기능이 어떤 그래픽 하드웨어로부터의 것인지와 무관하게, 동일한 PCIe 구성 공간을 유지할 수 있다. VMM(220)은 가상 그래픽 PCI 구성 공간(게스트 OS에 가시적임)으로부터 활성 기본 그래픽 장치로의 매핑을 유지할 수 있다.

[0024]

도 2를 참조하면, 게스트 OS(230)는 게스트 그래픽 드라이버(232)를 포함할 수 있다. VMM(220)은 게스트 그래픽 드라이버(232)가 결합된 가상 PCIe 그래픽 장치(222)에 의해 노출된 활성 기본 장치를 검출하기 위한 수단을 제공할 수 있다. 일 실시예에서, 게스트 그래픽 드라이버(232)는 래퍼-플러그인 구조(wrapper-plugin structure)를 가질 수 있다. 기본 장치 VGPU(226) 및 VF(214) 각각은 게스트 그래픽 드라이버(232)의 대응 플러그인에 의해 관리될 수 있다. 예를 들어, 게스트 그래픽 드라이버(232)는 VGPU(226)를 위한 드라이버 플러그인(236) 및 VF(214)를 위한 플러그인(238)을 포함할 수 있다. 기본 그래픽 장치의 각각의 드라이버 플러그인은 기본 그래픽 장치의 BAR 매핑에 대한 지식을 가질 수 있다. 일 실시예에서, 래퍼(234)는 결합된 가상 그래픽 장치(222)에 의해 노출된 활성 기본 그래픽 장치를 검출하고 검출된 기본 장치에 정합하는 적합한 플러그인을 선택할 수 있다. 래퍼(234)는 현재의 활성 기본 장치에 대한 플러그인을 결정할 수 있고/있거나, VMM(220)이 기본 그래픽 장치를 전환함에 응답해서 새로운 플러그인으로 전환할 수 있다. 래퍼(234)는 또한 하나의 또는

그 이상의 표준 그래픽 장치 드라이버 인터페이스들(DDI)의 집합을 게스트 OS(230) 및/또는 그래픽 스택(244)에 제공할 수 있으며, 실행을 위해 인터페이스들을 현재 활성 플러그인에 매핑할 수 있다. 예를 들어, OS는 그래픽 스택에 대한 표준 인터페이스, 예를 들어, Windows용 DirectX를 정의할 수 있다; 그러나, 일부 실시예들은 임의의 표준 인터페이스를 사용할 수 있다.

[0025]

예를 들어, VMM(220)에 의한 기본 그래픽 장치의 전환에 응답해서, 래퍼(234)는 통지 및 스크린-리드로우(screen-redraw)를 사용해서 사용자 경험을 제공할 수 있다. 일 실시예에서, VM 이동에 응답해서, 래퍼(234)는 게스트 OS(230) 및 그래픽 스택(244)에 기본 그래픽 장치의 전환을 통지할 수 있다. 예를 들어, 래퍼(234)는, 예를 들어, Windows Vista에서, 드라이버 콜의 타임-아웃을 통해 전환을 게스트 OS(230) 및 그래픽 스택(244)에 통지할 수 있다. 다른 실시예에서, 래퍼(234)는 그래픽 장치를 다시 초기화할 수 있는 게스트 OS(230)로부터 복구 시퀀스를 프롬프팅할 수 있다. 다른 실시예에서, 게스트 그래픽 드라이버(232)는 래퍼(234)로부터의 통지에 응답해서 활성 기본 장치를 위한 플러그인을 호출할 수 있다. 다른 실시예에서, VM 이동시 기본 그래픽 장치 전환의 통지에 응답해서, 그래픽 스택(244) 및/또는 게스트 사용자 인터페이스(UI) 애플리케이션(예를 들어, 242)은 그들의 윈도를 리드로우(redraw)할 수 있다. 일 실시예에서, 게스트 OS(230) 및 그래픽 스택(244)은 그래픽 스택(244) 및/또는 UI 애플리케이션(242)에 의해 그래픽 하드웨어 리셋 및/또는 자동 리드로우를 지원할 수 있다. 도 2의 실시예에서, 게스트 사용자 인터페이스(240)는, 예를 들어, 도 4a 및 도 4b를 참조해서, VM 이동의 개시 및/또는 관리에 사용될 수 있는 VM 툴들(246)을 더 포함할 수 있다.

[0026]

도 2가 VGPU(226) 및 VF(214)를 도시하지만, 일부 실시예들에서, VMM(220)에 의해 구현되는 결합된 가상 그래픽 장치는 하나의 또는 그 이상의 기본 하드웨어 및/또는 소프트웨어 기반 장치들을 커버할 수 있으며, 장치들이 이용 가능하게 되거나 또는 이용할 수 없게 됨에 응답해서 VMM(220)은 다수의 기본 장치들 간에 전환할 수 있다. 도 2가 결합된 가상 그래픽 장치의 실시예를 도시하지만, 일부 실시예들에서, VMM(220)은 기본 장치들의 가용성에 기초하여 결합된 가상 장치를 게스트에 제공하기 위해 임의의 다른 기본 장치들 간에 전환할 수 있다.

[0027]

도 3은 본 발명의 일 실시예에 따른 한 방법의 일 실시예를 도시한 흐름도이다. 일 실시예에서, 도 3의 흐름은 게스트 시동과 관련될 수 있다. 블록(302)에서, VMM은 VF 등의 기본 그래픽 장치 또는 에뮬레이트된 그래픽 장치를 게스트 OS에 할당할 수 있다. 블록(304)에서, VMM은 게스트를 위한 가상 PCIe 그래픽 장치를 구성할 수 있다. 예를 들어, VMM은 게스트 OS를 위해 할당된 기본 그래픽 장치를 통해 결합된 가상 PCIe 그래픽 장치를 제공할 수 있다. 블록(304)에서, VMM은 결합된 가상 PCIe 그래픽 장치의 BAR들을 블록(302)에서 할당된 기본 그래픽 장치의 리소스에 매핑할 수 있다. 일 실시예에서, 블록(306)에서, 게스트 OS는 부팅 중에 디스플레이를 위해 결합된 가상 PCIe 그래픽 장치에 의해 노출된 에뮬레이트된 VGA를 사용할 수 있다. 블록(308)에서, 게스트 OS는 결합된 가상 그래픽 PCIe 장치를 검출하고 검출된 가상 그래픽 장치에 대한 그래픽 드라이버 래퍼(예를 들어, 234)를 로드할 수 있다. 블록(310)에서, 게스트 래퍼 드라이버는 가상 PCIe 구성 공간 레지스터들 또는 BAR0 등의 에뮬레이트된 MMIO 레지스터의 정보를 통해 어떤 기본 그래픽 장치가 활성 상태인지를 검출할 수 있다. 블록(310)에서, 게스트 래퍼 드라이버는 에뮬레이트된 그래픽 장치(226) 또는 VF(214) 등의 검출된 활성 기본 그래픽 장치를 위한 대응 드라이버 플러그인을 로드할 수 있고/있거나, 로드된 드라이버 플러그인에 의해 활성 기본 그래픽 장치를 초기화할 수 있다.

[0028]

도 4a는 본 발명의 일 실시예에 따른 한 방법의 일 실시예를 도시한 흐름도이다. 일 실시예에서, 도 4a의 흐름은 게스트의 이동 소스 플랫폼에 의한 VM 이동을 준비하는 동작과 관련될 수 있다. 블록(402)에서, 게스트 VM의 VM 툴 모듈(예를 들어, 246)은 다가오는 VM 이동을 게스트 그래픽 드라이버 래퍼에게 통지할 수 있다. 블록(404)에서, 게스트 그래픽 드라이버 래퍼는 결합된 가상 그래픽 PCIe 장치에서 PCIe 기능 레벨 리셋(FLR)을 실행할 수 있으며, VGPU(226) 등의 에뮬레이트된 그래픽 장치 및 VF 장치(214) 등의 가상 VF PCIe 장치에 대한 다음 액세스를 무시할 수 있다. 일 실시예에서, FLR에 응답해서, 게스트는 FLR의 완료에 응답해서 VM 이동에 대비할 수 있다.

[0029]

도 4b는 본 발명의 일 실시예에 따른 한 방법의 일 실시예를 도시한 흐름도이다. 일 실시예에서, 도 4b의 흐름은 게스트에서 이동 목표 플랫폼에 의해 실행될 수 있는 VM 이동 동작과 관련될 수 있다. 블록(406)에서, 예를 들어, 제1 VM으로부터 제2 VM으로의 VM 이동에 응답해서, VMM은 결합된 가상 PCIe 그래픽 장치를 (예를 들어, 블록(404)과 관련하여 상술된 FLR에 의한 리셋 상태로) 게스트에 제공할 수 있다. 예를 들어, VM이 이동할 목표 플랫폼 및/또는 VMM 구성에 기초하여, 결합된 가상 PCIe 장치는 기본 장치로서 VGPU(226) 또는 VF(예를 들어, 214) 등의 에뮬레이트된 그래픽을 가질 수 있다. 블록(408)에서, 게스트 그래픽 드라이버 래퍼는 결합된 가상 PCIe 장치에 대한 액세스를 계속해서 무시할 수 있다. 블록(410)에서, VM 툴 모듈은 그래픽 하드웨어가

변경되었고 다시 사용할 준비가 되어 있음을 게스트 그래픽 드라이버 래퍼에 통지할 수 있다.

[0030]

블록(412)에서, 게스트 그래픽 드라이버 래퍼는 PCIe 구성 공간 및/또는 BARO 등의 에플레이트된 MMIO 레지스터의 정보를 통해 기본 그래픽 장치를 검출할 수 있으며, 검출된 기본 그래픽 장치에 대한 대응 드라이버 플러그인을 로드할 수 있다. 게스트 그래픽 드라이버 래퍼는 로드된 드라이버 플러그인을 사용해서 새로운 검출된 기본 그래픽 장치를 초기화할 수 있다. 블록(414)에서, 게스트 그래픽 드라이버 플러그인은 게스트 VM의 그래픽 스택의 리셋을 개시할 수 있으며, 그래픽 스택 및 게스트 UI 애플리케이션(들)은 리셋에 응답해서 그들의 윈도 또는 표현을 리드로우할 수 있다. 블록(416)에서, 게스트 그래픽 드라이버 및 게스트 스택은 정상 동작을 재개할 수 있다.

[0031]

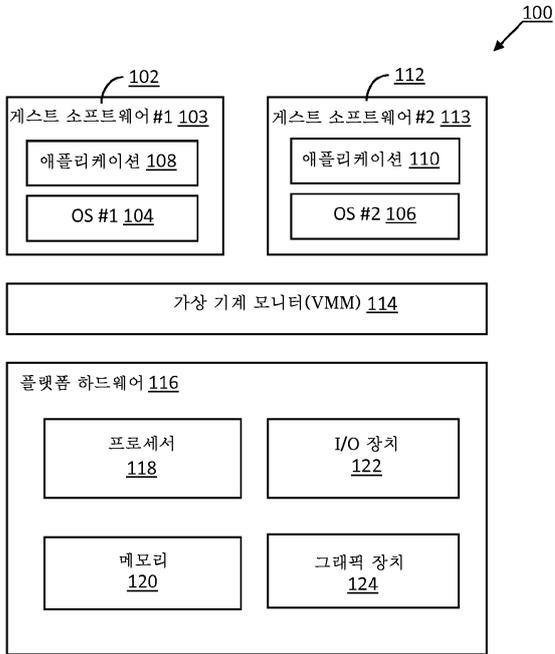
도 3, 도 4a 및/또는 도 4b의 흐름들이 일련의 프로세스들을 포함하는 것으로 도시되어 있지만, 일부 실시예들의 흐름들은 도시된 프로세스들을 상이한 순서로 실행할 수 있다. 다른 실시예에서, 도 3, 도 4a 및/또는 도 4b의 흐름들은 라이브 이동 및/또는 saveVM/loadVM 타입의 이동을 지원할 수 있다. 다른 실시예에서, 도 3, 도 4a 및/또는 도 4b의 흐름은 게스트 OS 및 그래픽 장치 드라이버 모델로부터 핫-플러그/언플러그 기능(hot-plug/unplug capability)을 요구하지 않을 수 있다. 도 3, 도 4a 및/또는 도 4b의 흐름들이 하나의 또는 그 이상의 기본 그래픽 장치들/기능들 간의 이동에 사용될 수 있지만, 일부 실시예들에서, 흐름들은 VMM의 임의의 다른 장치들/기능들 플러그-앤-플레이 지원을 구현하는데 사용될 수 있다. 일부 실시예들은 VMM의 그래픽 장치 플러그-앤-플레이 지원을 구현하기 위해 기본 장치들의 변경들과 함께 동작을 변경할 수 있는 결합된 가상 그래픽 장치를 노출시킬 수 있다. 일 실시예에서, 결합된 가상 그래픽 장치는 모든 VMM 구현들에 걸쳐 표준일 수 있는 인터페이스를 노출시킬 수 있으며, 게스트 OS 그래픽 장치 드라이버는 임의의 특정 VMM 구현과 무관할 수 있다. 다른 실시예에서, VMM에 의해 구현된 결합된 가상 그래픽 장치는 하드웨어, 소프트웨어, 펌웨어 및/또는 그 조합에 기초할 수 있는 하나의 또는 그 이상의 기본 장치들을 커버할 수 있으며, 이용 가능하게 되거나 이용할 수 없게 됨에 따라 기본 장치들 간에 전환할 수 있다. 다른 실시예에서, 결합된 가상 그래픽 장치는 모든 VMM 구현들에 걸쳐 표준일 수 있는 PCIe 인터페이스를 노출시킬 수 있다. 또 다른 실시예에서, 게스트 OS의 래퍼-플러그인 그래픽 장치 드라이버 구조는 기본 장치가 그의 동작들을 변경함에 따라 새로운 플러그인 모듈들로 전환할 수 있다. 래퍼와 플러그인들 간의 인터페이스는 각각의 게스트 OS 타입에 대해 표준일 수 있으며, 임의의 VMM 구현에 좌우되지 않을 수 있다. 결합된 가상 그래픽 장치에서, 하나의 또는 그 이상의 기본 그래픽 장치들/기능들이 단일 PCIe 장치로 결합될 수 있다. 결합된 가상 그래픽 장치는 VMM에 의해 구현될 수 있지만, 모든 VMM 구현들에 걸쳐 표준일 수 있는 PCIe 인터페이스를 게스트 OS에 노출시킬 수 있다. 게스트 OS의 래퍼-플러그인 그래픽 장치 드라이버에서, 기본 PCIe 그래픽 장치의 동작의 변경이 검출될 수 있으며, 적합한 플러그인으로 전환될 수 있다.

[0032]

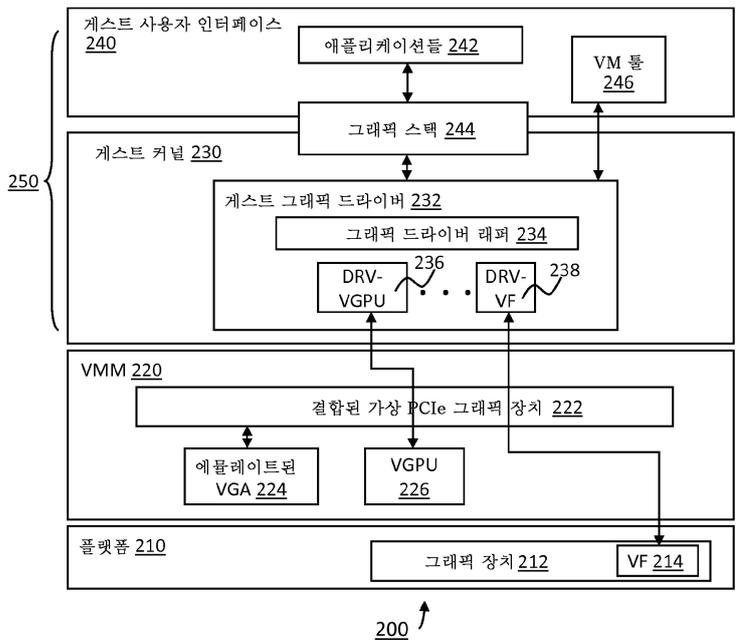
본 발명의 특정 특징들이 실시예들을 참조해서 기술되었지만, 본 설명은 제한적인 의미로 해석되도록 의도된 것이 아니다. 본 실시예들뿐만 아니라, 본 발명이 속한 당업자에게 명백한, 본 발명의 다른 실시예들의 각종 변경들은 본 발명의 원리 및 범위 내에 있는 것으로 여긴다.

도면

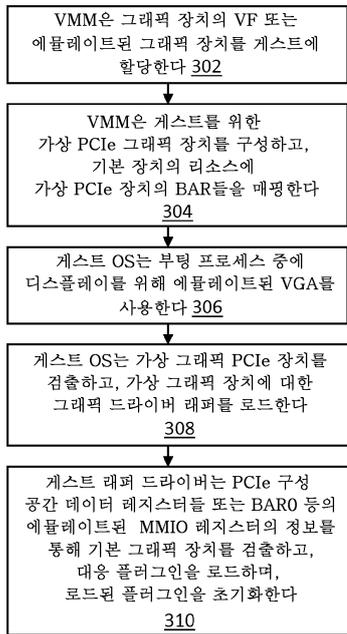
도면1



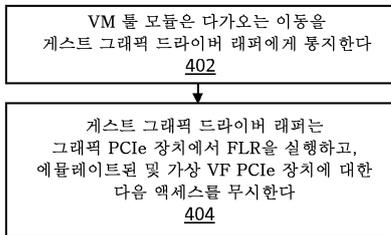
도면2



도면3



도면4a



도면4b

