República Federativa do Brasil
Ministério do Desenvolvimento, Indústria e do Comércio Exterior
Instituto Nacional da Propriedade Industrial

(21) PI 0715921-8 A2

(22) Data de Depósito: 28/08/2007
(43) Data da Publicação: 30/07/2013
(RPI 2221)

(51) Int.Cl.:
G06F 15/16

(54) Título: GERENCIAMENTO DE ESTADO DE HARDWARE DISTRIBUÍDO EM MÁQUINAS VIRTUAIS

(30) Prioridade Unionista: 29/09/2006 US 11/540.211

(73) Titular(es): Microsoft Corporation

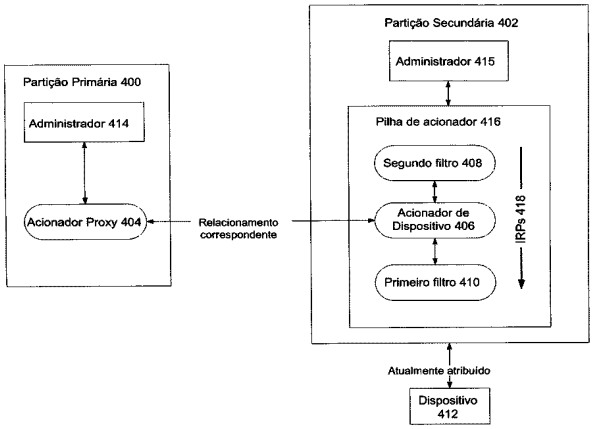
(72) Inventor(es): Adrian J. Oney, Jacob Oshins

(74) Procurador(es): Alexandre Ferreira

(86) Pedido Internacional: PCT US2007076947 de 28/08/2007

(87) Publicação Internacional: WO 2008/039625de 03/04/2008

(57) Resumo: "GERENCIAMENTO DE ESTADO DE HARDWARE DISTRIBUÍDO EM MÁQUINAS VIRTUAIS". São aqui divulgados mecanismos que gerenciam operações em ambientes de máquina virtual. Uma primeira partição pode ter um objeto acionador proxy correspondente a um objeto acionador em uma segunda partição.O objeto acionador pode controlar um dispositivo físico, mas, em virtude do objeto acionador proxy, a primeira partição pode reter alguma medida de controle sobre o dispositivo físico. O objeto acionador pode ser circundado por um primeiro objeto de filtro abaixo dele, e por um segundo objeto de filtro acima dele. O primeiro objeto de filtro pode fornecer interfaces ao objeto acionador de forma que o objeto acionador possa realizar várias funcionalidades relacionadas a barramento, e o segundo objeto pode receber instruções redirecionadas da primeira partição e fornecê-las ao objeto acionador e interceptar todas as instruções originadas da segunda partição, de maneira tal que se estas instruções estiverem inconsistentes com políticas ajustadas na primeira partição, elas possam se manipuladas.



"GERENCIAMENTO DE ESTADO DE HARDWARE DISTRIBUÍDO EM MÁQUINAS VIRTUAIS"

Antecedentes da Invenção

Na maior parte, dispositivos de entrada / saída (I/O) não destinam-se a ser compartilhados por múltiplos sistemas operacionais. Eles são projetados para ser controlados por um módulo de software integrado que gerencia todos os aspectos funcionais de tais dispositivos. É mesmo difícil, em um dado sistema operacional que controla toda uma máquina física, garantir que um único módulo de software controle um dispositivo. Em ambientes de máquina virtual que têm múltiplos sistemas operacionais em execução simultânea é muito mais difícil.

Projetistas de ambientes de máquina virtual estão sempre diante de um dilema. Sistemas operacionais individuais podem controlar diretamente dispositivos individuais, ou eles podem se basear tanto na camada de máquina virtual quanto em outros sistemas operacionais em execução na mesma máquina para serviços I/O. No primeiro caso, gerenciamento complexo de toda a máquina é envolvido e, no último caso, a lentidão é uma preocupação. Assim, um problema que o assunto em questão agora divulgado resolve é o problema da "atribuição de dispositivo" – em outras palavras, como atribuir dispositivos a módulos de maneira tal que a atribuição não seja complexa e o processamento ainda seja rápido.

Além do mais, máquinas virtuais (ou partições) que possuem dispositivos de hardware físico podem ser adicionalmente decompostas com base eu seus relacionamentos de confiança. Em alguns ambientes de máquina virtual, uma partição pode possuir um dispositivo, mas não confiar em nenhuma outra partição na máquina. Em outros, uma partição pode confiar em uma outra partição, especialmente uma que gerencia política para toda a máquina. Assim, um outro problema que o assunto em questão agora divulgado resolve é a atribuição de dispositivos a partições não privilegiadas, em que há pelo menos uma partição confiável (ou componente na própria camada da máquina virtual) que gerencia política para toda a máquina.

Sumário da Invenção

O assunto em questão agora divulgado resolve os problemas expostos associados com a atribuição de dispositivo, especialmente no contexto dos ambientes de máquina virtual com partições confiáveis e não privilegiadas. Em um aspecto exemplar e não limitante, uma primeira partição, que pode ser uma partição confiável primária, pode ter um objeto acionador proxy correspondente a um objeto acionador de dispositivo em uma segunda partição, que pode ser uma partição não confiável secundária. O objeto acionador de dispositivo na partição secundária pode controlar um dispositivo físico, mas, em virtude de este ajuste ter o objeto acionador proxy localizado na partição primária, a partição primária pode reter alguma medida de controle sobre o dispositivo físico. Assim, em um aspecto do assunto em

questão agora divulgado, o controle sobre o estado do dispositivo pode ser separado. A partição primária pode ter controle sobre o estado do dispositivo em todo o sistema, tais como o ligamento ou o desligamento de um dispositivo, enquanto que a partição secundária pode manter controle sobre estado de dispositivo mais local, tal como funcionalidade I/O.

Em um outro aspecto, o objeto acionador, residente em uma pilha, pode ser circundado por um primeiro objeto de filtro abaixo dele e por um segundo objeto de filtro acima dele. O primeiro objeto de filtro pode fornecer interfaces ao objeto acionador de forma que o objeto acionador possa realizar várias funcionalidades relacionadas a barramento. E, além do mais, o segundo objeto de filtro pode realizar pelo menos duas tarefas: (1) receber instruções redirecionadas (pelo objeto acionador proxy) da partição primária e fornecê-las ao objeto acionador na partição secundária, e (2) interceptar todas as instruções originadas dos módulos na partição secundária, de maneira tal que, se estas instruções estiverem inconsistentes com políticas ajustadas na partição primária, elas possam ser substituídas, alteradas ou ignoradas.

Deve-se notar que este Sumário é fornecido para introduzir uma seleção de conceitos de uma forma simplificada que é descrita com detalhes a seguir na Descrição Detalhada. Não pretende-se que este Sumário identifique recursos chaves ou recursos essenciais do assunto em questão reivindicado, nem pretende-se que seja usado como um auxílio na determinação do escopo do assunto em questão reivindicado.

Descrição Resumida dos Desenhos

O Sumário exposto, bem como a seguinte Descrição Detalhada, é mais bem entendido quando lido em conjunto com os desenhos anexos. A fim de ilustrar a presente divulgação, vários aspectos da divulgação são mostrados. Entretanto, a divulgação não é limitada aos aspectos específicos discutidos. As seguintes figuras são incluídas:

a figura 1 é um diagrama de blocos que representa a disposição em camadas lógicas da arquitetura de hardware e de software para ambiente operacional virtualizado em um sistema de computador;

a figura 2 é um diagrama de blocos que representa um sistema de computação virtualizado, em que a virtualização é realizada pelo sistema operacional hospedeiro (tanto diretamente quanto por meio de um hipervisor);

a figura 3 é um diagrama de blocos que representa um sistema de computação virtualizado alternativo, em que a virtualização é realizada por um monitor de máquina virtual em execução ao lado de um sistema operacional hospedeiro;

a figura 4 ilustra um sistema exemplar e não limitante para o gerenciamento de operações em um ambiente de máquina virtual;

a figura 5 ilustra uma implementação exemplar e não limitante do sistema discutido em relação à figura 4;

a figura 6 ilustra com detalhes como pacotes de solicitação de entrada / saída (IRPs) fluem através de uma pilha de acionador de dispositivo (um componente do sistema da figura 4);

a figura 7 ilustra como as operações de ligar e usar com mudança de estado e de gerenciamento de energia podem ser tratadas; e

a figura 8 ilustra, de forma resumida, um método exemplar e não limitante para produzir um mecanismo para gerenciar operações em um ambiente de máquina virtual.

Descrição Detalhada

Visão Geral

O assunto em questão agora divulgado, ao mesmo tempo em que resolve os problemas expostos na seção de Antecedentes da Invenção, habilita pelo menos o seguinte: (a) que uma partição confiável (que, freqüentemente, é a partição pai) possa realizar tarefas por toda a máquina, tais como a detecção de dispositivos plugados, hibernação ou desativação do sistema, e assim por diante; (b) que tarefas por toda a máquina possam ser realizadas tanto em uma partição quanto na própria camada do administrador da máquina; (c) o controle de um ou mais dispositivos físicos por uma partição não confiável, em que tal controle não faça com que outro software em execução na máquina trave; e (d) impedir a manutenção de toda a máquina como refém, impedir hibernação, desativação ou inserção de novos dispositivos.

Além do mais, o assunto em questão agora divulgado permite interceptação de mensagens em uma partição pai associada com um dispositivo físico e o redirecionamento de tais mensagens para uma partição filha. Na partição filha, quando o acionador de dispositivo for carregado para o dispositivo físico, este acionador pode ser envolto por um outro acionador que intercepta mensagens que vêm do administrador de Ligar e Usar da partição filha ou de outro administrador, substituindo-as por mensagens que correspondem a decisões de política tomadas na partição pai. Certamente, esta é meramente uma visão geral exemplar e não limitante, e vários outros aspectos relacionados a ela são aqui divulgados.

Máquinas Virtuais em Termos Gerais

A figura 1 é um diagrama que representa a disposição em camadas lógicas da arquitetura de hardware e de software para um ambiente virtualizado em um sistema de computador. Na figura 1, um programa de virtualização 110 é executado direta ou indiretamente na arquitetura de hardware físico 112. O programa de virtualização 110 pode ser (a) um monitor de máquina virtual que é executado ao lado de um sistema operacional hospedeiro ou (b) um sistema operacional hospedeiro com um componente hipervisor, em que o componente hipervisor realiza a virtualização. O programa de virtualização 110 virtualiza uma arquitetura de hardware hóspede 108 (mostrada como linhas tracejadas para ilustrar o fato de que este componente é uma partição ou uma "máquina virtual"), isto é, hardware que não

existe realmente, mas, em vez disto, é virtualizado pelo programa de virtualização 110. Um sistema operacional hóspede 106 executa a arquitetura de hardware hóspede 108, e uma aplicação de software 104 pode ser executada no sistema operacional hóspede 106. No ambiente operacional virtualizado da figura 1, a aplicação de sistema eletrônico 105 pode ser executada em um sistema de computador 102 mesmo se a aplicação de software 104 for projetada para ser executada em um sistema operacional que é, no geral, incompatível com um sistema operacional hospedeiro e com a arquitetura de hardware 112.

A seguir, a figura 2 ilustra um sistema operacional virtualizado que compreende uma camada de software do sistema operacional hospedeiro (OS hospedeiro) 204 em execução diretamente acima do hardware de computador físico 202, em que o OS hospedeiro 204 fornece acesso aos recursos do hardware de computador físico 202 pela exposição de interfaces às partições A 208 e B 210 para uso pelos sistemas operacionais A e B, 212 e 214, respectivamente. Isto habilita o OS hospedeiro 204 a passar despercebido pelas camadas do sistema operacional 212 e 214 em execução acima dele. Novamente, para realizar a virtualização, o OS hospedeiro 204 pode ser um sistema operacional especialmente desenhado com capacidades de virtualização nativas ou, alternativamente, ele pode ser um sistema operacional padrão com um componente hipervisor incorporado para realizar a virtualização (não mostrado).

Novamente em relação à figura 2, acima do OS hospedeiro 204 estão duas partições, partição A 208, que pode ser, por exemplo, um processador Intel 386 virtualizado, e partição B 210, que pode ser, por exemplo, uma versão virtualizada de um processador da família Motorola 680X0 de processadores. Em cada partição 208 e 210 estão sistemas operacionais hóspedes (OSs hóspedes) A 212 e C 214, respectivamente. Em execução no topo do OS hóspede A 212 estão duas aplicações, aplicação A1 216 e aplicação A2 218, e em execução no topo do OS hóspede B 214 está a aplicação B1 220.

Em relação à figura 2, é importante notar que a partição A 208 e a partição B 214 (que são mostradas em linhas tracejadas) são representações de hardware de computador virtualizado que existem somente como construções de software. Elas são tornadas possíveis em função da execução de software(s) de virtualização especializado(s) que não somente apresenta(m) a partição A 208 e a partição B 210 ao OS hóspede A 212 e ao OS hóspede B 214, respectivamente, mas que também realiza(m) todas as etapas de software necessárias para que o OS hóspede A 212 e o OS hóspede B 214 interajam indiretamente com o hardware de computador físico real 202.

A figura 3 ilustra um sistema de computação virtualizado alternativo em que a virtualização é realizada por um monitor de máquina virtual (VMM) 204' em execução ao lado do sistema operacional hospedeiro 204". Em certos casos, o VMM 204' pode ser uma aplicação em execução acima do sistema operacional hospedeiro 204" e que interage com o

hardware do computador 202 somente por meio do sistema operacional hospedeiro 204". Em outros casos, da forma mostrada na figura 3, o VMM 204' pode, em vez disto, compreender um sistema de software parcialmente independente que, em alguns níveis, interage indiretamente com o hardware do computador 202 por meio do sistema operacional hospedeiro 204", mas, em outros níveis, o VMM 204' interage diretamente com o hardware de computador 202 (similar à maneira que o sistema operacional hospedeiro interage diretamente com o hardware de computador). E em ainda outros casos, o VMM 204' pode compreender um sistema de software completamente independente que, em todos os níveis, interage diretamente com o hardware de computador 202 (similar à maneira que o sistema operacional hospedeiro interage diretamente com o hardware de computador) sem utilizar o sistema operacional hospedeiro 204" (embora ainda interagindo com o sistema operacional hospedeiro 204" a fim de coordenar o uso do hardware de computador 202 e evitar conflito e congêneres).

Todas estas variações para implementar as partições supramencionadas são apenas implementações exemplares, e nada aqui deve ser interpretado como limitante da divulgação em nenhum aspecto da virtualização em particular.

Aspectos do Gerenciamento de Estado do Dispositivo e do Sistema em Máquinas Virtuais

As partições expostas das figuras 2 e 3 podem ser implementadas na forma de partições primárias e partições secundárias ou, em um aspecto não limitante, como partições confiáveis e partições não confiáveis (não privilegiadas), respectivamente, ou, em um ainda outro aspecto não limitante, como partições pais e partições filhas, respectivamente. Qualquer que seja o relacionamento estabelecido entre tais partições, um objetivo da presente divulgação é permitir a atribuição de dispositivos (sejam físicos ou virtuais) de tais partições primárias para partições secundárias, de forma que as partições secundárias possam exercer controle sobre os dispositivos, ainda, permitindo que as partições primárias tenham entrada em relação a como tais dispositivos podem e irão comportar.

Partições primárias podem ter objetos acionadores proxy que substituem proxies para objetos acionadores de dispositivo em partições secundárias. Tipicamente, tais objetos acionadores de dispositivo residentes nas pilhas de acionador podem ser circundados no topo e na base por filtros para garantir que tais objetos acionadores de dispositivo estejam se comportando apropriadamente de acordo com políticas implementadas pelas partições primárias. Tais políticas e outras instruções (incluindo código e/ou dados) podem ser comunicadas pelos objetos acionadores proxy a qualquer objeto na pilha de acionador na partição secundária.

Por exemplo, em um aspecto não limitante da presente divulgação, a figura 4 ilustra um sistema para gerenciar operações em um ambiente de máquina virtual, desse modo,

permitindo o gerenciamento de estado de hardware distribuído. Em outras palavras, ele ilustra, por exemplo, como o objeto acionador proxy 404 na partição primária 400 pode comunicar com os objetos 408, 406, 410 na pilha de acionador 416, de maneira tal que, mesmo embora o objeto acionador de dispositivo 406 mantenha controle de um dispositivo 412, o objeto acionador proxy 404 ainda tenha entrada considerando como um dispositivo 412 como este será controlado em virtude da maneira pela qual os filtros 408, 410 apresentam instruções ao objeto acionador de dispositivo 406.

Por exemplo, o segundo filtro 408 pode encaminhar instruções do objeto acionador proxy 404 ao objeto acionador de dispositivo 406, desse modo, tendo uma declaração sobre como o acionador de dispositivo 406 se comportará (e, portanto, como o dispositivo 412 que o objeto acionador de dispositivo 406 está controlando se comportará). Ou, alternativamente, o segundo filtro 408 pode interceptar instruções emitidas a partir de qualquer lugar na segunda partição 402 e tanto passá-las ao longo do objeto acionador de dispositivo 406 quanto substituir tais instruções por outras instruções – que podem estar em conformidade com algumas políticas ajustadas pela partição primária 400 (em que, por exemplo, tais instruções podem ser fornecidas pelo objeto acionador proxy 404).

Assim, conforme a figura 4, em um aspecto do assunto em questão agora divulgado, um objeto acionador proxy 404 localizado em uma partição primária 400 pode ser usado, em que o objeto acionador proxy 404 é um acionador proxy para o dispositivo 412. Em outras palavras, estritamente falando, ele não é o objeto acionador para o dispositivo 412, mas, em virtude de ele poder comunicar com qualquer um dos filtros 408, 410 na pilha de acionador 416, e de estes filtros 408, 410 poderem passar (ou não passar) instruções que o objeto acionador de dispositivo real 406 verá, ele é um acionador de dispositivo para o dispositivo 412, embora por proxy.

Além do mais, a figura 4 mostra um objeto acionador de dispositivo 406 que fica localizado em uma pilha de acionador 416 na partição secundária 402, em que o objeto acionador de dispositivo 406 é configurado para controlar o dispositivo 412. Percebe-se que há um primeiro objeto de filtro 410 localizado abaixo do objeto acionador de dispositivo 416 na pilha de acionador 416. O primeiro objeto de filtro 406 tem muitas funcionalidades, uma das quais pode ser a apresentação de uma interface ao objeto acionador de dispositivo 406 de forma que o objeto acionador de dispositivo 406 possa encaixar em funções relacionadas a barramento, incluindo o controle do dispositivo 412. Certamente, tais funções relacionadas a barramento não são limitadas somente ao controle do dispositivo 412, mas podem incluir a obtenção de informação que o objeto acionador de dispositivo 406 solicita, tal como, por exemplo, o primeiro filtro 410 pode inquirir o acionador proxy 404 sobre informação de espaço de endereço para o dispositivo 412.

Adicionalmente, um segundo objeto de filtro 408 localizado acima do objeto aciona-

dor de dispositivo 406 na pilha de acionador 416 pode ser configurado para (a) direcionar um primeiro conjunto de instruções do objeto acionador proxy 404 para o objeto acionador de dispositivo 406 e/ou (b) interceptar um segundo conjunto de instruções proposto para o objeto acionador de dispositivo 406, em que o as segundas instruções podem ser originadas da partição secundária 402. Isto significa que o segundo filtro 408 pode ter uma funcionalidade dual: servir como um mecanismo que retransmite instruções do acionador proxy 404 para o acionador de dispositivo 406, e como um mecanismo que intercepta instruções emitidas para o acionador de dispositivo 406 – em que, tipicamente, estas últimas instruções serão emitidas pela partição secundária 402, embora elas possam ser emitidas com a mesma facilidade por uma outra partição, tal como uma partição terciária não confiável.

Neste último caso, mediante interceptação das segundas instruções, o segundo objeto de filtro 408 pode comparar as segundas instruções em relação às políticas ajustadas pela partição primária 400 e/ou por um módulo de virtualização, tais como um hipervisor, um monitor de máquina virtual, e assim por diante (mostrado nas figuras 2 e 3). Se as segundas instruções conflitarem com aquelas políticas, o segundo objeto de filtro 408 pode substituir estas segundas instruções por instruções que são consistentes com as políticas. Então, o segundo objeto de filtro 408 pode transmitir estas instruções consistentes ao objeto acionador de dispositivo 406.

Em um aspecto do assunto em questão agora divulgado, conforme a figura 4, a partição primária 400 pode ter um objeto 414 (tal como um objeto administrador de ligar e usar e/ou um objeto administrador de energia, ou qualquer outro objeto que pode emitir instruções) que comunica com o objeto de dispositivo proxy 404, em que o objeto 414 da partição primária 400 direciona o objeto de dispositivo proxy 404 a implementar eventos de mudança de estado. Tais eventos de mudança de estado podem ser conduzidos por meio das primeiras instruções. É desnecessário dizer que as primeiras instruções podem ser usadas para emitir tanto eventos para toda a máquina, tais como desativação do sistema, ou modo de espera do sistema para toda a máquina física e para cada partição, quanto, alternativamente, eventos mais locais, tais como comandar o objeto acionador de dispositivo 406 para desativar o dispositivo 412.

Em um outro aspecto, o sistema da figura 4 pode ter um objeto de partição secundária 415 (novamente, tais como um objeto administrador de ligar e usar e/ou um objeto administrador de energia, ou qualquer outro objeto que pode emitir instruções) que comunica com o segundo objeto de filtro 406 por meio das segundas instruções supramencionadas. O objeto de partição secundária 415 pode direcionar o segundo objeto de filtro 408 para implementar instruções relacionadas a um evento de ligar e usar (inserção de um dispositivo de armazenamento em massa, de uma câmera digital, etc.) e/ou a um evento de energia (desativação, hibernação, etc.).

Tanto no caso de interceptação quanto no caso do direcionamento, se o objeto acionador de dispositivo 406 recusar em se conformar com pelo menos uma política ajustada pela partição primária 400 (a saber, um módulo anexo, tal como o administrador 414), o controle do dispositivo 412 pelo objeto acionador de dispositivo 406 pode ser revogado. A revogação pode depender de várias variáveis, tais como um conjunto de regras estritas ou diretrizes observadas no geral, ou qualquer coisa similar a isto. Versados na técnica percebem prontamente a multiplicidade de vários regimes de verificação de política que pode ser implementada com o presente aspecto divulgado.

Estes são aspectos meramente exemplares. Por exemplo, entende-se que cada relacionamento de comunicação unidirecional pode exigir seu complemento. Em outras palavras, já que o acionador proxy 404 pode comunicar com objetos 408, 406, 410 na pilha de dispositivo, entende-se que comunicação oposta também é aqui contemplada. Por exemplo, o acionador proxy 404 também pode receber comunicações dos objetos de partição secundária 402, tais como solicitações para acesso ao espaço de configuração e a outros recursos que não podem ser mapeados na memória ou no espaço I/O da partição secundária 402.

Implementações Exemplares do Gerenciamento das Operações em Máquinas Virtuais

Vários sistemas operacionais são aqui contemplados para uso nas máquinas virtuais e partições mostradas nas figuras 2, 3 e 4 até agora. Por exemplo, tipicamente, o Windows realiza I/O pela construção de pilhas dos objetos acionadores de dispositivo que, cada qual, desempenha um papel no gerenciamento e no controle dos dispositivos. Cada pacote de solicitação I/O (IRP) individual 418 pode ser transmitido ao objeto acionador de topo (por exemplo, segundo objeto de filtro 408) em uma pilha (por exemplo, pilha de acionador 406), que passa o IRP para baixo da pilha quando tiver finalizado com ele. Por sua vez, cada objeto acionador recebe o IRP e realiza o processamento apropriado àquela camada na pilha. No geral, as camadas mais de base são preocupadas com o ajuste e com a configuração. Por exemplo, na figura 5, esta camada de base corresponderá aos objetos "PCL.sys", "VMBus.sys" e "ACPI.sys" (estes acrônimos dizem respeito a um objeto "interconexão de componente periférico" (PCI), um objeto "barramento de máquina virtual" (VMBus), e um objeto "interface avançada de configuração e energia" (ACPI)). Versados na técnica percebem prontamente que estes e outros acrônimos são aqui usados.

No Windows, dispositivos físicos (por exemplo, dispositivo 412) são distribuídos pelos acionadores de barramento (por exemplo, "PCI.sys") que possuem o gerenciamento de todo um barramento em um sistema de computador físico. Um dispositivo pode ser ligado em um barramento "PCI Express" no sistema e ele pode ser descoberto por um acionador de barramento "PCI Express" no Windows. É este acionador ("PCI.sys") que pode se envol-

ver em um protocolo com um administrador, tal como um administrador de Ligar e Usar (PnP), o que fará com que um outro acionador (chamado de um Acionador Funcional, ou FDO) seja carregado. Este FDO (por exemplo, acionador de dispositivo 406) controla realmente o dispositivo. "PCI.sys" também pode expor interfaces que são específicas à configuração de um dispositivo PCI Express, permitindo que o FDO manipule o dispositivo durante ajuste e configuração.

Quando um acionador de barramento descobrir um novo dispositivo, ele pode criar um novo objeto de dispositivo, criando a base de uma nova pilha de dispositivo. O administrador PnP pode transmitir uma série de solicitações à nova pilha, a identificando e coletando uma série de propriedades. A partir desta informação, ele pode decidir quais acionadores dispor no topo da camada do objeto de dispositivo que foi criado pelo acionador de barramento.

Com este contexto em vigor, a seguir, a figura 5 ilustra uma implementação exemplar e não limitante do assunto em questão discutido na figura 4. Na figura 5, uma série de pilhas de dispositivo com uma série de barramentos PCI é mostrada em uma partição pai (ou confiável) 500 e em uma partição filha (ou não confiável / não privilegiada) 502. Estas são partições meramente exemplares que correspondem às partições primária 400 e secundária 402 da figura 4, respectivamente. Outras partições em vários relacionamentos confiáveis e não confiáveis podem ser usadas. As presentes pilhas de dispositivo divulgadas podem ser de vários tipos, por exemplo: um controlador de vídeo 508, um controlador IDE 510, primeira pilha de rede 504, uma segunda pilha de rede 506, uma pilha de ponte PCI 512, e uma pilha de raiz PCI 514. Versados na técnica percebem prontamente a miríade de diferentes pilhas de dispositivo que pode ser usada em um ambiente de máquina virtual típico.

Na figura 5, pilhas de dispositivo são representadas por retângulos, e objetos de dispositivo por ovais. Como exposto, o objeto de dispositivo mais de base em uma pilha pode ser o acionador de barramento que, na figura 5, é o objeto "PCI.sys", exceto no caso do barramento raiz PCI. Neste exemplo, o barramento raiz pode ser descoberto considerando o software embarcado ACPI, que é o motivo pelo qual "ACPI.sys" está na base da pilha raiz PCI 514. Os outros objetos acionadores, por exemplo, "Net.sys" na pilha de Rede 1 504 ou "Vídeo" na pilha de Vídeo 508 estão no topo do objeto PCI.sys e eles podem controlar os dispositivos físicos reais.

Neste exemplo, a partição pai 500 pode controlar todos os dispositivos na máquina. Mesmo quando ela atribuir um dispositivo a uma outra partição, tal como a partição filha 502, ela pode reter alguma medida de controle sobre os dispositivos de forma que a partição pai não possa impedir as operações que envolvem todo o barramento ou toda a máquina. Como foi discutido anteriormente, uma maneira para a partição pai 500 retém tal controle é pelo uso de filtros acima e abaixo dos acionadores de partição filha e pelo uso e manipula-

ção de vários conjuntos de instruções associados com eles.

Por exemplo, se a partição pai 500 desejar permitir que uma partição filha 502 controle diretamente um dispositivo, talvez, uma interface de rede, ela pode, primeiro, descarregar todos os acionadores que estão atualmente controlando aquele dispositivo na partição pai 500 (para evitar qualquer conflito com um acionador na partição filha). Então, ela pode carregar um acionador diferente na partição pai 500, uma que existe para gerenciar solicitações que são transmitidas à partição filha. Na figura 5, esta função é realizada pelo objeto "VMRedir.sys" na pilha de Rede 2 506. Assim, como foi sugerido anteriormente, o objeto "VMRedir.sys" na partição pai 500 é um objeto proxy para o objeto "Net.sys" na pilha de Rede 2 516 da partição filha 502.

Na partição filha 502, um acionador pode construir uma pilha de dispositivo que corresponde a um dispositivo físico (por exemplo, um cartão de interface de rede (NIC)). Isto pode ser feito por qualquer conjunto de acionadores de dispositivo com detecção de máquina virtual, mas, na figura 5, é mostrado como se fosse feito pelo objeto "VMBus.sys", que pode ser um mecanismo para gerenciar I/O de partição cruzada (veja SN: 11/128647, "Partition Bus", Documento Judicial MSFT – 4771). No topo deste objeto acionador de barramento, é disposto em camadas um objeto de dispositivo ("VMPCI.sys") que gerencia dispositivos PCI Express que foram atribuídos à partição filha. No topo deste, é carregado o acionador regular (FDO) para este NIC do PCI Express ("Net.sys"). E, no topo deste, um outro objeto de dispositivo é disposto em camadas (novamente, "VMPCI.sys") de forma que ele possa interceptar alguns IRPs transmitidos a esta pilha a partir da partição secundária e/ou redirecionar instruções provenientes da partição pai 500. Tanto o primeiro filtro 408 quanto o segundo filtro 410 da figura 4 podem corresponder ao objeto "VMPCI.sys" da figura 5.

A seguir, a figura 6 ilustra com mais detalhes como IRPs 602 fluem através de uma pilha de acionador de dispositivo 516 mostrada na figura 5. As linhas com setas de IRPs 602 representam o fluxo do Administrador PnP e/ou do Administrador de Energia 500 (que podem ser ou podem não ser entidades separadas). O objeto de dispositivo de topo (parte do "VMPCI.sys") intercepta estes IRPs e os trata de uma maneira que faz com que a partição secundária 502 acredite que a pilha de dispositivo 516 tenha seguido a orientação de todas as políticas que vêem da partição secundária. As linhas com setas também mostram o fluxo de IRP 602 que pode ser observado pelo FDO, a saber, "Net.sys", e elas mostram IRPs que fluem para baixo até o acionador de barramento ("VMBus.sys").

A figura 7 mostra como os IRPs 702 de PnP com mudança de estado e de Gerenciamento de Energia 700 (mostrados em linhas pontilhadas) podem ser transmitidos na partição primária 500, e como estes IRPs 704 (novamente, mostrados em linhas pontilhadas) são encaminhados à partição secundária 502. Em essência, estes IRPs 704 são redirecionados do objeto "PCI.sys" na pilha de rede 506 da partição primária 500 ao objeto

"VMPCI.sys" na pilha de rede 516 da partição secundária 502. Lá, o objeto "VMPCI.sys" (isto é, filtro) pode passá-los para baixo até o acionador de dispositivo "Net.sys" ou usá-los como regras ou diretrizes para interceptar todos os IRPs 706 da partição secundária 502.

Em um aspecto do assunto em questão agora divulgado, IRPs 702 que são transmitidos na partição primária 500 podem fazer com que mensagens sejam transmitidas à partição secundária 502. Aquelas mensagens podem ser retornadas em IRPs pelo VMPCI.sys e transmitidas a um acionador de NIC (a saber, "Net.sys"). O acionador de NIC pode transmiti-los para baixo (com supõe-se) até o objeto "VMPCI.sys", e este objeto pode capturá-los naquele local e interrompê-los de ir adicionalmente para baixo na pilha. Neste ínterim, todos os IRPs com mudança de estado 706 que podem originar na partição secundária 502 (mostrada, pelo menos em parte, em linhas tracejadas) podem ser transmitidos para baixo na pilha até o acionador de NIC (a saber, "Net.sys") ou eles podem ir e voltar depois dele, já que as camadas acima e abaixo de "Net.sys" são os objetos "VMPCI.sys".

Além do mais, todas as tentativas de gerenciar dispositivos no nível do barramento que originam com "Net.sys" podem ser transmitidas para baixo na pilha até "VMPCI.sys", e elas podem ser retransmitidas à partição primária 500 e progressivamente até o objeto "PCI.sys". Exemplos de gerenciamento no nível de barramento podem incluir uma solicitação para ler o espaço de configuração PCI Express do dispositivo de rede. Versados na técnica, lendo a presente divulgação, percebem as várias outras permutações que os aspectos supradiscutidos podem tomar. Não pretende-se que nenhum destes aspectos seja limitante, mas meramente exemplar.

Resumo

Em resumo, vários sistemas de gerenciamento de operações em um ambiente de máquina virtual foram discutidos. Estes sistemas, como é aparente aos versados na técnica, podem ser implementados como métodos ou instruções que subsistem em mídia legível por computador. Por exemplo, um método exemplar e não limitante pode compreender as seguintes etapas (em que a ordem destas etapas pode ser implementada em várias permutações). Na caixa 800, em uma partição, uma pilha de dispositivo pode ser construída correspondente a um dispositivo físico. Então, na caixa 802, um primeiro objeto de filtro pode ser disposto em camadas na pilha de dispositivo, em que este filtro apresenta uma interface a um objeto acionador de dispositivo para o objeto se envolver em operações relacionadas a barramento. A seguir, na caixa 804, o objeto acionador de dispositivo pode ser disposto em camadas no topo do primeiro objeto de filtro, em que o objeto acionador pode controlar o dispositivo físico por meio do primeiro objeto de filtro. Finalmente, na caixa 806, um segundo objeto de filtro pode ser disposto em camadas no topo do objeto acionador de dispositivo, em que o segundo objeto de filtro pode ser configurado para direcionar um primeiro conjunto de instruções de um objeto acionador proxy à pilha de dispositivo, em que o acionador proxy

pode ficar residente em uma partição diferente desta partição.

Agora que a pilha de dispositivo com um primeiro objeto de filtro, um objeto acionador de dispositivo, e um segundo objeto de filtro está construída, esta pilha de dispositivo pode interagir com um objeto acionador proxy na outra partição. Assim, na caixa 808, a pilha de dispositivo pode processar instruções, IRPs, mensagens e congêneres (essencialmente, qualquer informação, seja código e/ou dados). Por exemplo, na caixa 810, ela pode processar o supramencionado primeiro conjunto de instruções proposto para o objeto acionador de dispositivo, em que as segundas instruções originam da partição (isto é, elas são instruções intrapartição da mesma partição da pilha - em oposição a ser interpartição, portanto, originada de uma partição diferente), e em que estas segundas instruções podem ser relacionadas a eventos de ligar e usar e de energia. Por último, na caixa 814, o primeiro filtro pode fornecer uma interface ao objeto acionador e pode obter a informação necessária de uma partição diferente para o objeto acionador (por exemplo, informação relacionada a barramento).

Um outro aspecto do que foi descrito até aqui pode incluir um conceito dual de controle. Por exemplo, o conceito de controle pode ser separado como controle do estado do dispositivo, que diz respeito à operação do próprio dispositivo, e como controle do estado do dispositivo que afeta todo o sistema. Por exemplo, o caso anterior pode incluir informação associada com operações I/O individuais (por exemplo, o deslocamento em um disco que está sendo atualmente lido). E, além do mais, o último caso pode incluir o interruptor elétrico para o dispositivo, em que ele não pode ser ligado a menos que todo o sistema também esteja energizado. Assim, neste aspecto, o que ocorre é uma separação do estado do dispositivo que diz respeito à I/O e sua atribuição a uma partição secundária, ainda mantendo controle do estado do dispositivo que diz respeito a todo o sistema na partição primária.

Os métodos, sistemas, aparelhos do assunto em questão agora divulgado também podem ser incorporados na forma de código de programa (tais como instruções legíveis por computador) que é transmitido sobre a mesma mídia de transmissão, tais como sobre fiação ou cabeamento elétrico, por meio de fibras óticas, ou por meio de qualquer outra forma de transmissão, em que, quando o código de programa for recebido, carregado e executado em uma máquina, tais como em uma EPROM, em um arranjo de porta, em um dispositivo lógico programável (PLD), em um computador cliente, tal como aquele mostrado na figura a seguir, em um gravador de vídeo ou congêneres, a máquina se transforma em um aparelho para praticar o presente assunto em questão. Quando implementado em um processador de uso geral, o código de programa combina com o processador para fornecer um aparelho exclusivo que opera para realizar a funcionalidade de salvar e restaurar do presente assunto em questão.

Por último, embora a presente divulgação tenha sido descrita em conjunto com os

aspectos preferidos, da forma ilustrada nas várias figuras, entende-se que outros aspectos similares podem ser usados, ou que modificações e adições podem ser feitas nos aspectos descritos para realizar a mesma função da presente divulgação sem dela desviar. Por exemplo, em vários aspectos da divulgação, o gerenciamento das operações em um ambiente de máquina virtual foi discutido. Entretanto, outros mecanismos equivalentes a estes aspectos descritos também são contemplados pelos preceitos aqui expostos. Portanto, a presente divulgação não deve ser limitada a nenhum único aspecto, mas, em vez disto, deve ser interpretada em amplitude de escopo de acordo com as reivindicações anexas.

REIVINDICAÇÕES

1. Sistema para gerenciamento de operações em um ambiente de máquina virtual, **CARACTERIZADO** pelo fato de que compreende:

pelo menos um objeto acionador proxy localizado em uma primeira partição, em que o dito pelo menos um objeto acionador proxy é um acionador proxy para um dispositivo;

pelo menos um objeto acionador de dispositivo localizado em uma pilha de acionador em uma segunda partição, em que o dito pelo menos um objeto acionador é configurado para controlar o dito dispositivo;

pelo menos um primeiro objeto de filtro localizado abaixo do dito pelo menos um objeto acionador de dispositivo na dita pilha de acionador, em que o dito pelo menos um primeiro objeto de filtro apresenta uma interface ao dito pelo menos um objeto acionador de dispositivo para que o pelo menos um objeto acionador de dispositivo se envolva em funções relacionadas a barramento, incluindo o controle do dito dispositivo; e

pelo menos um segundo objeto de filtro localizado acima do dito pelo menos um objeto acionador de dispositivo na dita pilha de acionador, em que o dito pelo menos um segundo objeto é configurado para pelo menos um de (a) direcionar primeiras instruções do dito pelo menos um objeto acionador proxy ao dito pelo menos um objeto acionador de dispositivo e (b) interceptar segundas instruções propostas para o dito pelo menos um objeto acionador de dispositivo, em que as ditas segundas instruções originam da dita segunda partição.

2. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que compreende adicionalmente um primeiro objeto de partição que comunica com o dito pelo menos um objeto de dispositivo proxy, em que o dito primeiro objeto de partição direciona o dito pelo menos um objeto de dispositivo proxy para implementar pelo menos um evento de mudança de estado, em que o dito pelo menos um evento de mudança de estado é conduzido por meio das ditas primeiras instruções.

3. Sistema, de acordo com a reivindicação 2, **CARACTERIZADO** pelo fato de que o dito primeiro objeto de partição é pelo menos um de (a) um objeto administrador de ligar e usar e (b) um objeto administrador de energia.

4. Sistema, de acordo com a reivindicação 2, **CARACTERIZADO** pelo fato de que o dito pelo menos um evento de mudança de estado é um evento em todo o sistema.

5. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que compreende adicionalmente um segundo objeto de partição que comunica com o dito pelo menos um segundo objeto de filtro por meio das ditas segundas instruções, em que o dito segundo objeto de partição direciona o dito pelo menos um segundo objeto de filtro para implementar instruções relacionadas a pelo menos um de (a) um evento de ligar e usar e (b) um evento de energia.

6. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que, mediante interceptação das ditas segundas instruções, o dito pelo menos um segundo objeto de filtro compara as ditas segundas instruções em relação a pelo menos uma política ajustada por um de (a) a dita primeira partição e de (b) um módulo de virtualização, em que, se as ditas segundas instruções conflitarem com a dita pelo menos uma política, o dito pelo menos um segundo objeto de filtro substitui as ditas segundas instruções com instruções consistentes com a dita pelo menos uma política e as passa para baixo até o dito pelo menos um objeto acionador de dispositivo.

7. Sistema, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que se o dito pelo menos um objeto acionador de dispositivo recusar se conformar com pelo menos uma política ajustada pela dita primeira partição, o controle do dito dispositivo pelo dito pelo menos um objeto acionador de dispositivo é revogado.

8. Método para o gerenciamento de operações em um ambiente de máquina virtual, **CARACTERIZADO** pelo fato de que compreende:

construir, em uma partição, uma pilha de dispositivo que corresponde a um dispositivo físico;

dispor em camadas, na dita pilha de dispositivo, um primeiro objeto de filtro que apresenta uma interface a um objeto acionador de dispositivo;

dispor em camadas o dito objeto acionador de dispositivo no topo do dito primeiro objeto de filtro, em que o dito objeto acionador controla o dito dispositivo físico por meio do dito objeto de filtro; e

dispor em camadas um segmento objeto de filtro no topo do dito objeto acionador de dispositivo, em que o dito segundo objeto de filtro é configurado para direcionar um primeiro conjunto de instruções de um objeto acionador proxy para a dita pilha de dispositivo, em que o dito acionador de proxy fica residente em uma partição diferente da dita partição.

9. Método, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que compreende adicionalmente interceptar um segundo conjunto de instruções proposto para o dito objeto acionador de dispositivo, em que as ditas segundas instruções originam da dita partição.

10. Método, de acordo com a reivindicação 9, **CARACTERIZADO** pelo fato de que compreende adicionalmente, depois da dita interceptação, se as ditas segundas instruções conflitarem com políticas ajustadas por um módulo na dita partição diferente, substituir instruções alternativas.

11. Método, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que compreende adicionalmente remover um objeto acionador para o dito dispositivo físico da dita partição diferente.

12. Método, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que

compreende adicionalmente carregar o dito acionador proxy que gerencia solicitações da dita partição diferente, em que as ditas solicitações são passadas pelo dispositivo acionador proxy até a dita partição para realizar tarefas pelo dito objeto acionador de dispositivo.

13. Método, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que compreende adicionalmente redirecionar solicitações para o dito objeto acionador proxy à dita pilha de acionador, em que as ditas solicitações têm efeitos em todo o sistema.

14. Método, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que compreende detectar recusa pelo objeto acionador de dispositivo em participar de acordo com políticas ajustadas pela dita partição diferente, e se tal recusa ocorrer, revogar controle do dito dispositivo físico do dito objeto acionador de dispositivo.

15. Método, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que compreende designar a dita partição como uma partição não privilegiada e a dita partição diferente como uma partição confiável do dito ambiente de máquina virtual.

16. Mídia legível por computador que suporta instruções executáveis por computador implementadas em um dispositivo de computação físico, **CARACTERIZADA** pelo fato de que compreende:

instanciar objetos acionadores proxy em uma primeira partição, em que os ditos objetos acionadores proxy agem como proxies para dispositivos;

instanciar objetos acionadores de dispositivo em pilhas de acionador em uma segunda partição, em que os ditos objetos acionadores são configurados para controlar os ditos dispositivos;

instanciar os primeiros objetos de filtro abaixo dos ditos objetos acionadores de dispositivo nas ditas pilhas de acionador, em que os ditos primeiros objetos de filtro apresentam interfaces aos ditos objetos acionadores de dispositivo para os ditos objetos acionadores controlarem os ditos dispositivos; e

instanciar segundos objetos de filtro acima dos ditos objetos acionadores de dispositivo nas ditas pilhas de acionador, em que os ditos segundos objetos de filtro são configurados para pelo menos um de (a) direcionar a primeira partição, com base em instruções, para os ditos objetos acionadores de dispositivo e (b) interceptar a segunda partição com base em instruções propostas para os ditos objetos acionadores de dispositivo.

17. Mídia legível por computador, de acordo com a reivindicação 16, **CARACTERIZADA** pelo fato de que compreende adicionalmente descarregar todos os objetos acionadores para os ditos dispositivos da dita primeira partição.

18. Mídia legível por computador, de acordo com a reivindicação 16, **CARACTERIZADA** pelo fato de que compreende adicionalmente direcionar a dita primeira partição com base em instruções para os ditos objetos acionadores de dispositivo depois de solicitações pelos módulos de gerenciamento para os ditos objetos de dispositivo proxy, em

que os ditos módulos de gerenciamento formam um conjunto de módulos que inclui administradores de ligar e usar e administradores de energia.

19. Mídia legível por computador, de acordo com a reivindicação 16, **CARACTERIZADA** pelo fato de que compreende adicionalmente designar a dita primeira partição como uma partição confiável e a dita segunda partição como uma partição não confiável.

20. Mídia legível por computador, de acordo com a reivindicação 16, **CARACTERIZADA** pelo fato de que compreende interceptar a dita segunda partição com base em instruções propostas para os ditos objetos acionadores de dispositivo e examinar a dita segunda partição com base em instruções em relação à conformidade com a primeira partição com base em políticas, em que, mediante a dita interceptação, substitui a dita segunda partição com base em instruções com instruções em conformidade com as ditas políticas se ocorrer uma não conformidade entre a dita segunda partição com base em instruções e a dita primeira partição com base em políticas.

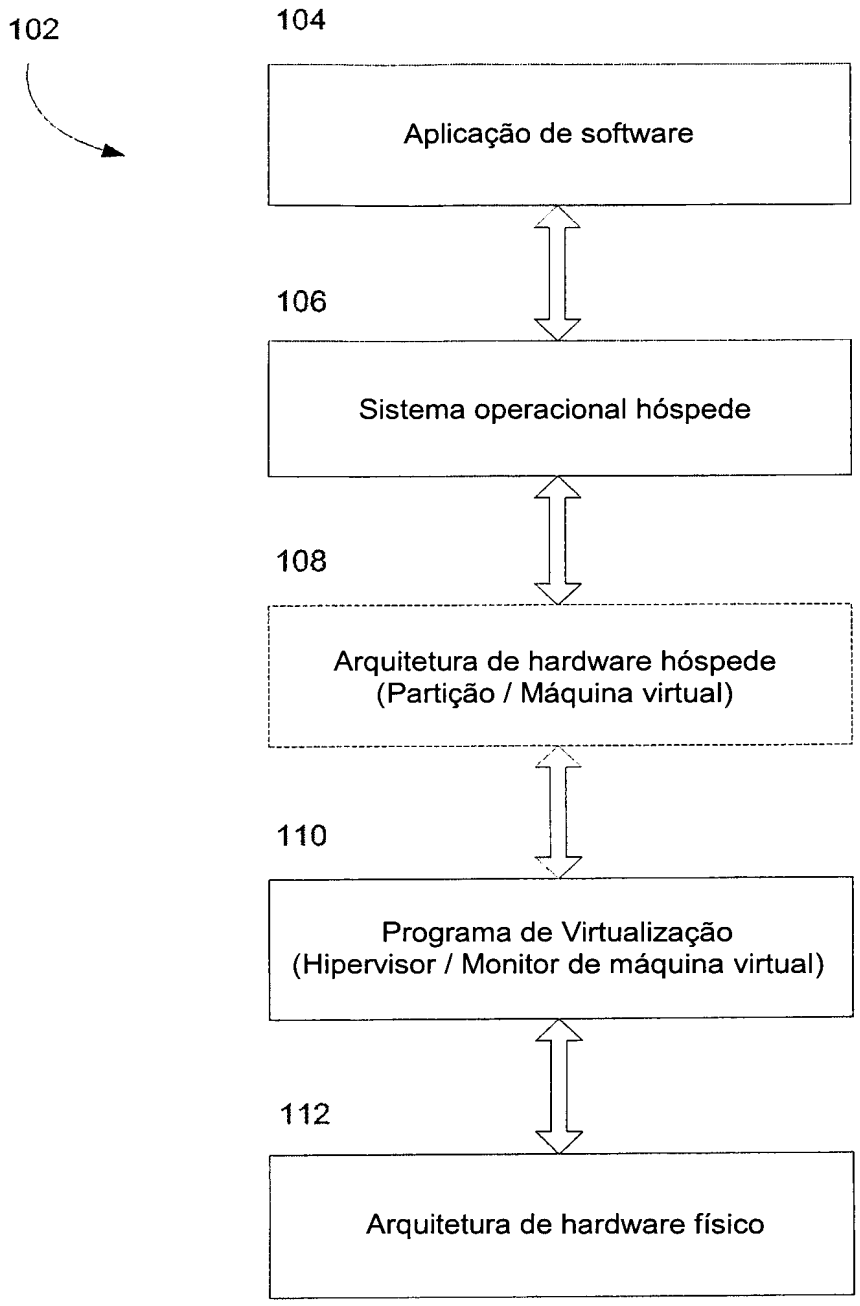

102
104
Aplicação de software
106
Sistema operacional hóspede
108
Arquitetura de hardware hóspede
(Partição / Máquina virtual)
110
Programa de Virtualização
(Hipervisor / Monitor de máquina virtual)
112
Arquitetura de hardware físico

Fig. 1

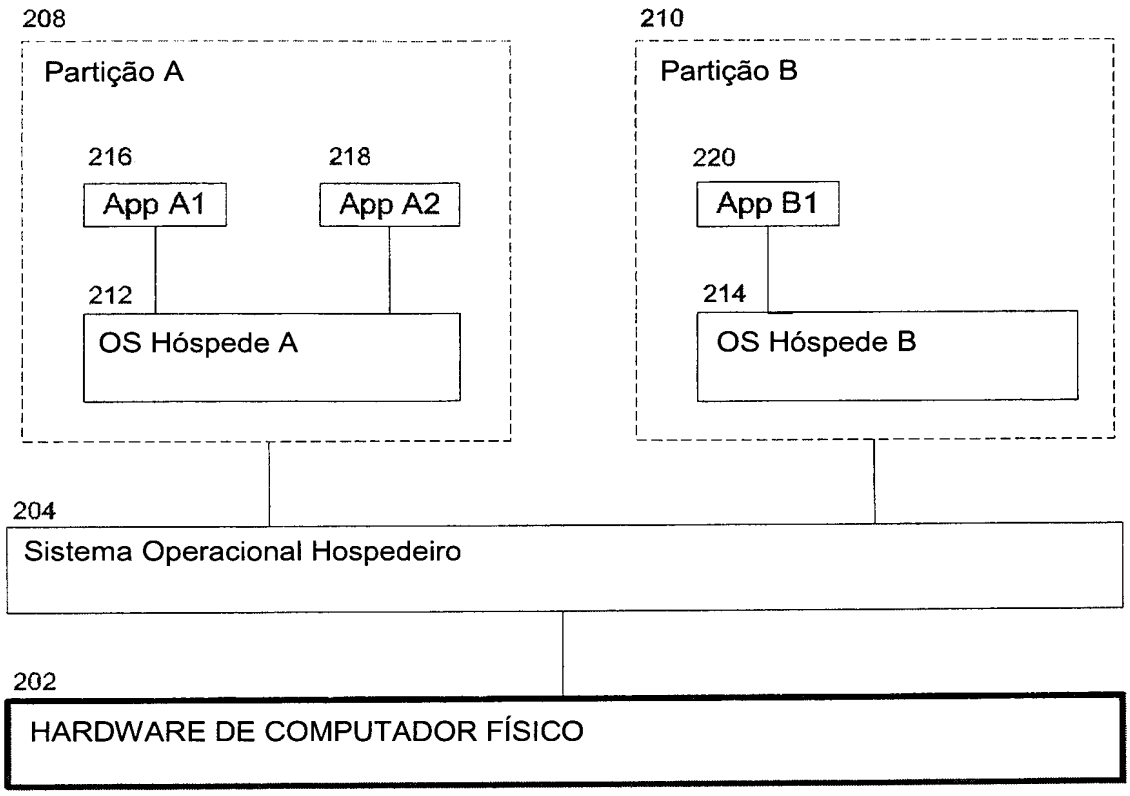

208
Partição A
216
App A1
218
App A2
212
OS Hóspede A
210
Partição B
220
App B1
214
OS Hóspede B
204
Sistema Operacional Hospedeiro
202
HARDWARE DE COMPUTADOR FÍSICO

Fig. 2

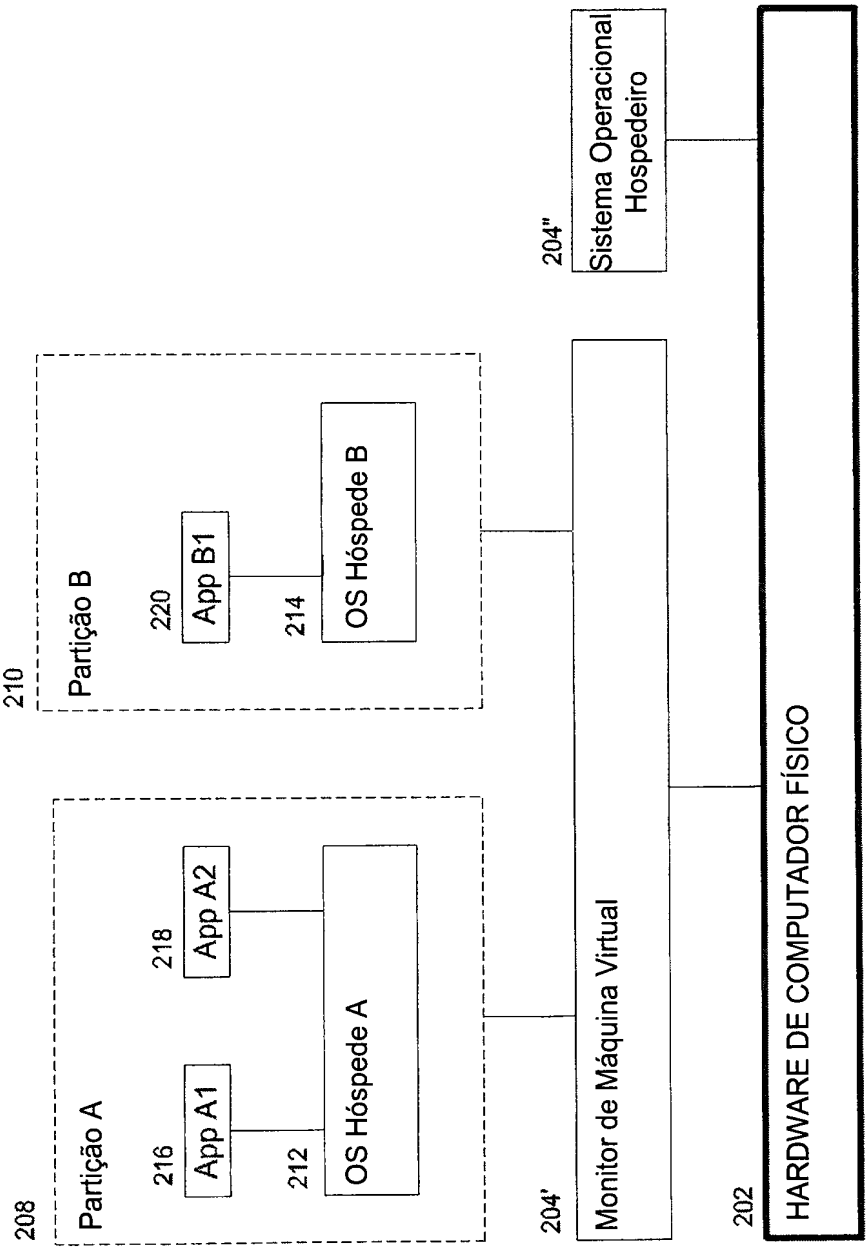

208
Partição A
216
App A1
218
App A2
212
OS Hóspede A
210
Partição B
220
App B1
214
OS Hóspede B
204'
Monitor de Máquina Virtual
204"
Sistema Operacional Hospedeiro
202
HARDWARE DE COMPUTADOR FÍSICO

Fig. 3

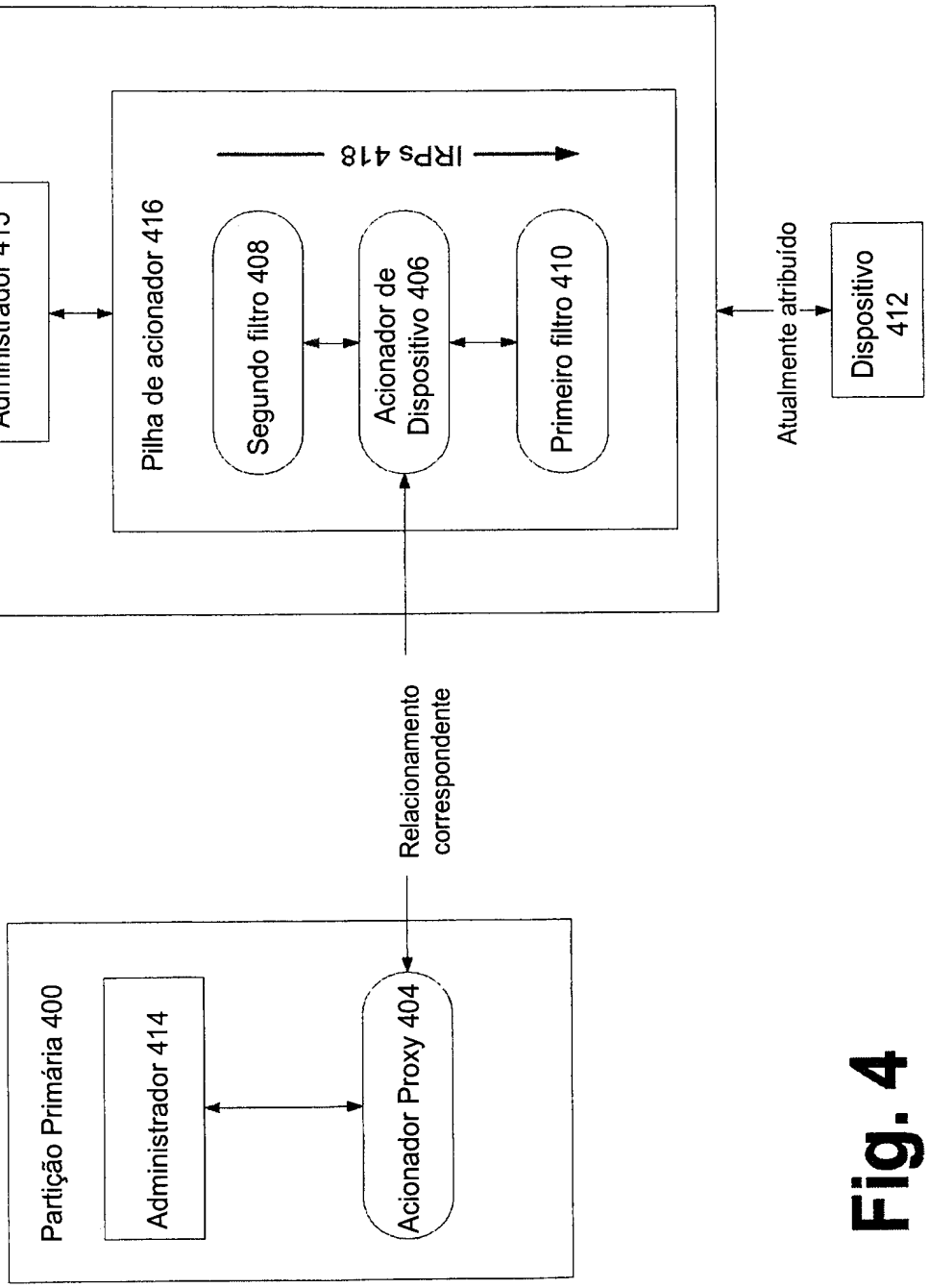

Partição Primária 400
Administrador 414
Acionador Proxy 404
Relacionamento correspondente
Partição Secundária 402
Administrador 415
Pilha de acionador 416
IRPs 418
Segundo filtro 408
Acionador de Dispositivo 406
Primeiro filtro 410
Atualmente atribuído
Dispositivo 412

Fig. 4

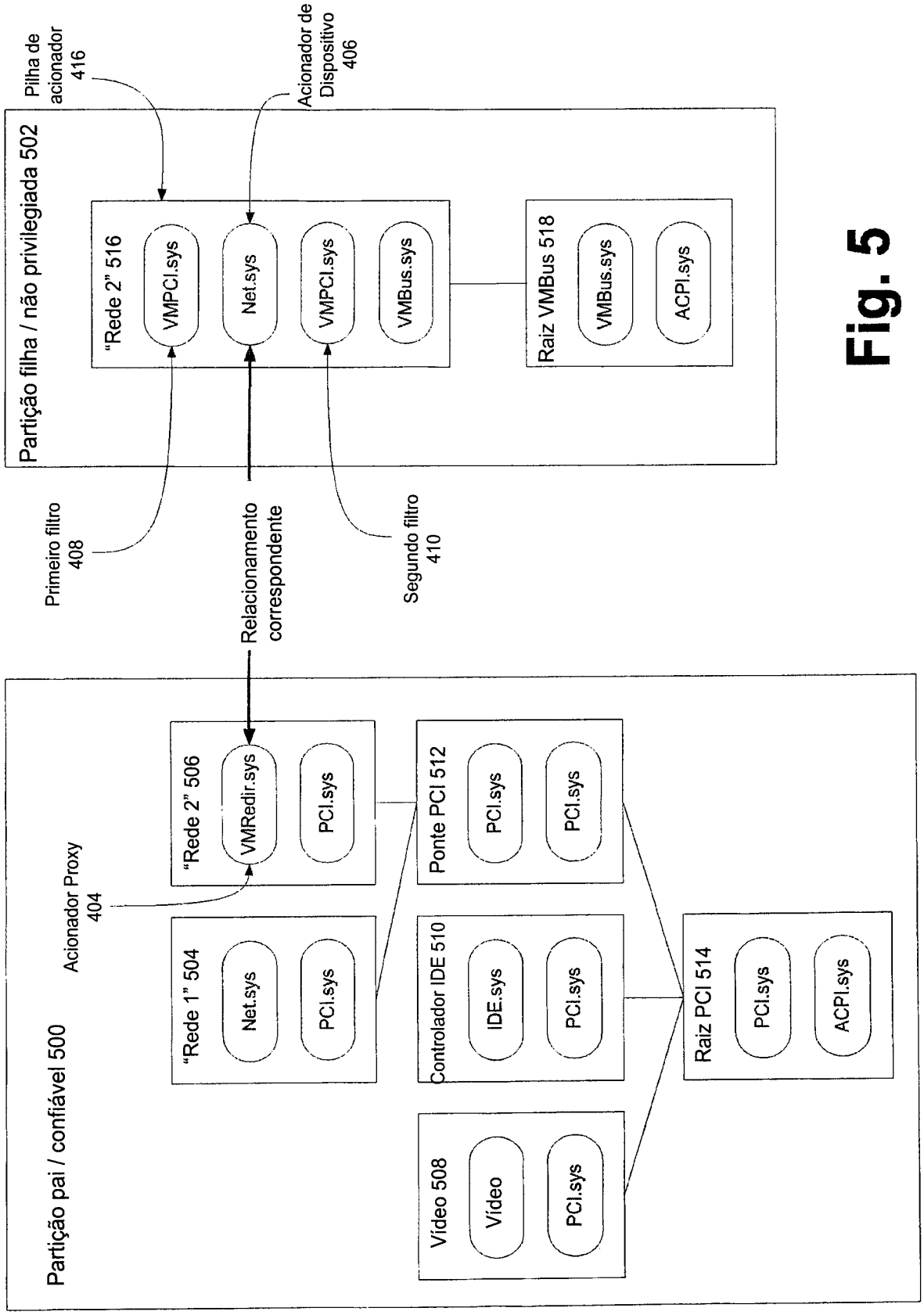

Partição filha / não privilegiada 502
Pilha de acionador 416
Acionador de Dispositivo 406
"Rede 2" 516
VMPCI.sys
Net.sys
VMPCI.sys
VMBus.sys
Raiz VMBus 518
VMBus.sys
ACPI.sys
Primeiro filtro 408
Relacionamento correspondente
Segundo filtro 410
Partição pai / confiável 500
Acionador Proxy 404
"Rede 2" 506
VMRedir.sys
PCI.sys
"Rede 1" 504
Net.sys
PCI.sys
Ponte PCI 512
PCI.sys
PCI.sys
Controlador IDE 510
IDE.sys
PCI.sys
Vídeo 508
Vídeo
PCI.sys
Raiz PCI 514
PCI.sys
ACPI.sys

Fig. 5

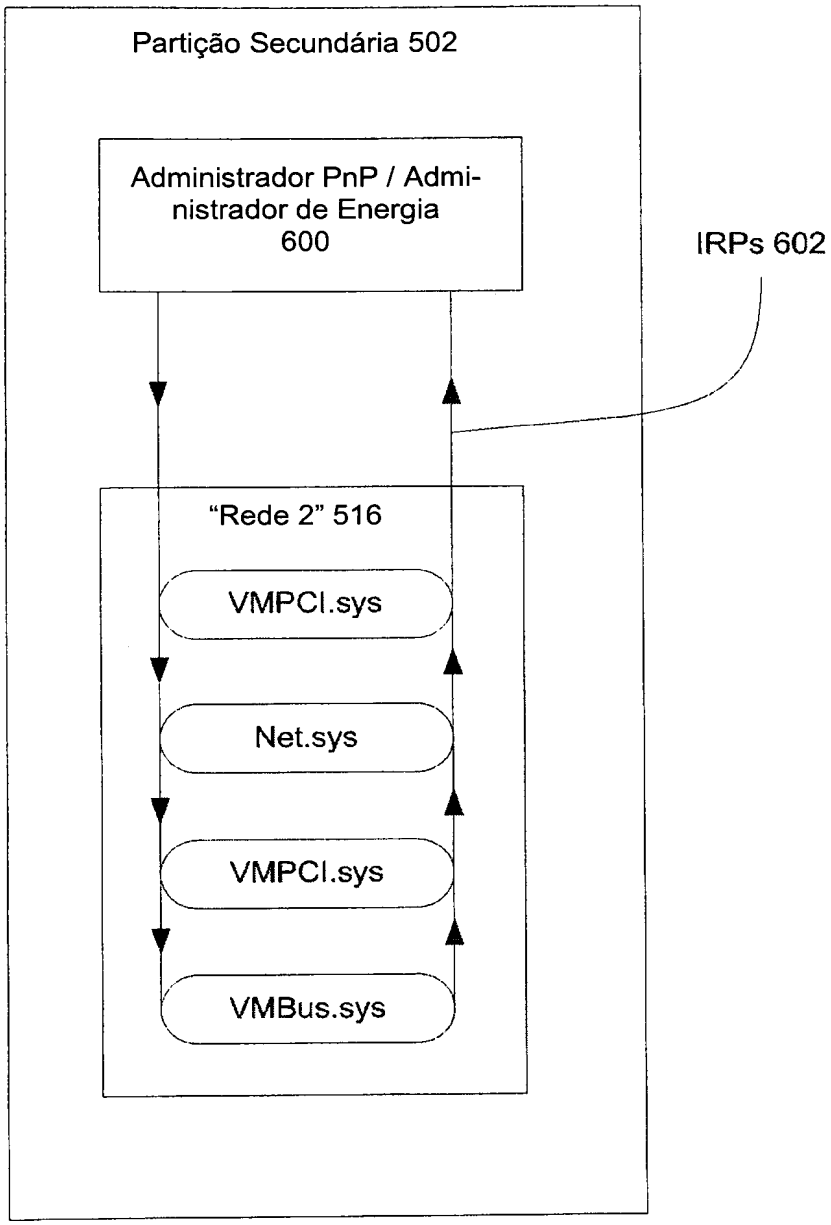

Partição Secundária 502
Administrador PnP / Administrador de Energia 600
IRPs 602
"Rede 2" 516
VMPCI.sys
Net.sys
VMPCI.sys
VMBus.sys

Fig. 6

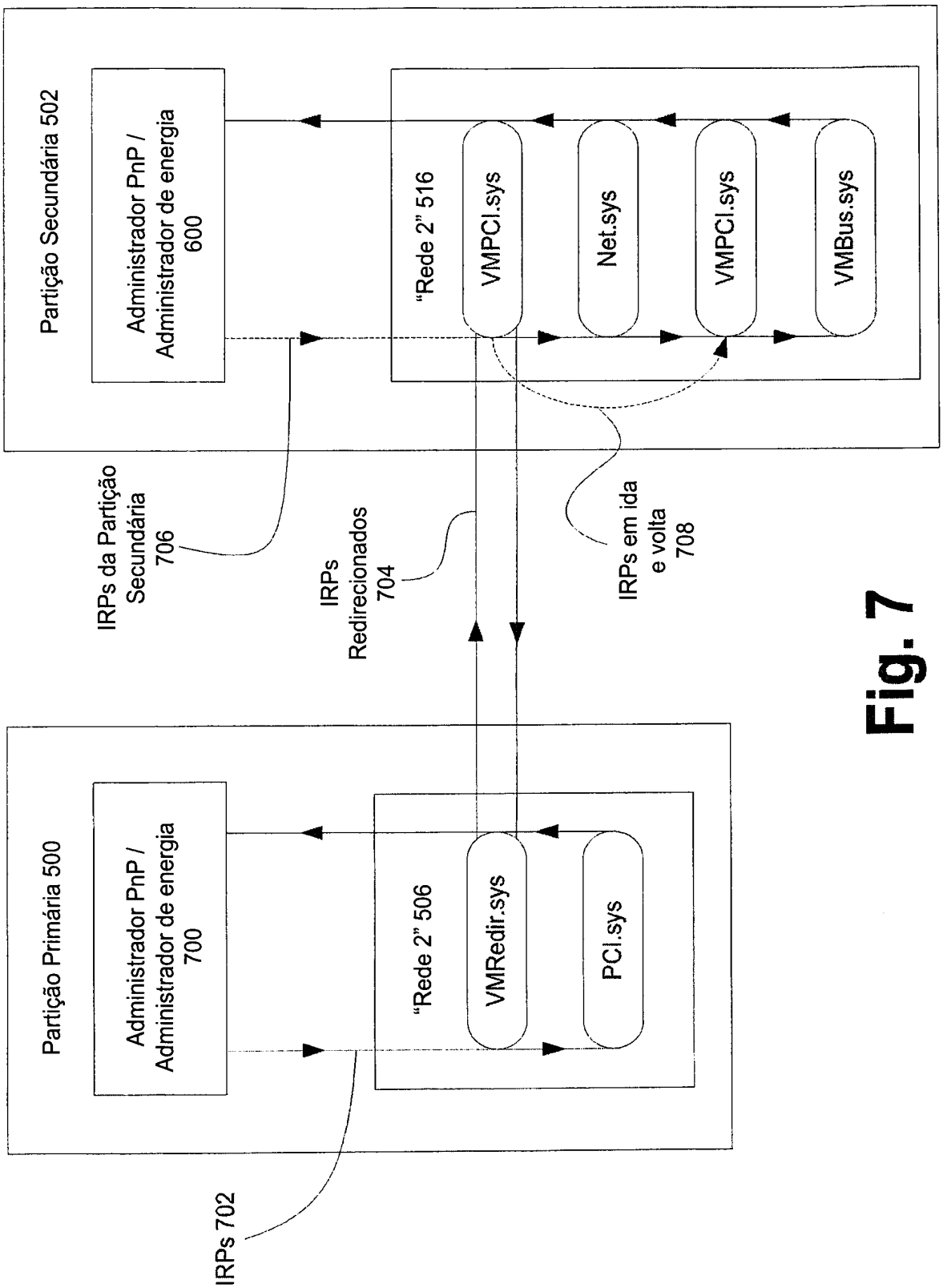

Partição Secundária 502
Administrador PnP / Administrador de energia 600
"Rede 2" 516
VMPCI.sys
Net.sys
VMPCI.sys
VMBus.sys
IRPs da Partição Secundária 706
IRPs Redirecionados 704
IRPs em ida e volta 708
Partição Primária 500
Administrador PnP / Administrador de energia 700
"Rede 2" 506
VMRedir.sys
PCI.sys
IRPs 702

Fig. 7

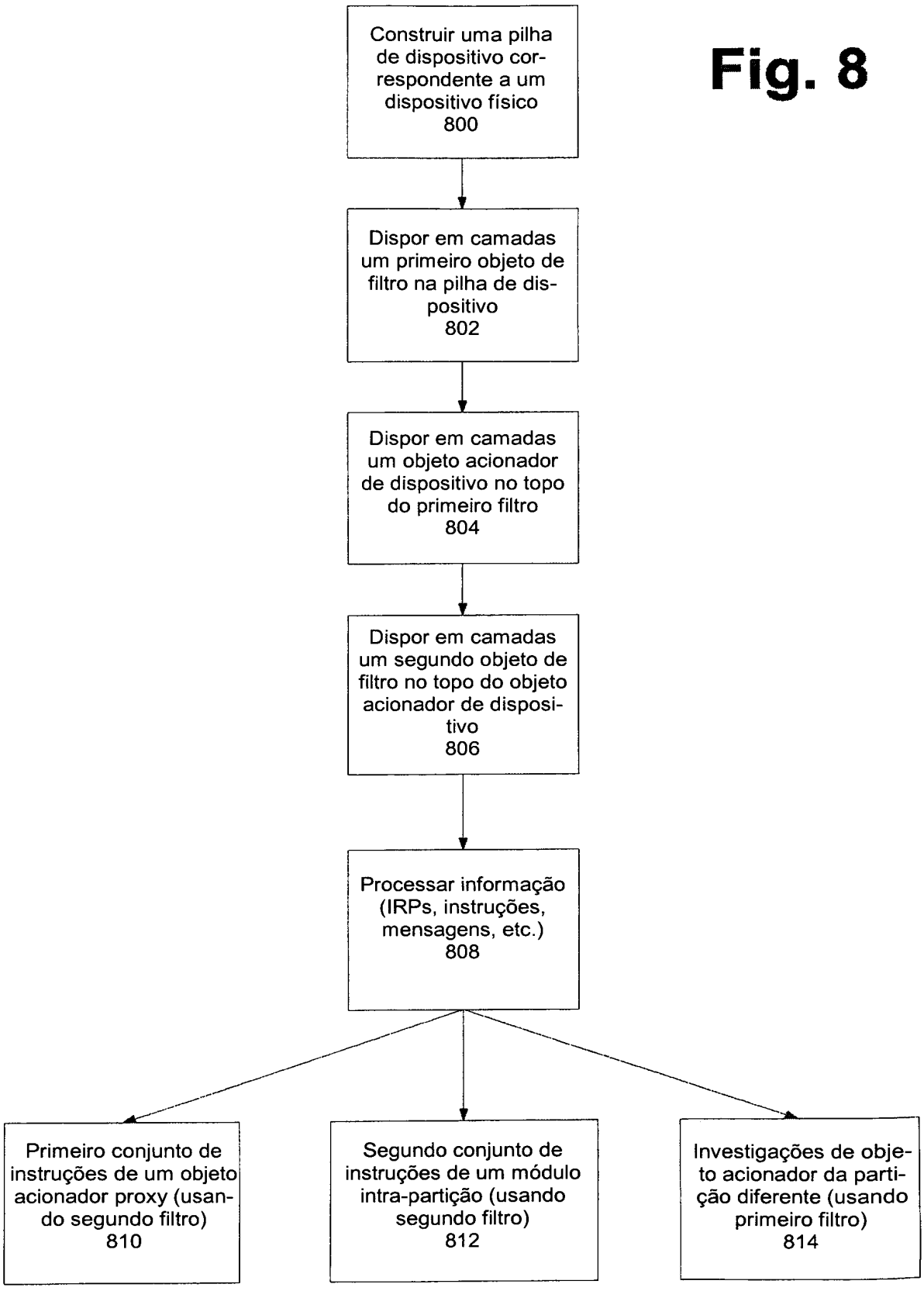

Construir uma pilha de dispositivo correspondente a um dispositivo físico
800
Dispor em camadas um primeiro objeto de filtro na pilha de dispositivo
802
Dispor em camadas um objeto acionador de dispositivo no topo do primeiro filtro
804
Dispor em camadas um segundo objeto de filtro no topo do objeto acionador de dispositivo
806
Processar informação (IRPs, instruções, mensagens, etc.)
808
Primeiro conjunto de instruções de um objeto acionador proxy (usando segundo filtro)
810
Segundo conjunto de instruções de um módulo intra-partição (usando segundo filtro)
812
Investigações de objeto acionador da partição diferente (usando primeiro filtro)
814

Fig. 8

RESUMO

"GERENCIAMENTO DE ESTADO DE HARDWARE DISTRIBUÍDO EM MÁQUINAS VIRTUAIS"

São aqui divulgados mecanismos que gerenciam operações em ambientes de máquina virtual. Uma primeira partição pode ter um objeto acionador proxy correspondente a um objeto acionador em uma segunda partição. O objeto acionador pode controlar um dispositivo físico, mas, em virtude do objeto acionador proxy, a primeira partição pode reter alguma medida de controle sobre o dispositivo físico. O objeto acionador pode ser circundado por um primeiro objeto de filtro abaixo dele, e por um segundo objeto de filtro acima dele. O primeiro objeto de filtro pode fornecer interfaces ao objeto acionador de forma que o objeto acionador possa realizar várias funcionalidades relacionadas a barramento, e o segundo objeto pode receber instruções redirecionadas da primeira partição e fornecê-las ao objeto acionador e interceptar todas as instruções originadas da segunda partição, de maneira tal que se estas instruções estiverem inconsistentes com políticas ajustadas na primeira partição, elas possam ser manipuladas.