



República Federativa do Brasil
Ministério da Economia
Instituto Nacional da Propriedade Industrial

(21) BR 112021008419-3 A2



(22) Data do Depósito: 14/11/2019

(43) Data da Publicação Nacional: 14/09/2021

(54) **Título:** ACESSO MAIS RÁPIDO DA MEMÓRIA DA MÁQUINA VIRTUAL APOIADO POR UMA MEMÓRIA VIRTUAL DO DISPOSITIVO DE COMPUTAÇÃO HOST

(51) **Int. Cl.:** G06F 12/1036; G06F 12/1009; G06F 12/109; G06F 12/08; G06F 9/50; (...).

(30) **Prioridade Unionista:** 21/11/2018 US 16/198,620; 27/05/2019 US 16/423,137.

(71) **Depositante(es):** MICROSOFT TECHNOLOGY LICENSING, LLC.

(72) **Inventor(es):** YEVGENIY BAK; MEHMET IYIGUN; ARUN U. KISHAN.

(86) **Pedido PCT:** PCT US2019061345 de 14/11/2019

(87) **Publicação PCT:** WO 2020/106533 de 28/05/2020

(85) **Data da Fase Nacional:** 30/04/2021

(57) **Resumo:** ACESSO MAIS RÁPIDO DA MEMÓRIA DA MÁQUINA VIRTUAL APOIADO POR UMA MEMÓRIA VIRTUAL DO DISPOSITIVO DE COMPUTAÇÃO HOST. Para aumentar a velocidade com que os níveis hierárquicos de uma Tabela de Endereços de Segunda Camada (SLAT) são percorridos como parte de um acesso à memória onde a memória física do convidado de um ambiente de máquina virtual é apoiada por memória virtual atribuída a um ou mais processos em execução em um dispositivo de computação host, um ou mais níveis hierárquicos de tabelas dentro da SLAT podem ser ignorados ou de outra forma não referenciados. Embora a SLAT possa ser preenchida com correlações de memória em níveis hierarquicamente superiores de tabelas, a tabela de páginas do dispositivo de computação host, suportando o fornecimento de memória virtual do dispositivo de computação host, pode manter um conjunto contíguo correspondente de correlações de memória no nível de tabela hierarquicamente mais baixo, permitindo assim que o dispositivo de computação host pague, ou manipule de outra forma, pedaços menores de memória. Se tal manipulação ocorrer, a SLAT pode ser preenchida novamente com correlações de memória no nível de tabela hierarquicamente mais baixo.

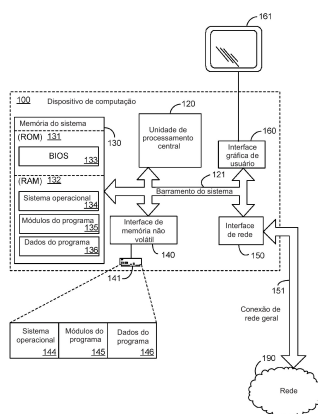


FIG. 1

Relatório Descritivo da Patente de Invenção para **"ACESSO MAIS RÁPIDO DA MEMÓRIA DA MÁQUINA VIRTUAL APOIADO POR UMA MEMÓRIA VIRTUAL DO DISPOSITIVO DE COMPUTAÇÃO HOST"**.

FUNDAMENTOS

[001] Dispositivos de computação de modem operam executando instruções executáveis por computador a partir de memória volátil de alta velocidade na forma de memória de acesso aleatório (RAM), e a execução de tais instruções executáveis por computador muitas vezes envolve a leitura de dados da RAM. Devido ao custo, limitações de tamanho físico, requisitos de energia e outras restrições semelhantes, os dispositivos de computação normalmente compreendem menos RAM do que o exigido pelos processos que são normalmente executados em tais dispositivos de computação. Para acomodar tais restrições, a memória virtual é utilizada, por meio da qual a memória que parece estar disponível para processos em execução em um dispositivo de computação é maior do que o circuito de memória física fornece. A relação entre a memória virtual e a memória física é normalmente gerenciada por um ou mais gerenciadores de memória, que implementam, mantêm e/ou fazem referência a uma "tabela de páginas" cujas informações delineiam a relação entre um ou mais endereços de memória virtual e as localizações dos dados correspondentes, seja na memória física ou em alguma forma de mídia de armazenamento. Para acomodar as quantidades de memória relevantes para os dispositivos de computação modernos e os processos executados no mesmo, uma tabela de página moderna é normalmente composta de vários níveis hierárquicos de tabelas, com uma tabela de nível hierarquicamente superior tendo entradas que identificam cada uma tabela de nível hierarquicamente inferior diferente, e com a tabela de nível hierarquicamente mais baixo compreendendo entradas que não identificam

uma tabela adicional, mas sim identificam os próprios endereços de memória.

[002] Entre os processos que podem ser executados por dispositivos de computação estão os processos que virtualizam, ou abstraem, o hardware subjacente do dispositivo de computação. Tais processos incluem máquinas virtuais, que podem simular um dispositivo de computação subjacente completo para processos executados dentro do contexto de computação virtualizado fornecido por tais máquinas virtuais. Um hipervisor, ou conjunto semelhante de instruções executáveis por computador, pode facilitar o fornecimento de uma máquina virtual virtualizando ou abstraindo o hardware subjacente de um dispositivo físico de computação que hospeda tal hipervisor. Um hipervisor pode manter uma Tabela de endereços de segunda camada (SLAT), que também pode ser hierarquicamente organizada de maneira análoga às tabelas de páginas mencionadas anteriormente. A SLAT pode manter informações que delineiam a relação entre um ou mais endereços de memória que pareçam ser locais de memória física para processos em execução no topo do hipervisor, incluindo processos em execução dentro do contexto de uma máquina virtual cuja virtualização de computação subjacente o hardware é facilitado pelo hipervisor e pelos locais de memória da própria memória física atual.

[003] Quando um processo em execução no contexto de uma máquina virtual, por exemplo, acessa a memória, duas pesquisas diferentes podem ser realizadas. Uma pesquisa pode ser realizada dentro do próprio contexto da máquina virtual para correlacionar o endereço de memória virtual solicitado com um endereço de memória física. Como essa pesquisa é realizada dentro do próprio contexto da máquina virtual, o endereço de memória física identificado é apenas um endereço de memória física conforme percebido por processos em execução no contexto da máquina virtual. Tal pesquisa pode ser realizada

por um gerenciador de memória executando dentro do contexto da máquina virtual e pode ser feita com referência a uma tabela de página que existe dentro do contexto da máquina virtual. Uma segunda pesquisa pode então ser realizada fora do contexto da máquina virtual. Mais especificamente, o endereço de memória física (dentro do contexto da máquina virtual) identificado pela primeira pesquisa, pode ser correlacionado a um endereço de memória física real. Tal segunda pesquisa pode envolver uma ou mais unidades de processamento do dispositivo de computação que faz referência à SLAT, que pode correlacionar endereços de memória física percebidos com endereços de memória física atuais.

[004] Conforme indicado, ambas as tabelas de página e a SLAT podem ser arranjos hierárquicos de diferentes níveis hierárquicos de tabelas. Assim, o desempenho de uma pesquisa de tabela, seja realizada pelo gerenciador de memória com referência à tabela de página, ou realizada pelo hipervisor com referência à SLAT, pode envolver a determinação de uma entrada de tabela apropriada dentro de um nível de tabela hierárquico mais alto, referenciar uma tabela de nível hierarquicamente inferior que é identificada por essa entrada de tabela, determinar uma entrada de tabela apropriada dentro dessa tabela de nível hierarquicamente inferior, fazer referência a uma tabela de nível ainda hierarquicamente inferior que é identificada por essa entrada de tabela, e assim por diante, até que uma tabela de nível hierárquico mais baixo seja alcançada, após o que as entradas individuais dessa tabela de nível hierárquico mais baixo identificam um ou mais endereços específicos, ou faixas de endereços, da própria memória, como supostamente identificam uma tabela adicional. Cada referência a uma tabela de nível hierarquicamente inferior consome ciclos do processador e aumenta a duração de um acesso à memória.

[005] No caso de um acesso à memória a partir de um processo

de execução dentro do contexto de uma máquina virtual, por exemplo, a duração de tal acesso à memória pode incluir tanto o atravessamento dos níveis hierárquicos da tabela de páginas, realizada pelo gerenciador de memória, no contexto da máquina virtual, e, adicionalmente, o atravessamento dos níveis hierárquicos da SLAT realizado pelo hipervisor. O atraso adicional introduzido pela pesquisa realizada pelo hipervisor, com referência à SLAT, renderiza o acesso à memória de processos em execução dentro do contexto de uma máquina virtual, ou, na verdade, qualquer processo que acesse a memória por meio de um hipervisor, mais ineficiente em comparação com os processos acessando a memória mais diretamente. Essas ineficiências podem desencorajar os usuários de alcançar os benefícios de segurança e outros benefícios conferidos pelo acesso à memória por meio de um hipervisor.

SUMÁRIO

[006] Para aumentar a eficiência da utilização da memória por processos de máquina virtual em execução em um dispositivo de computação host, esses processos de máquina virtual podem ser apoiados pela memória virtual do host, permitindo assim que as eficiências de memória virtual tradicional sejam aplicadas à memória consumidas por tais processos de máquina virtual. Para aumentar a velocidade com que os níveis hierárquicos de uma Tabela de Endereço de Segunda Camada (SLAT) são percorridos como parte de um acesso à memória em um ambiente onde a memória física do convidado do ambiente de máquina virtual é apoiada por memória virtual atribuída a um ou mais processos em execução em um dispositivo de computação host, um ou mais níveis hierárquicos de tabelas dentro da SLAT podem ser ignorados ou não referenciados, o que resulta em atravessamento de SLAT mais eficiente e acesso à memória mais eficiente. Embora a SLAT possa ser preenchida com correlações de memória

em níveis hierarquicamente superiores de tabelas, a tabela de páginas do dispositivo de computação host, que suporta o fornecimento de memória virtual do dispositivo de computação host, pode manter um conjunto contíguo correspondente de correlações de memória no nível de tabela hierarquicamente mais baixo, permitindo assim que o dispositivo de computação host retire a página, ou de outra forma manipule, pedaços menores de memória. Se tal manipulação ocorrer, a SLAT pode ser preenchida novamente com correlações de memória no nível de tabela hierarquicamente mais baixo. Por outro lado, se o host pode remontar um conjunto contíguo suficientemente grande de páginas pequenas, a SLAT pode novamente ser preenchida com uma correlação em níveis extremamente altos de tabelas, resultando novamente em um atravessamento de SLAT mais eficiente e acesso à memória mais eficiente. Dessa forma, um atravessamento SLAT mais eficiente pode ser alcançado, mantendo os benefícios de eficiência de utilização de memória de apoiar os processos de máquina virtual com a memória virtual de um dispositivo de computação host.

[007] Este sumário é fornecido para introduzir uma seleção de conceitos de uma forma simplificada que são descritos abaixo na Descrição Detalhada. Este sumário não se destina a identificar características-chave ou características essenciais do assunto reivindicado, nem se destina a ser usado para limitar o escopo do assunto reivindicado.

[008] Características e vantagens adicionais serão tornadas aparentes a partir da seguinte descrição detalhada que prossegue com referência aos desenhos acompanhantes.

DESCRIÇÃO DOS DESENHOS

[009] A seguinte descrição detalhada pode ser melhor compreendida quando considerada em conjunto com os desenhos anexos, dos quais:

[0010] A Figura 1 é um diagrama de blocos de um dispositivo de

computação exemplar

[0011] A Figura 2 é um diagrama de sistema de um sistema de computação de hardware e software exemplar;

[0012] A Figura 3 é um diagrama de blocos que ilustra a terminologia utilizada neste documento;

[0013] A Figura 4 é um diagrama de fluxo que ilustra um método exemplar para apoiar a memória física do convidado com a memória virtual do host;

[0014] A Figura 5 é um diagrama de fluxo que ilustra várias ações exemplares no ciclo de vida dos dados na memória física do convidado que é apoiada pela memória virtual do host;

[0015] A Figura 6 é um diagrama de blocos que ilustra um mecanismo exemplar para apoiar a memória física do convidado com a memória virtual do host;

[0016] A Figura 7 é um diagrama de blocos que ilustra um mecanismo exemplar para traduzir de forma mais eficiente entre a memória física do convidado e a memória física do host, enquanto apoia a memória física do convidado com a memória virtual do host; e

[0017] A Figura 8 é um diagrama de fluxo que ilustra um método exemplar para traduzir de forma mais eficiente entre a memória física do convidado e a memória física do host, ao mesmo tempo que apoia a memória física do convidado com a memória virtual do host.

DESCRIÇÃO DETALHADA

[0018] A descrição a seguir se refere ao aumento da eficiência do acesso à memória, ao mesmo tempo em que fornece uma utilização de memória mais eficiente em dispositivos de computação que hospedam um ou mais processos de máquina virtual que fornecem ambientes de computação de máquina virtual. Para aumentar a eficiência da utilização de memória por processos de máquina virtual em execução em um dispositivo de computação host, esses processos de máquina

virtual podem ser apoiados pela memória virtual do host, permitindo assim que as eficiências de memória virtual tradicional sejam aplicadas à memória consumida por tais processos de máquina virtual. Para aumentar a velocidade com que os níveis hierárquicos de uma Tabela de Endereços de Segunda Camada (SLAT) são percorridos como parte de um acesso à memória em um ambiente onde a memória física do convidado do ambiente da máquina virtual é apoiada pela memória virtual atribuída para um ou mais processos em execução em um dispositivo de computação host, um ou mais níveis hierárquicos de tabelas dentro da SLAT podem ser ignorados ou não referenciados, o que resulta em atravessamento SLAT mais eficiente e acesso à memória mais eficiente. Embora a SLAT possa ser preenchida com correlações de memória em níveis hierarquicamente mais elevados de tabelas, a tabela de páginas do dispositivo de computação host, que suporta o fornecimento de memória virtual do dispositivo de computação host, pode manter um conjunto contíguo correspondente de correlações de memória no nível de tabela hierarquicamente mais baixo, permitindo assim que o dispositivo de computação host retire da página, ou de outra forma manipule, pedaços menores de memória. Se tal manipulação ocorrer, a SLAT pode ser preenchida novamente com correlações de memória no nível de tabela hierarquicamente mais baixo. Por outro lado, se o host pode remontar um conjunto contíguo suficientemente grande de páginas pequenas, a SLAT pode novamente ser preenchida com uma correlação em níveis extremamente altos de tabelas, resultando novamente em um atravessamento SLAT mais eficiente e acesso à memória mais eficiente. Dessa forma, um atravessamento SLAT mais eficiente pode ser alcançado, mantendo os benefícios de eficiência de utilização de memória de apoio de processos de máquina virtual com a memória virtual de um dispositivo de computação host.

[0019] Embora não seja necessária, a descrição abaixo estará no

contexto geral de instruções executáveis por computador, como módulos de programa, sendo executados por um dispositivo de computação. Mais especificamente, a descrição fará referência a atos e representações simbólicas de operações que são realizadas por um ou mais dispositivos de computação ou periféricos, a menos que indicado de outra forma. Como tal, será entendido que tais atos e operações, que às vezes são referidos como sendo executados por computador, incluem a manipulação por uma unidade de processamento de sinais elétricos que representam dados em uma forma estruturada. Esta manipulação transforma os dados ou os mantém em locais na memória, o que reconfigura ou de outra forma altera a operação do dispositivo de computação ou periféricos de uma maneira bem compreendida por aqueles versados na técnica. As estruturas de dados onde os dados são mantidos são locais físicos que possuem propriedades particulares definidas pelo formato dos dados.

[0020] Geralmente, os módulos do programa incluem rotinas, programas, objetos, componentes, estruturas de dados e semelhantes que executam tarefas específicas ou implementam tipos de dados abstratos específicos. Além disso, aqueles versados na técnica apreciarão que os dispositivos de computação não precisam ser limitados a computadores pessoais convencionais e incluem outras configurações de computação, incluindo servidores, dispositivos portáteis, sistemas multiprocessadores, dispositivos eletrônicos de consumo programáveis ou baseados em microprocessador, PCs de rede, minicomputadores, computadores mainframe e semelhantes. Da mesma forma, os dispositivos de computação não precisam ser limitados a dispositivos de computação autônomos, uma vez que os mecanismos também podem ser praticados em ambientes de computação distribuídos, onde as tarefas são realizadas por dispositivos de processamento remoto que estão ligados por meio de uma rede de comunicações. Em um ambi-

ente de computação distribuído, os módulos do programa podem estar localizados em ambos os dispositivos de armazenamento de memória local e remoto.

[0021] Antes de prosseguir com uma descrição detalhada da alocação de memória e mecanismos de acesso referenciados acima, uma descrição detalhada de um dispositivo de computação host exemplar, que fornece contexto para as descrições abaixo, é fornecida com referência ao dispositivo de computação 100 exemplar mostrado na Figura 1. O dispositivo de computação 100 exemplar pode incluir, mas não está limitado a, uma ou mais unidades de processamento central (CPUs) 120, uma memória de sistema 130 e um barramento de sistema 121 que acopla vários sistemas componentes, incluindo a memória do sistema para a unidade de processamento 120. O barramento do sistema 121 pode ser qualquer um dos vários tipos de estruturas de barramento, incluindo um barramento de memória ou controlador de memória, um barramento periférico e um barramento local usando qualquer uma de uma variedade de arquiteturas de barramento. O dispositivo de computação 100 pode incluir opcionalmente hardware gráfico, incluindo, mas não se limitando a, uma interface de hardware gráfico 160 e um dispositivo de exibição 161, que pode incluir dispositivos de exibição capazes de receber entrada de usuário baseada em toque, tal como um dispositivo de exibição sensível ao toque ou com capacidade multitoque. Dependendo da implementação física específica, uma ou mais das CPUs 120, a memória do sistema 130 e outros componentes do dispositivo de computação 100 podem ser colocados fisicamente, como em um único chip. Em tal caso, parte ou todo o barramento do sistema 121 pode ser nada mais do que caminhos de silício dentro de uma única estrutura de chip e sua ilustração na Figura 1 pode ser nada mais do que conveniência notacional para fins de ilustração.

[0022] O dispositivo de computação 100 também inclui tipicamente mídia legível por computador, que pode incluir qualquer mídia disponível que pode ser acessada pelo dispositivo de computação 100 e inclui mídia volátil e não volátil e mídia removível e não removível. A título de exemplo, e não de limitação, a mídia legível por computador pode compreender mídia de armazenamento de computador e mídia de comunicação. A mídia de armazenamento de computador inclui mídia implementada em qualquer método ou tecnologia para armazenamento de conteúdo, tal como instruções legíveis por computador, estruturas de dados, módulos de programa ou outros dados. A mídia de armazenamento de computador inclui, mas não está limitada a, RAM, ROM, EEPROM, memória flash ou outra tecnologia de memória, CD-ROM, discos versáteis digitais (DVD) ou outro armazenamento de disco óptico, cassetes magnéticos, fita magnética, armazenamento de disco magnético ou outros dispositivos de armazenamento magnético, ou qualquer outro meio que pode ser usado para armazenar o conteúdo desejado e que pode ser acessado pelo dispositivo de computação 100. A mídia de armazenamento de computador, entretanto, não inclui a mídia de comunicação. Os meios de comunicação tipicamente incorporam instruções legíveis por computador, estruturas de dados, módulos de programa ou outros dados em um sinal de dados modulado, tal como uma onda portadora ou outro mecanismo de transporte e inclui qualquer meio de entrega de conteúdo. A título de exemplo, e não como limitação, a mídia de comunicação inclui mídia com fio, tal como uma rede com fio ou conexão direta com fio, e mídia sem fio, como tal acústica, RF, infravermelho e outras mídias sem fio. As combinações de qualquer um dos itens acima também devem ser incluídas no escopo da mídia legível por computador.

[0023] A memória do sistema 130 inclui mídia de armazenamento de computador na forma de memória volátil e/ou não volátil, tal como

memória somente para leitura (ROM) 131 e memória de acesso aleatório (RAM) 132. Um sistema básico de entrada / saída 133 (BIOS), contendo as rotinas básicas que ajudam a transferir conteúdo entre elementos dentro do dispositivo de computação 100, tal como durante a inicialização, é normalmente armazenado na ROM 131. A RAM 132 normalmente contém dados e/ou módulos de programa que são imediatamente acessíveis para e/ou atualmente sendo operado pela unidade de processamento 120. A título de exemplo, e não de limitação, a Figura 1 ilustra o sistema operacional 134, outros módulos de programa 135 e dados de programa 136.

[0024] O dispositivo de computação 100 também pode incluir outros meios de armazenamento de computador removíveis / não removíveis, voláteis / não voláteis. Apenas a título de exemplo, a Figura 1 ilustra uma unidade de disco rígido 141 que lê ou grava em mídia magnética não removível e não volátil. Outros meios de armazenamento de computador removíveis / não removíveis, voláteis / não voláteis que podem ser usados com o dispositivo de computação exemplar incluem, mas não estão limitados a, cassetes de fita magnética, cartões de memória flash, discos versáteis digitais, fita de vídeo digital, RAM de estado sólido, ROM de estado sólido e outras mídias de armazenamento de computador, conforme definido e delineado acima. A unidade de disco rígido 141 é normalmente conectada ao barramento de sistema 121 por meio de uma interface de memória não volátil, como a interface 140.

[0025] Os drives e sua mídia de armazenamento de computador associados discutidos acima e ilustrados na Figura 1, fornecem armazenamento de instruções legíveis por computador, estruturas de dados, módulos de programa e outros dados para o dispositivo de computação 100. Na Figura 1, por exemplo, a unidade de disco rígido 141 é ilustrada como armazenando o sistema operacional 144, outros mó-

dulos de programa 145 e dados de programa 146. Observe que esses componentes podem ser iguais ou diferentes do sistema operacional 134, outros módulos de programa 135 e dados de programa 136. O sistema operacional 144, outros módulos de programa 145 e dados de programa 146 recebem diferentes números aqui ilustrando que, no mínimo, eles são cópias diferentes.

[0026] O dispositivo de computação 100 pode operar em um ambiente de rede usando conexões lógicas para um ou mais computadores remotos. O dispositivo de computação 100 é ilustrado como sendo conectado à conexão de rede geral 151 (a uma rede 190) através de uma interface de rede ou adaptador 150, que é, por sua vez, conectado ao barramento de sistema 121. Em um ambiente de rede, módulos de programa representados em relação ao dispositivo de computação 100, ou porções ou periféricos do mesmo, podem ser armazenados na memória de um ou mais outros dispositivos de computação que são acoplados de forma comunicativa ao dispositivo de computação 100 por meio da conexão de rede geral 161. Será apreciado que as conexões de rede mostradas são o exemplo e outros meios de estabelecer um link de comunicação entre dispositivos de computação podem ser usados.

[0027] Embora descrito como um único dispositivo físico, o dispositivo de computação 100 exemplar pode ser um dispositivo de computação virtual, caso em que a funcionalidade dos componentes físicos descritos acima, como a CPU 120, a memória do sistema 130, a interface de rede 160 e outros componentes semelhantes podem ser fornecidos por instruções executáveis por computador. Essas instruções executáveis por computador podem ser executadas em um único dispositivo físico de computação ou podem ser distribuídas por vários dispositivos físicos de computação, incluindo a distribuição por vários dispositivos físicos de computação de maneira dinâmica, de modo que

os dispositivos físicos de computação específicos que hospedam tais instruções executáveis por computadores podem mudar dinamicamente com o tempo, dependendo da necessidade e da disponibilidade. Na situação em que o dispositivo de computação 100 exemplar é um dispositivo virtualizado, os dispositivos físicos de computação subjacentes que hospedam tal dispositivo de computação virtualizado podem, eles próprios, compreender componentes físicos análogos aos descritos acima e operando de maneira semelhante. Além disso, os dispositivos de computação virtual podem ser utilizados em camadas múltiplas com um dispositivo de computação virtual em execução dentro da construção de outro dispositivo de computação virtual. O termo "dispositivo de computação", portanto, conforme utilizado neste documento, significa um dispositivo físico de computação ou um ambiente de computação virtualizado, incluindo um dispositivo de computação virtual, dentro do qual as instruções executáveis por computador podem ser executadas de uma maneira consistente com sua execução por um dispositivo de computação físico. Da mesma forma, os termos que se referem aos componentes físicos do dispositivo de computação, como aqui utilizados, significam aqueles componentes físicos ou suas virtualizações executando as mesmas funções ou funções equivalentes.

[0028] Voltando à Figura 2, o sistema 200 mostrado na mesma ilustra um sistema de computação que compreende hardware de dispositivo de computação e processos sendo executados no mesmo. Mais especificamente, o sistema exemplar 200 mostrado na Figura 2 inclui o próprio hardware do dispositivo de computação, tal como as unidades de processamento acima mencionadas 120 e RAM 132, ilustradas como parte do dispositivo de computação 100 descrito acima. Além disso, o sistema exemplar 200 inclui um hipervisor, tal como o hipervisor 210 exemplar, que pode ser composto por instruções executáveis de computador que são executadas pelo dispositivo de compu-

tação 100 exemplar para permitir que o dispositivo de computação 100 realize a funcionalidade associada ao hipervisor 210 exemplar. Como será descrito em mais detalhes abaixo, o hipervisor 210 exemplar pode incluir, ou pode fazer referência, uma Tabela de endereços de segunda camada (SLAT), tal como a SLAT 220 exemplar.

[0029] O hipervisor 210 exemplar pode virtualizar um conjunto de hardware de computação que pode ser compatível com o hardware do dispositivo de computação 100 ou que pode diferir do mesmo, incluindo diferenças no tipo e/ou capacidade do processador, diferenças em quantidade e/ou tipo de RAM, diferenças em quantidade e/ou tipo de mídia de armazenamento e outras diferenças semelhantes. Tal virtualização pode, entre outras coisas, habilitar um ou mais processos de máquina virtual, como os processos de máquina virtual exemplares 260 e 250. O hipervisor 210 pode apresentar, tanto para processos em execução dentro do contexto dessas máquinas virtuais, e a qualquer outro processo em execução no topo do hipervisor, a aparência de execução direta no hardware do dispositivo de computação 110. O sistema exemplar 200 mostrado na Figura 2 ilustra um sistema operacional, tal como o sistema operacional exemplar 240 executando no topo do hipervisor 210 exemplar. Por sua vez, conforme ilustrado pelo sistema 200 exemplar, o sistema operacional 240 exemplar pode ter execução no mesmo um ou mais programas de aplicativos executáveis por computador, tais como os processos de máquina virtual 260 e 250 mencionados acima.

[0030] O sistema operacional 240 exemplar pode compreender vários componentes, subcomponentes ou aspectos dos mesmos que são relevantes para as descrições abaixo. Embora ilustrado como elementos visualmente distintos na Figura 2, tal representação visual não se destina a transmitir uma independência computacional específica, como limites de processo, silos de memória independentes ou

outros delineamentos computacionais. Um aspecto do sistema operacional 240 pode ser um gerenciador de memória, como o gerenciador de memória exemplar 241. Mais especificamente, o gerenciador de memória exemplar 241 pode ler código ou dados de uma ou mais mídias de armazenamento legíveis por computador e copiar tal código ou dados na memória, que pode ser suportada pela RAM 132.

[0031] Conforme utilizado neste documento, o termo "memória virtual" não é feito com referência a uma máquina virtual, mas sim ao conceito de apresentação para programas de aplicativo ou outros processos em execução em um dispositivo de computação, a aparência de ter acesso a uma maior quantidade de memória do que está fisicamente presente na RAM 132. Assim, por exemplo, a funcionalidade de memória virtual fornecida pelo sistema operacional 240 pode permitir o carregamento de código e/ou dados na memória que é, em última instância, suportado pela memória física, como a memória física 132, mas cuja quantidade é maior que a memória física. O gerenciador de memória 241, em execução como parte do sistema operacional 240 pode manter, modificar e/ou utilizar a tabela de páginas 250 para contabilizar a correlação entre um ou mais endereços de memória virtual e um ou mais endereços de memórias físicas. Mais especificamente, a tabela de páginas 250 fornece tradução de endereço entre um esquema de endereçamento de memória e outro, esquema de endereçamento de memória diferente. Para conseguir isso, a tabela de página 250 pode compreender vários níveis hierárquicos de tabelas, descritos em mais detalhes abaixo, que podem correlacionar um ou mais endereços de memória de acordo com um esquema de endereçamento de memória, ou seja, endereços de memória virtual, para um ou mais endereços de memória de acordo com outros, diferentes esquemas de endereçamento de memória, nomeadamente endereços de memória física.

[0032] O sistema exemplar 200 mostrado na Figura 2 ilustra programas de aplicação em execução no topo do sistema operacional 240. Em particular, tais programas de aplicação podem ser programas de aplicação de máquina virtual que podem instanciar processos de máquina virtual, tais como os processos de máquina virtual exemplares 260 e 270. Cada processo de máquina virtual pode criar um ambiente de computação virtual que pode hospedar seu próprio sistema operacional, programas de aplicativos e outras funcionalidades semelhantes que podem ser hospedadas por um ambiente de computação físico, tal como aquele apresentado pelo dispositivo de computação 100. Dentro do sistema exemplar 200 mostrado na Figura 2, o processo de máquina virtual exemplar 260 é ilustrado como criando um ambiente de máquina virtual 261, e o processo de máquina virtual exemplar 270 é ilustrado como criando um ambiente de máquina virtual 271. O ambiente de máquina virtual exemplar 271 é ilustrado como hospedando um sistema operacional, na forma do sistema operacional exemplar 281, que pode ser análogo ao sistema operacional exemplar 240, ou pode ser um tipo diferente de sistema operacional. Para fins de ilustração, o sistema operacional exemplar 281, em execução dentro do ambiente de máquina virtual exemplar 271, é ilustrado como compreendendo um gerenciador de memória, ou seja, o gerenciador de memória exemplar 282, que, por sua vez, faz referência a uma tabela de página, nomeadamente a tabela de página exemplar 283, para fornecer memória virtual dentro do ambiente de máquina virtual 271. Essa memória virtual dentro do ambiente de máquina virtual 271 pode ser utilizada por processos em execução no topo do sistema operacional exemplar 281, tais como os processos exemplares 291 e 292, que são ilustrados na Figura 2 como também em execução dentro do ambiente de máquina virtual exemplar 271.

[0033] De uma maneira semelhante à implementada pelos geren-

ciadores de memória 241 e 282, o hipervisor 210, ao facilitar a apresentação de um conjunto virtual de hardware de computação, incluindo memória, pode traduzir de forma semelhante endereços de memória de um esquema de endereçamento de memória para outro, um esquema de endereçamento de memória diferente. Mais especificamente, o hipervisor 210 pode traduzir endereços de memória que foram apresentados pelo hipervisor 210 como sendo endereços de memória física em endereços de memória física atuais, como, por exemplo, endereços de memória que identificam locais físicos dentro da memória 132. Tal tradução pode ser realizada com referência a outra tabela hierarquicamente organizada, ou seja, a SLAT 220, que também pode fornecer tradução de endereço de memória entre um esquema de endereçamento de memória, neste caso, endereços que são percebidos por processos em execução no topo do hipervisor 210 como sendo endereços de memória física, e outro, esquema de endereçamento de memória diferente, a saber, neste caso, endereços de memória física atuais.

[0034] De acordo com um aspecto, a memória virtual fornecida pelo sistema operacional 240 pode ser utilizada para apoiar a memória física de uma máquina virtual em vez de usar alocações de memória física não paginada no dispositivo de computação host. Isso permite que o gerenciador de memória 241 gerencie a memória física do host associada à memória física do convidado. Em particular, a lógica de gerenciamento de memória que já existe no host, como o gerenciador de memória 241, agora pode ser aprimorada para gerenciar a memória física das máquinas virtuais de convidado. Isso pode permitir o uso de um hipervisor menor, em termos da quantidade de código usado para implementar o hipervisor, como o hipervisor 210 exemplar. Um hipervisor menor pode ser mais seguro, pois há menos código que pode ser explorado ou que pode conter erros. Além disso, isso permite maior

densidade da máquina virtual no host, uma vez que menos memória física do host pode ser utilizada para implementar as máquinas virtuais. Por exemplo, anteriormente, porque um processo de máquina virtual exigia memória não paginada, se o dispositivo de computação 100 exemplar compreendesse 3 GB da RAM 132, então ele poderia hospedar apenas um processo de máquina virtual que criou uma máquina virtual com 2 GB de RAM. Por outro lado, se a memória consumida por um processo de máquina virtual, como o processo de máquina virtual exemplar 270, fosse memória virtual, como a fornecida pelo gerenciador de memória 241 do sistema operacional 240, então um espaço de memória virtual de 4 GB poderia fornecer suporte para dois processos de máquina virtual, cada um criando uma máquina virtual com 2 GB de RAM. De tal maneira, o dispositivo de computação exemplar 100 poderia hospedar dois processos de máquina virtual, em vez de apenas um, aumentando assim a densidade dos processos de máquina virtual em execução no dispositivo de computação 100 exemplar.

[0035] Outros benefícios incluem uma única base de código de gerenciamento de memória gerenciando toda a memória no sistema (máquina de host e virtual). Assim, melhorias, correções e/ou o ajuste em uma base de código beneficia a todos. Além disso, isso pode resultar em uma redução no custo de engenharia devido à necessidade de manter apenas uma base de código. Outro exemplo de benefício pode ser que as máquinas virtuais podem se beneficiar imediatamente de melhorias de densidade, como paginação, compartilhamento de página, envelhecimento e corte do conjunto de trabalho, cluster de falha, etc. Outro exemplo de benefício pode ser que a memória virtual e os limites de consumo de memória física podem ser definidos em máquinas virtuais como qualquer outro processo, permitindo que os administradores controlem o comportamento do sistema. Outro exemplo de benefício pode ser que recursos adicionais podem ser adicionados ao

gerenciador de memória do host para fornecer mais desempenho, densidade e funcionalidade para máquinas virtuais (e outras cargas de trabalho de máquina não virtual provavelmente se beneficiarão deles também).

[0036] Voltando à Figura 3, o sistema 300 ilustra a terminologia utilizada neste documento para se referir aos vários níveis de tabelas de página e esquemas de endereçamento de memória correspondentes que são aplicáveis ao sistema 200 mostrado na Figura 2. Mais especificamente, dentro do ambiente de máquina virtual 271, o gerenciador de memória exemplar 282 pode referenciar a tabela de página 283 para correlacionar endereços de memória virtual, dentro do contexto do ambiente de máquina virtual 271, para endereços de memória física, novamente, dentro do contexto do ambiente de máquina virtual 271. Conforme indicado anteriormente, o termo "memória virtual" não se refere à memória dentro de um ambiente de máquina virtual, mas sim a uma quantidade de memória disponibilizada para processos em execução em um dispositivo de computação que é suportado por, mas é maior do que, a capacidade de memória física. Assim, para distinguir nominativamente entre memória virtual dentro de um ambiente de máquina virtual e memória virtual dentro de um ambiente de computação não virtualizado ou "metal nu", os adjetivos "convidado" e "host" serão utilizados, com o adjetivo "convidado" sendo indicativo de um ambiente de computação de máquina virtual e o adjetivo "host" sendo indicativo de um ambiente de computação de metal nu no qual executa um processo de máquina virtual que fornece o ambiente de máquina virtual.

[0037] Como tal, e conforme ilustrado no sistema 301, os endereços dentro do esquema de endereçamento 311 de "memória virtual de convidado" podem ser correlacionados a endereços dentro do esquema de endereçamento 312 de "memória física de convidado" pela tabela de páginas 283 dentro do ambiente de máquina virtual 271 que foi

mostrado na Figura 2. Da mesma forma, os endereços dentro do esquema de endereçamento 321 de "memória virtual do host" podem ser correlacionados a endereços dentro do esquema de endereçamento 322 de "memória física do host" pela tabela de página 250 que faz parte do sistema operacional exemplar 240 que foi mostrado na Figura 2 como sendo executado em um ambiente de computação não virtual. Embora não ilustrado pelo sistema exemplar 301 na Figura 3, de acordo com um aspecto, e como será detalhado mais abaixo, a memória física do convidado 312 de um ambiente de máquina virtual pode ser apoiada pela memória virtual do host 321, fornecendo assim os benefícios acima mencionados. Tal apoio da memória física do host 312 com a memória virtual do host 321 pode acarretar a modificação da SLAT 220, que provê a tradução entre o esquema de endereçamento 312 de "memória física do convidado" e o esquema de endereçamento 322 de "memória física do host", conforme ilustrado pelo sistema exemplar 301.

[0038] Voltando ao sistema 302, também mostrado na Figura 3, um arranjo hierárquico de tabelas é ilustrado, tal como pode ser utilizado para implementar a tabela de página exemplar 250, a tabela de página exemplar 283 e/ou a SLAT 220 exemplar. Para acomodar quantidades de memória aplicáveis a dispositivos de computação modernos e processos de programas de aplicativos, as tabelas de páginas 250, 283 e/ou a SLAT 220 terão vários, normalmente quatro, níveis de tabelas hierárquicas. No entanto, para facilidade de ilustração, o sistema 302 ilustra apenas dois níveis de tabelas hierárquicas. Uma tabela 340 de nível hierarquicamente superior pode compreender múltiplas entradas de tabela, como as entradas de tabela exemplares 341, 342, 343 e 344. Por exemplo, a tabela 340 de nível hierarquicamente superior pode ter 4 KB de tamanho e pode compreender 512 entradas de tabela discretas, que podem ter oito bytes cada. Cada entrada de

tabela pode compreender um apontador, ou outro identificador semelhante, de uma tabela de nível hierarquicamente inferior, como uma das tabelas exemplares de nível hierarquicamente inferior 350 e 360. Assim, por exemplo, a entrada de tabela exemplar 341 pode compreender uma identificação da tabela 350 exemplar de nível hierarquicamente inferior, conforme ilustrado pela seta 371. Se a tabela 340 de nível hierarquicamente superior tiver 4 KB de tamanho e cada entrada de tabela tiver 8 bytes de tamanho, pelo menos alguns desses oito bytes de dados podem ser um apontador ou outro identificador semelhante da tabela de nível hierarquicamente inferior 350. A entrada de tabela 342 exemplar pode, de maneira semelhante, compreender uma identificação da tabela de nível hierarquicamente inferior 360, conforme ilustrado pela seta 372. Cada uma das entradas de tabela discretas restantes da tabela de nível hierarquicamente superior exemplar 340 pode, de modo semelhante, identificar tabelas únicas de nível hierarquicamente inferior. Assim, no exemplo em que a tabela de nível hierarquicamente superior 340 tinha 4 KB de tamanho e compreendia 512 entradas de tabela, cada um dos oito bytes de tamanho, tal tabela de nível hierarquicamente superior 340 pode compreender 512 identificações de 512 tabelas de nível hierarquicamente inferior únicas, como as tabelas exemplares de nível hierarquicamente inferior 350 e 360.

[0039] Cada uma das tabelas de nível hierarquicamente inferior pode, de forma análoga, compreender entradas de tabela individuais. Por exemplo, a tabela de nível hierarquicamente inferior exemplar 350 pode compreender as entradas de tabela exemplares 251, 352, 353 e 354. Da mesma forma, a tabela exemplar de nível hierarquicamente inferior 360 pode compreender as entradas de tabela 361, 362, 363 e 364. Como um exemplo, cada uma das tabelas de nível hierarquicamente inferior pode ter um tamanho e estrutura equivalentes à tabela

de nível hierarquicamente superior 340. Assim, a tabela exemplar de nível hierarquicamente inferior 350 pode ter 4 KB de tamanho e pode compreender 512 entradas de tabelas, como as entradas de tabela exemplares 351, 352, 353 e 354, que podem ter oito bytes cada. No exemplo ilustrado pelo sistema 302, as entradas de cabo das tabelas de nível hierarquicamente inferior, como, por exemplo, as entradas de tabela exemplares 351, 352, 353 e 354 da tabela de nível hierarquicamente inferior 350 podem, cada uma, identificar um faixa contígua de endereços de memória. Por exemplo, a entrada de tabela 351 exemplar pode identificar a faixa de endereços de memória 371, conforme ilustrado pela seta 391. A faixa de endereços de memória 371 pode compreender uma "página" de memória e pode ser a menor quantidade individualmente gerenciável de memória. Assim, ações como mover informações armazenadas em memória volátil temporariamente para mídia de armazenamento não volátil, ou vice-versa, como parte da habilitação de uma quantidade de memória virtual maior do que a quantidade de memória física instalada, podem ser realizadas em página por página. Além disso, as permissões de acesso podem ser estabelecidas página a página. Por exemplo, porque a faixa de memória 371 é identificada individualmente pela entrada da tabela 351, as permissões de acesso podem se aplicar a todos os endereços de memória dentro da faixa de memória 371. Por outro lado, as permissões de acesso dos endereços de memória dentro do faixa de memória 371 pode ser independente das permissões de acesso estabelecidas para a faixa de memória 372, que pode ser identificada exclusivamente por uma entrada de tabela diferente, ou seja, a entrada de tabela 352, conforme ilustrado pela seta 392.

[0040] A quantidade de memória em uma única página de memória, como a faixa de memória 371, pode ser dependente de vários fatores, incluindo design de processador e outros fatores de hardware,

bem como conexões de comunicação e semelhantes. Em um exemplo, uma página de memória, como aquela representada pela faixa de memória 371, pode ter 4 KB de tamanho.

[0041] Ao receber uma solicitação para acessar locais de memória identificados por um ou mais endereços de memória, uma entrada de tabela em uma tabela de nível hierarquicamente superior pode ser identificada que corresponde a uma faixa de endereços de memória que inclui os endereços de memória que devem ser acessados. Por exemplo, se os endereços de memória a serem acessados incluírem a memória representada pela memória 371, a entrada da tabela 341 pode ser identificada na tabela 340 de nível hierarquicamente superior. A identificação da entrada da tabela 341 pode, por sua vez, resultar na identificação da tabela de nível hierarquicamente inferior 350. Dentro da tabela de nível hierarquicamente inferior 350, os endereços de memória a serem acessados podem ser determinados como parte da memória que é identificada pela entrada da tabela 351. As informações dentro da entrada da tabela 351 podem então identificar a memória 371 procurada ser acessada.

[0042] Cada uma dessas passagens de um nível hierarquicamente inferior da tabela de páginas pode aumentar o atraso entre o recebimento de uma solicitação de acesso à memória e o retorno do acesso a essa memória. De acordo com um aspecto, uma entrada de tabela de uma tabela de nível hierarquicamente superior, tal como, por exemplo, a entrada de tabela exemplar 341 da tabela de nível hierarquicamente superior 340, pode identificar não uma tabela de nível hierarquicamente inferior, como a tabela de nível hierarquicamente inferior exemplar 350, mas sim uma página de memória que pode corresponder a toda a faixa de memória que teria sido identificada pelas entradas da tabela em uma tabela de nível hierarquicamente inferior. Assim, por exemplo, se cada página de memória 371, 372 e 373 identifi-

cada pelas entradas de tabela individuais 351, 352 e 353 da tabela de nível hierarquicamente inferior 350 tinha 4 KB de tamanho, então uma faixa de memória de 2 MB de tamanho pode ser identificada pela quantidade combinada de todas as entradas de tabela individuais da tabela de nível hierarquicamente inferior 350, uma vez que a tabela exemplar de nível hierarquicamente inferior 350 pode compreender 512 entradas de tabela, cada uma identificando uma faixa de memória de 4 KB em tamanho. Em tal exemplo, se todos os 2 MB de memória fossem tratados como uma única página de memória de 2 MB, tal uma única página de memória grande poderia ser identificada diretamente pela entrada de tabela exemplar 341 da tabela de nível hierarquicamente superior 340. Em tal instância, porque a entrada de tabela exemplar 341 identificaria diretamente uma única página de memória de 2 MB, tal como a página de memória de 2 MB exemplar 370, e não identificaria uma tabela de nível hierarquicamente inferior, não haveria necessidade de fazer referência a qualquer tabela de nível inferior. Por exemplo, se uma tabela de página ou SLAT compreendesse quatro níveis hierárquicos, utilizando uma grande página de memória, como a página de memória grande exemplar 370, permitiria que um desses níveis hierárquicos de tabelas fosse ignorado e não referenciado, fornecendo assim tradução de endereços de memória entre a memória virtual e física referenciando apenas três níveis hierárquicos de tabelas, em vez de quatro, fornecendo acesso à memória de forma mais eficiente, quase 25% mais eficiente. Como uma página é a menor quantidade de memória gerenciável individualmente, se uma página de memória de 2 MB for usada, as permissões de acesso à memória, por exemplo, podem ser as mesmas para todos os 2 MB de memória em uma página de memória tão grande.

[0043] Como outro exemplo, se houvesse uma tabela hierarquicamente organizada que estivesse hierarquicamente acima da tabela

exemplar 340, e cada entrada na tabela exemplar 340, tal como as entradas 341, 342, 343 e 344 poderia endereçar uma página de memória de 2 MB, tal como a página de memória de 2 MB exemplar 370, então as 512 individuais tais entradas de tabela poderiam, quando agregados juntos, endereçarem 1 GB de memória. Se uma única página de 1 GB de memória foi usada, tanto a referência à tabela 340, quanto uma referência subsequente a uma tabela de nível hierarquicamente inferior, como a tabela exemplar 350, poderiam ser evitadas. Voltando ao exemplo em que uma tabela de página ou SLAT compreende quatro níveis hierárquicos, a utilização de uma página de memória tão grande permitiria que dois desses níveis hierárquicos de tabelas fossem ignorados, e não referenciados, fornecendo assim a tradução de endereço de memória entre memória virtual e física referenciando apenas os níveis hierárquicos das tabelas, em vez de quatro, fornecendo acesso à memória em aproximadamente metade do tempo. Novamente, como uma página é a menor quantidade de memória gerenciável individualmente, se uma página de memória de 1 GB for usada, as permissões de acesso à memória, por exemplo, podem ser as mesmas para todos os 1 GB de memória em uma página de memória tão grande.

[0044] O termo "página de memória grande", como utilizado neste documento, significa uma faixa contígua de memória que é dimensionada de modo que englobe todas as faixas de memória que teriam sido identificadas por uma tabela única no nível hierarquicamente mais baixo e pode ser identificada de forma única e direta por uma única entrada de tabela de uma tabela em um nível hierárquico acima da tabela de nível hierarquicamente mais baixo. No exemplo de dimensionamento específico fornecido acima, uma "página de memória grande", como esse termo é definido neste documento, seria uma página de memória de 2 MB de tamanho. No entanto, conforme indicado, o termo "página de memória grande" não se refere a um tamanho espe-

cífico, mas sim a quantidade de memória em uma "página de memória grande" depende do design hierárquico da própria tabela de páginas e da quantidade de memória referenciada por cada entrada de tabela no nível hierarquicamente mais baixo da tabela de página.

[0045] De maneira semelhante, o termo "página de memória enorme", conforme utilizado neste documento, significa uma faixa contígua de memória que é dimensionada de modo que abrange todos os faixas de memória que teriam sido identificados por uma única tabela em um segundo nível hierarquicamente mais baixo e pode ser única e diretamente identificada por uma única entrada de tabela em um nível hierárquico acima da tabela de segundo nível hierárquico mais baixo. No exemplo de dimensionamento específico fornecido acima, uma "página de memória enorme", como esse termo é utilizado aqui, seria uma página de memória de 1 GB de tamanho. Novamente, como indicado, o termo "página de memória enorme" não se refere a um tamanho específico, mas sim a quantidade de memória em uma "página de memória enorme" depende do design hierárquico da própria tabela de páginas e da quantidade de memória referenciada por cada entrada da tabela no nível hierarquicamente mais baixo da tabela de página.

[0046] Voltando à Figura 2, as descrições dos mecanismos pelos quais a memória virtual do host (de um dispositivo de computação host) pode ser utilizada para apoiar a memória física do convidado (de uma máquina virtual) são fornecidos com referência a uma pilha de virtualização, tal como a pilha de virtualização exemplar 230 mostrada na Figura 2. Um processo de modo de usuário pode ser executado no dispositivo de computação host para fornecer memória virtual para apoiar máquinas virtuais de convidado. Um desses processos de modo de usuário pode ser criado para cada máquina do convidado. Alternativamente, um único processo de modo de usuário pode ser usado para várias máquinas virtuais, ou múltiplos processos podem ser usa-

dos para uma única máquina virtual. Alternativamente, a memória virtual pode ser implementada em outras formas que não usando um processo de modo de usuário, como será ilustrado abaixo. Para fins de ilustração, a memória virtual atribuída ao processo de máquina virtual 270 pode ser utilizada para apoiar a memória física do host apresentada pelo processo de máquina virtual 270 no ambiente de máquina virtual 271 e, da mesma forma, a memória virtual atribuída ao processo da máquina virtual O 260 pode ser utilizado para apoiar a memória física do convidado apresentada pelo processo da máquina virtual 260 no ambiente da máquina virtual 261. Embora ilustrado como tal no sistema exemplar 200 da Figura 2, o processo do modo de usuário acima mencionado pode ser separado de processos de máquina virtual, tais como processos de máquina virtual exemplares 260 e 270.

[0047] De acordo com um aspecto, uma pilha de virtualização, tal como a pilha de virtualização exemplar 230, pode alocar memória virtual do host no espaço de endereço de um processo de modo de usuário designado que hospedará a máquina virtual, tal como o processo de máquina virtual exemplar 260 ou 270. O gerenciador de memória do host 241 pode tratar esta memória como qualquer outra alocação virtual, o que significa que ela pode ser paginada, a página física a apoiando pode ser alterada para fins de satisfazer alocações de memória contíguas em outro lugar no sistema, as páginas físicas podem ser compartilhadas com outra alocação virtual em outro processo (que por sua vez pode ser outra alocação de apoio da máquina virtual ou qualquer outra alocação no sistema). Ao mesmo tempo, muitas otimizações são possíveis para fazer o gerenciador de memória do host tratar as alocações virtuais de apoio da máquina virtual especialmente conforme necessário. Além disso, se a pilha de virtualização 230 escolher priorizar o desempenho sobre a densidade, ela pode executar muitas operações suportadas pelo gerenciador de memória do sistema

operacional 241, como bloquear as páginas na memória para garantir que a máquina virtual não experimente paginação para essas porções. Da mesma forma, páginas grandes podem ser usadas para fornecer ainda mais desempenho para a máquina virtual, como será detalhado a seguir.

[0048] Uma determinada máquina virtual pode ter todos os seus endereços de memória física do convidado na memória física do convidado apoiado pela memória virtual do host ou pode ter alguns de seus endereços de memória física do convidado na memória física do convidado apoiado pela memória virtual do host e alguns por mecanismos legados, como alocações de memória física não paginável feitas a partir da memória física do host.

[0049] Quando uma nova máquina virtual é criada, a pilha de virtualização 230 pode usar um processo de modo de usuário para hospedar a alocação de memória virtual para apoiar a memória física do convidado. Pode ser um processo vazio recém-criado, um processo existente que hospeda múltiplas máquinas virtuais ou um processo por máquina virtual que também contém outras alocações virtuais relacionadas à máquina virtual que não são visíveis para a própria máquina virtual (por exemplo, estruturas de dados de pilha de virtualização). Também é possível usar o espaço de endereço virtual de kernel para apoiar a máquina virtual. Uma vez que tal processo seja encontrado ou criado, a pilha de virtualização 230 pode fazer uma alocação virtual de memória privada (ou um mapeamento de seção / arquivo) em seu espaço de endereço que corresponde à quantidade de memória física do convidado que a máquina virtual deve ter. Especificamente, a memória virtual pode ser uma alocação privada, um mapeamento de arquivo, um mapeamento de seção apoiado por arquivo de paginação ou qualquer outro tipo de alocação suportada pelo gerenciador de memória do host 241. Isso não precisa ser uma alocação contígua, embora, como

será detalhado abaixo, tal alocação pode fornecer vantagens de velocidade de acesso. Essa alocação pode, em vez disso, ser um número arbitrário de alocações.

[0050] Uma vez que as alocações de memória virtual foram feitas, elas podem ser registradas com os componentes que irão gerenciar o espaço de endereço físico da máquina virtual, como o ambiente de máquina virtual exemplar 271, e mantê-lo em sincronia com as páginas de memória física do host que o gerenciador de memória do host 241 escolherá para apoiar as alocações de memória virtual. Esses componentes podem ser o hipervisor 210 e a pilha de virtualização 230, que podem ser implementados como parte do kernel do host e/ou um driver. O hipervisor 210 pode gerenciar a tradução entre os faixas de endereço de memória física do convidado e os faixas de endereço de memória física do host correspondente, utilizando a SLAT 220. Em particular, a pilha de virtualização 230 pode atualizar a SLAT 220 com as páginas de memória física do host que estão apoiando o páginas da memória física correspondentes do convidado. O hipervisor 210 pode expor a capacidade da pilha de virtualização 230 de receber intercepções quando um certo tipo de acesso é realizado por uma máquina virtual do convidado, como a máquina virtual exemplar 271, para um determinado endereço de memória física do convidado. Por exemplo, a pilha de virtualização 230 pode solicitar o recebimento de uma intercepção quando um determinado endereço físico é escrito pela máquina virtual do convidado 271.

[0051] De acordo com um aspecto, quando uma máquina virtual, como a máquina virtual exemplar 271, é criada pela primeira vez, a SLAT 220 pode não conter quaisquer entradas válidas correspondendo a tal máquina virtual porque nenhum endereço de memória física do host pode ter sido alocado para apoiar os endereços de memória física do convidado de tal máquina virtual (embora, conforme ilustrado

abaixo, em algumas modalidades, a SLAT 220 pode ser pré-preenchida para alguns endereços de memória física do convidado em ou próximo ao mesmo tempo que a criação da máquina virtual 271). O hipervisor 210 pode estar ciente das faixas de endereço de memória física do convidado que serão utilizadas pela máquina virtual 27, mas nenhuma delas precisa ser apoiada por qualquer memória física do host neste ponto. Quando o processo 270 da máquina virtual começa a ser executado, ele pode começar a acessar suas páginas de memória física (convidado). À medida que cada novo endereço de memória física é acessado, ele pode gerar uma interceptação do tipo apropriado (ler / gravar / executar), uma vez que a entrada SLAT correspondente ainda não está preenchida com os endereços de memória física do host correspondentes. O hipervisor 210 pode receber a interceptação de acesso de convidado e pode encaminhá-la para a pilha de virtualização 230. Por sua vez, a pilha de virtualização 230 pode referir-se a uma estrutura de dados mantida pela pilha de virtualização 230 para encontrar a faixa de endereço de memória virtual do host que corresponde à faixa de endereço de memória física do convidado solicitado (e o processo de host cujo espaço de endereço virtual o suporte foi alocado, tal como o processo de máquina virtual exemplar 270). Nesse ponto, a pilha de virtualização 230 pode saber o endereço de memória virtual do host específico que corresponde ao endereço de memória física do convidado que gerou a interceptação.

[0052] A pilha de virtualização 230 pode então emitir uma falha virtual para o gerenciador de memória do host 241 no contexto do processo que hospeda a faixa de endereço virtual, tal como o processo de máquina virtual 270. A falha virtual pode ser emitida com o tipo de acesso correspondente (ler / gravar / executar) da interceptação original que ocorreu quando a máquina virtual 271 acessou seu endereço físico na memória física do convidado. Uma falha virtual pode executar

um caminho de código análogo ou idêntico que uma falha de página normal usaria para tornar o endereço virtual especificado válido e acessível pela CPU do host. Uma diferença pode ser que um caminho de código de falha virtual pode retornar o número da página física que o gerenciador de memória 241 usou para tornar o endereço virtual válido. Este número de página física pode ser o endereço de memória física do host que está apoiando o endereço virtual do host, que, por sua vez, está apoiando o endereço de memória física do convidado que gerou originalmente a interceptação de acesso no hipervisor 210. Neste ponto, a pilha de virtualização 230 pode gerar ou atualizar uma entrada de tabela na SLAT 220 correspondente ao endereço de memória física do convidado original que gerou a interceptação com o endereço de memória física do host e o tipo de acesso (ler / gravar / executar) que foi usado para tornar o endereço virtual válido no host. Uma vez feito isso, o endereço de memória física do convidado pode ser imediatamente acessível com esse tipo de acesso para a máquina virtual do convidado 271. Por exemplo, um processador virtual paralelo na máquina virtual do convidado 271 pode acessar imediatamente tal endereço sem bater uma interceptação. O manuseio de interceptação original pode ser concluído e o processador virtual original que gerou a interceptação pode repetir sua instrução e prosseguir para acessar a memória agora que a SLAT contém uma entrada de tabela que pode correlacionar o endereço físico do convidado solicitado com uma memória física do host correspondente.

[0053] Se e/ou quando o gerenciador de memória do host 241 decidir realizar qualquer ação que poderia ou iria alterar o suporte do endereço físico do host do endereço virtual do host que se tornou válido por meio de uma falha virtual, ele pode realizar uma liberação do buffer de tradução (TLB) para esse endereço virtual do host. Ele já pode executar tal ação para estar em conformidade com um contrato exis-

tente que o gerenciador de memória do host 241 pode ter com CPUs de hardware no host. A pilha de virtualização 230 pode interceptar tais liberações TLB e invalidar as entradas SLAT correspondentes de quaisquer endereços virtuais de host que são liberados que estão apoiando quaisquer endereços de memória física do convidado em quaisquer máquinas virtuais. A chamada de liberação TLB pode identificar uma faixa de endereços virtuais sendo liberados. A pilha de virtualização 230 pode então procurar os endereços virtuais do host que estão sendo liberados contra suas estruturas de dados, que podem ser indexados por endereço virtual do host para encontrar faixas físicas do convidado que podem ser apoiadas pelo endereço virtual do host dado. Se qualquer uma dessas faixas for encontrada, as entradas SLAT correspondentes a esses endereços de memória física do convidado podem ser invalidadas. Além disso, o gerenciador de memória do host pode tratar as alocações virtuais que retornam a memória virtual de forma diferente, se necessário ou desejado, para otimizar o comportamento de descarga de TLB (por exemplo, para reduzir o tempo de invalidação SLAT, interceptações de memória subsequentes, etc.)

[0054] A pilha de virtualização 230 pode sincronizar cuidadosamente a atualização da SLAT 220 com o número de página de memória física do host retornado da falha virtual (que é atendida pelo gerenciador de memória 241) contra liberações TLB realizadas pelo host (que são emitidos pelo gerenciador de memória 241). Isso pode ser feito para evitar a adição de sincronização complexa entre o gerenciador de memória do host 241 e a pilha de virtualização 230. O número da página física retornado pela falha virtual pode estar obsoleto no momento em que é retornado à pilha de virtualização 230. Por exemplo, os endereços virtuais podem já ter sido invalidados. Ao interceptar as chamadas de descarga TLB do gerenciador de memória do host 241, a pilha de virtualização 230 pode saber quando esta corrida ocor-

reu e tentar novamente a falha virtual para adquirir o número de página física atualizado.

[0055] Quando a pilha de virtualização 230 invalida uma entrada SLAT, qualquer acesso subsequente a esse endereço de memória física do convidado pela máquina virtual 271 gerará novamente uma interceptação para o hipervisor 210, que por sua vez será encaminhada para a pilha de virtualização 230 para ser resolvida conforme descrito acima. O mesmo processo pode se repetir quando um endereço de memória física do convidado é acessado para leitura primeiro e, em seguida, é gravado posteriormente. A gravação gerará uma interceptação separada porque a entrada SLAT só pode ter se tornado válida com um tipo de acesso de "leitura". Essa interceptação pode ser enviada para a pilha de virtualização 230 como de costume e uma falha virtual com acesso de "gravação" pode ser emitida para o gerenciador de memória do host 241 para o endereço virtual apropriado. O gerenciador de memória do host 241 pode atualizar seu estado interno (normalmente na entrada da tabela de página (ou "PTE")) para indicar que a página da memória física do host agora está suja. Isso pode ser feito antes de permitir que a máquina virtual grave em seu endereço de memória física do convidado, evitando assim a perda e/ou corrupção de dados. Se e/ou quando o gerenciador de memória do host 241 decidir cortar esse endereço virtual (que realizará uma liberação de TLB e invalidará a entrada SLAT correspondente como um resultado), o gerenciador de memória do host 241 pode saber que a página está suja e precisa ser gravado em um meio de armazenamento legível por computador não volátil, tal como um arquivo de paginação em uma unidade de disco rígido, antes de ser reaproveitado. Dessa forma, a sequência descrita acima não é diferente do que aconteceria para uma alocação virtual privada regular para qualquer outro processo em execução no dispositivo de computação host 100.

[0056] Como uma otimização, o gerenciador de memória do host 241 pode escolher realizar uma passagem de combinação de página sobre toda a sua memória 120. Esta pode ser uma operação onde o gerenciador de memória do host 241 encontra páginas idênticas em todos os processos e as combina em uma cópia somente para leitura da página compartilhada por todos os processos. Se e/ou quando qualquer um dos endereços virtuais combinados são gravados, o gerenciador de memória 241 pode realizar uma operação de cópia na gravação para permitir que a gravação prossiga. Tal otimização pode agora funcionar de forma transparente em máquinas virtuais, como as máquinas virtuais exemplares 261 e 271, para aumentar a densidade de máquinas virtuais sendo executadas em um determinado dispositivo de computação combinando páginas idênticas em máquinas virtuais e, assim, reduzindo consumo de memória. Quando ocorre a combinação de páginas, o gerenciador de memória do host 241 pode atualizar as PTEs que mapeiam os endereços virtuais que estão sendo afetados. Durante esta atualização, ele pode realizar uma liberação de TLB porque o endereço de memória física do host pode mudar para esses endereços virtuais de uma página privada exclusiva para a página compartilhada. Como parte disso, conforme descrito acima, a pilha de virtualização 230 pode invalidar as entradas SLAT correspondentes. Se e/ou quando os endereços de memória física do convidado, cujos endereços virtuais foram combinados para apontar para a página compartilhada, são lidos, a resolução de falha virtual pode retornar o número da página física da página compartilhada durante o tratamento de interceptação e a SLAT 220 pode ser atualizada para apontar para a página compartilhada.

[0057] Se qualquer um dos endereços de memória física do convidado combinados são gravados pela máquina virtual, a falha virtual com acesso de gravação pode realizar uma operação de cópia na gra-

vação e uma nova página de memória física do host privado. O número pode ser retornado e atualizado na SLAT 220. Por exemplo, a pilha de virtualização 230 pode direcionar o gerenciador de memória do host 241 para realizar uma passagem de combinação de páginas. Em algumas modalidades, é possível para a pilha de virtualização 230 especificar qual parte da memória deve ser verificada para combinação ou quais processos devem ser verificados. Por exemplo, a pilha de virtualização 230 pode identificar os processos, tais como os processos de máquina virtual exemplares 260 e 270, cujas alocações de memória virtual do host estão apoiando a memória física do convidado das máquinas virtuais correspondentes, ou seja, as máquinas virtuais exemplares 261 e 271, como sendo os processos que devem ser verificados para combinação.

[0058] Mesmo quando a SLAT 220 é atualizada para permitir acesso de gravação devido a uma falha virtual sendo realizada para gravação, o hipervisor 210 pode suportar o acionamento de intercepções em gravações para tais entradas SLAT se solicitado pela pilha de virtualização 230. Isso pode ser útil porque a pilha de virtualização 230 pode querer saber quando as gravações ocorrem, independentemente do fato de ser aceitável para o gerenciador de memória do host 241 que essas gravações ocorram. Por exemplo, a migração ao vivo de máquinas virtuais ou captura de imagem de máquina virtual pode envolver a pilha de virtualização 230 monitorando as gravações. A pilha de virtualização 230 ainda pode ser notificada quando as gravações ocorrem, mesmo se o estado do gerenciador de memória do host já tiver sido atualizado de acordo com as gravações, como, por exemplo, se a PTE já tiver sido marcada como suja.

[0059] O gerenciador de memória do host 241 pode ser capaz de manter um histórico de acesso preciso para cada página virtual do host apoiando o espaço de endereço de memória física convidado,

assim como faz para páginas virtuais de host regulares alocadas em qualquer outro espaço de endereço de processo. Por exemplo, um "bit acessado" em uma PTE pode ser atualizado durante falhas virtuais realizadas como parte do tratamento de intercepções de memória. Quando o gerenciador de memória do host limpa o bit acessado em qualquer PTE, ele já pode liberar o TLB para evitar corrupção de memória. Conforme descrito antes, essa liberação de TLB pode invalidar a entrada SLAT correspondente, que pode, por sua vez, gerar uma intercepção de acesso se a máquina virtual acessar seu endereço de memória física do convidado novamente. Como parte do tratamento da intercepção, o processamento de falha virtual no gerenciador de memória do host 241 pode definir o bit acessado novamente, mantendo assim o histórico de acesso adequado para a página. Alternativamente, por motivos de desempenho, tal como para evitar intercepções de acesso no hipervisor 210, tanto quanto possível, o gerenciador de memória do host 241 pode consumir informações de acesso à página diretamente do hipervisor 210 conforme medido a partir das entradas SLAT, se suportado pelo hardware subjacente. O gerenciador de memória do host 241 pode cooperar com a pilha de virtualização 230 para traduzir informações de acesso na SLAT 220 (que é organizado por endereços de memória física do convidado) para os endereços de memória virtual do host apoiando esses endereços de memória física do convidado para saber quais endereços foram acessados.

[0060] Por ter um histórico de acesso preciso para as páginas, o gerenciador de memória do host 241 pode executar seus algoritmos de envelhecimento inteligente e corte de conjuntos de trabalho de processo. Isso pode permitir que o gerenciador de memória do host 241 examine o estado de todo o sistema e faça escolhas inteligentes sobre quais endereços cortar e/ou paginar para o disco, ou outras operações

semelhantes, para aliviar a pressão da memória se necessário ou por outras razões.

[0061] Em algumas modalidades, limites de memória virtual e física podem ser impostos em máquinas virtuais pelo gerenciador de memória do host 241, assim como qualquer outro processo no sistema. Isso pode ajudar a área restrita do administrador do sistema ou, de outra forma, restringir ou habilitar máquinas virtuais, como as máquinas virtuais exemplares 261 e 271. O sistema host pode usar os mesmos mecanismos para fazer isso como faria para processos nativos. Por exemplo, em algumas modalidades onde um desempenho superior é desejado, uma máquina virtual pode ter a totalidade de sua memória física do convidado apoiada diretamente pela memória física do host. Alternativamente, as partes podem ser apoiadas pela memória virtual do host, enquanto outras porções são apoiadas pela memória física do host 120. Ainda como outro exemplo, onde o desempenho inferior pode ser aceitável, uma máquina virtual pode ser apoiada principalmente pela memória virtual do host e a memória virtual do host pode ser limitada a menos do que o apoio total pela memória física do host. Por exemplo, uma máquina virtual, como a máquina virtual exemplar 271, pode apresentar uma memória física do convidado de 4 GB de tamanho, que, por sua vez, utilizando os processos descritos acima, pode ser apoiada por 4 GB de memória virtual do host. No entanto, a memória virtual do host pode ser limitada para ter apenas 2 GB de memória física do host. Isso pode causar paginação para o disco ou outros obstáculos de desempenho, mas pode ser uma maneira de os administradores acelerarem com base nos níveis de serviço ou exercerem outro controle na implantação da máquina virtual. Da mesma forma, uma certa quantidade de memória física pode ser garantida para uma máquina virtual (enquanto ainda permite que a máquina virtual seja apoiada pela memória virtual do host) conforme suportado

pelo gerenciador de memória do host para fornecer um certo nível consistente de desempenho.

[0062] Uma quantidade de memória física do convidado apresentada por dentro do ambiente de máquina virtual pode ser alterada dinamicamente. Quando a memória física do convidado precisa ser adicionada, outra faixa de endereço de memória virtual do host pode ser alocado conforme descrito anteriormente. Uma vez que a pilha de virtualização 230 está pronta para lidar com intercepções de acesso na memória, a faixa de endereço de memória física do convidado pode ser adicionada ao ambiente da máquina virtual. Ao remover os endereços de memória física do convidado, as porções da faixa de endereço virtual do host apoiando os endereços de memória física do convidado removidos podem ser liberadas pelo gerenciador de memória do host 241 (e atualizado de acordo com as estruturas de dados da pilha de virtualização 230). Alternativamente, ou além disso, várias APIs do gerenciador de memória do host podem ser chamadas nessas porções da faixa de endereço virtual do host para liberar as páginas de memória física do host sem liberar o espaço de endereço virtual do host para ele. Alternativamente, em algumas modalidades, é possível fazer nada, uma vez que o gerenciador de memória do host 241 pode eventualmente cortar essas páginas do conjunto de trabalho e, eventualmente, gravá-las no disco, como um arquivo de página, porque elas não serão mais acessadas em prática.

[0063] De acordo com um aspecto, a SLAT 220 pode ser pré-preenchida com alguns ou todos os mapeamentos de endereços de memória física do convidado para os endereços de memória física do host. Isso pode reduzir o número de operações de tratamento de falhas realizadas na inicialização da máquina virtual. No entanto, conforme a máquina virtual opera, as entradas na SLAT 220 podem ser invalidadas por várias razões, e o tratamento de falhas descrito acima

pode ser usado para mais uma vez correlacionar endereços de memória física do convidado com endereços de memória física do host. As entradas SLAT podem ser pré-preenchidas antes da inicialização do ambiente de computação virtual ou no tempo de execução. Além disso, toda a SLAT 220 ou apenas partes dela podem ser pré-preenchidas.

[0064] Outra otimização pode ser a pré-busca nas porções adicionais da memória física do host da memória virtual do host, apoiando a memória física do convidado, que foram previamente retirados de página, de modo que, quando uma interceptação de memória subsequente chega, a falha virtual pode ser satisfeita mais rapidamente, uma vez que pode ser satisfeita sem ter que ir para o disco para ler os dados.

[0065] Voltando à Figura 4, um método 400 é ilustrado fornecendo uma visão geral dos mecanismos descritos em detalhes acima. O método 400 pode incluir atos para apoiar a memória física do convidado com a memória virtual do host. O método inclui, a partir de uma máquina virtual em execução em um dispositivo de computação host, a tentativa de acessar a memória física do convidado usando um acesso à memória física do convidado (ato 402). Por exemplo, o ambiente de máquina virtual 271, mostrado na Figura 2, pode acessar a memória física do convidado que, para processos em execução dentro do ambiente de máquina virtual 271, aparece como memória física atual.

[0066] O método 400 pode ainda incluir determinar que o acesso à memória física do convidado se refere a um endereço de memória física do convidado que não tem uma entrada válida em uma estrutura de dados que correlaciona endereços de memória física do convidado com endereços de memória física do host (ato 404). Por exemplo, pode-se determinar que não há entrada válida na SLAT 220, mostrado na Figura 2.

[0067] Como resultado, o método 400 pode incluir a identificação de um endereço de memória virtual do host que corresponde ao endereço de memória física do convidado e a identificação de um endereço de memória física do host que corresponde ao endereço de memória virtual do host (ato 406). Por exemplo, a pilha de virtualização 230, mostrada na Figura 2, pode identificar um endereço de memória virtual do host que corresponde ao endereço de memória física do convidado e o gerenciador de memória 241, também mostrado na Figura 2, pode identificar um endereço de memória física do host que corresponde ao endereço de memória virtual do host.

[0068] O método 400 pode incluir ainda a atualização da estrutura de dados que correlaciona endereços de memória física do convidado com endereços de memória física do host com uma correlação do endereço de memória física do convidado e o endereço de memória física do host identificado (ato 408). Por exemplo, a pilha de virtualização 230, mostrada na Figura 2, pode obter o endereço de memória física do host a partir do gerenciador de memória 241, também mostrado na Figura 2, e atualizar a SLAT 220 (mostrada na Figura 2) com uma correlação do endereço de memória física do convidado e o endereço de memória física do host identificado.

[0069] O método 400 pode ser praticado causando uma interceptação. A interceptação pode ser encaminhada para uma pilha de virtualização no host. Isso pode fazer com que a pilha de virtualização identifique o endereços de memória virtual do host que corresponde ao endereços de memória física do convidado e emita uma falha a um gerenciador de memória para obter o endereços de memória física do host que corresponde ao endereço de memória virtual do host. A pilha de virtualização pode então atualizar a estrutura de dados que correlaciona os endereços de memória física do convidado com os endereços de memória física do host com uma correlação do endereços de me-

mória física do convidado e o endereços de memória física do host identificados.

[0070] O método 400 pode ainda incluir determinar um tipo para o acesso à memória física do convidado e atualizar a estrutura de dados que correlaciona os endereços de memória física do convidado com os endereços de memória física do host com o tipo determinado correlacionado ao endereços de memória física do convidado e o endereço de memória física do host identificado. Por exemplo, se o acesso à memória física do convidado for uma leitura, a SLAT pode ser atualizada para indicar isso.

[0071] O método 400 pode ainda incluir a realização de uma ação que pode alterar um endereço de memória física do host apoiando o endereço de memória virtual do host. Como resultado, o método pode incluir invalidar uma entrada que correlaciona o endereço de memória física do convidado com o endereço de memória física do host na estrutura de dados que correlaciona os endereços de memória física do convidado com os endereços de memória física do host. Isso pode causar o acesso subsequente ao endereços de memória física do convidado para gerar uma falha que pode ser usada para atualizar a estrutura de dados que correlaciona os endereços de memória física do convidado com os endereços de memória física do host com uma correlação correta para a memória virtual do host apoiando a memória física do convidado. Por exemplo, a ação pode incluir uma operação de combinação de página. A combinação de páginas pode ser usada para aumentar a densidade de máquinas virtuais em um host.

[0072] O método 400 pode incluir a inicialização da máquina virtual do convidado. Como parte da inicialização da máquina virtual do convidado, o método 400 pode incluir pré-preencher pelo menos uma parte da estrutura de dados que correlaciona os endereços de memória física do convidado com endereços de memória física do host com al-

guns ou todos os endereços de memória física do convidado para hospedar mapeamentos de endereço de memória física para a máquina virtual do convidado. Assim, por exemplo, a memória física do host pode ser pré-alocada para uma máquina virtual e as correlações apropriadas inseridas na SLAT. Isso resultaria em menos exceções sendo necessárias para inicializar a máquina virtual do convidado.

[0073] Com referência agora à Figura 5, um exemplo de fluxo 500 é ilustrado mostrando várias ações que podem ocorrer ao longo de uma parte do tempo de vida de alguns dados em um endereço de memória física do convidado de exemplo (doravante abreviado como "GPA") 0x1000. Conforme ilustrado em 502, uma entrada SLAT para GPA 0x1000 é inválida, o que significa que não há entrada SLAT para esse GPA específico. Conforme ilustrado em 504, uma máquina virtual tenta realizar uma leitura em GPA 0x1000 provocando uma interceptação de leitura da máquina virtual (doravante abreviada como "VM"). Conforme ilustrado em 506, um hipervisor encaminha a interceptação para a pilha de virtualização no host. Em 508, a pilha de virtualização executa uma pesquisa de virtualização para um endereço de memória virtual (VA) do host correspondente ao GPA 0x1000. A pesquisa produz VA 0x8501000. Em 510, uma falha virtual é gerada para um acesso de leitura no VA 0x8501000. O tratamento de falha virtual do gerenciador de memória retorna um endereço físico do sistema (doravante abreviado como "SPA") de 0x88000 definindo um endereço na memória do sistema onde os dados em GPA 0x1000 estão fisicamente localizados. Portanto, conforme ilustrado em 514, a SLAT é atualizada para correlacionar GPA 0x1000 com SPA 0x88000 e para marcar o acesso aos dados como "Somente para leitura". Em 516, a pilha de virtualização conclui o tratamento de interceptação de leitura e o hipervisor retoma a execução da máquina virtual do convidado.

[0074] Conforme ilustrado em 518, o tempo passa. Em 520, a má-

quina virtual tenta acesso de gravação em GPA 0x1000. Em 522, o hipervisor encaminha o acesso de gravação para a pilha de virtualização no host. Em 524, a pilha de virtualização executa uma pesquisa de virtualização para VA de host para GPA 0x1000. Conforme observado anteriormente, ele está em VA 0x8501000. Em 526, uma falha virtual ocorre para acesso de gravação em VA 0x8501000. Em 528, a falha virtual retorna SPA 0388000 na memória física. Em 530, a entrada SLAT para GPA 0x1000 é atualizada para indicar que o acesso aos dados é "Leitura / Gravação". Em 532, a pilha de virtualização conclui o tratamento de interceptação de gravação. O hipervisor retoma a execução da máquina virtual do convidado.

[0075] Conforme ilustrado em 534, algum tempo passa. Em 536, o gerenciador de memória do host executa uma passagem de combinação de páginas para combinar quaisquer páginas na memória física do host que sejam funcionalmente idênticas. Em 538, o gerenciador de memória do host encontra candidatos combinados para VA 0x8501000 e outro endereço virtual em outro processo. Em 540, o host executa uma liberação TLB para VA 0x8501000. Em 542, a pilha de virtualização intercepta a liberação de TLB. Em 544, a entrada SLAT para GPA 0x1000 é inválida. Em 546, uma interceptação de máquina virtual é executada para GPA 0x1000. Em 548, ocorre uma falha virtual para acesso de leitura em VA 0x8501000. Em 550, a falha virtual retorna SPA 0x52000, que é a página compartilhada entre N processos da passagem de combinação de página em 536. Em 552, a entrada SLAT para GPA 0x1000 é atualizada para se correlacionar com SPA 0x52000, com acesso definido como "Somente para leitura".

[0076] Conforme ilustrado em 554, algum tempo passa. Em 556, uma interceptação de gravação de máquina virtual ocorre para GPA 0x1000. Em 558, ocorre uma falha virtual para acesso de gravação em VA 0x8501000. Em 560, o gerenciador de memória do host realiza

uma gravação de cópia em VA 0x8501000. Em 562, o host realiza uma liberação TLB para VA 0x850100. Conforme ilustrado em 564, isso faz com que a entrada SLAT para GPA 031000 seja invalidada. Em 566, uma falha virtual retorna SPA 0x11000, que é uma página privada após a gravação de cópia. Em 568, a entrada SLAT para GPA 0x1000 é atualizada para SPA 0x1000 com acesso definido como "Leitura / Gravação". Em 570, a pilha de virtualização conclui o tratamento de interceptação de leitura e o hipervisor retoma a execução da máquina virtual.

[0077] Assim, como ilustrado acima, o espaço de endereço físico da máquina virtual é apoiado pela memória virtual do host (normalmente alocado em um espaço de endereço do usuário do processo do host), que está sujeito ao gerenciamento regular da memória virtual pelo gerenciador de memória do host. A memória virtual que apoia a memória física da máquina virtual pode ser de qualquer tipo suportado pelo gerenciador de memória do host 118 (alocação privada, mapeamento de arquivo, mapeamentos de seção apoiados por arquivo de paginação, alocação de grande página, etc.). Um gerenciador de memória do host pode realizar suas operações existentes e aplicar políticas na memória virtual e/ou aplicar políticas especializadas sabendo que a memória virtual está apoiando o espaço de endereço físico da máquina virtual conforme necessário.

[0078] Voltando à Figura 6, o sistema exemplar 600 mostrado nele compreende uma representação em bloco dos mecanismos descritos na Figura 4 e detalhados na Figura 5, conforme descrito acima. Mais especificamente, a memória virtual do convidado 311, memória física do host 312, memória virtual do host 321 e memória física do host 322 são ilustradas como blocos retangulares cuja largura representa uma faixa de memória. Para simplificar a ilustração, os blocos retangulares são de largura aproximadamente igual, embora, na operação prática,

uma quantidade de memória virtual provavelmente exceda a quantidade de memória física que suporta tal memória virtual.

[0079] Conforme descrito em detalhes acima, quando um processo em execução dentro de um ambiente de computação de máquina virtual, como o ambiente de computação de máquina virtual exemplar 271, mostrado na Figura 2, acessa uma parte da memória virtual apresentada pelo sistema operacional em execução dentro de tal ambiente de computação de máquina virtual, uma tabela de página desse sistema operacional, como a tabela de página exemplar 283, pode compreender uma PTE que pode correlacionar o endereço de memória virtual acessado a um endereço de memória física, pelo menos como percebido por processos em execução no ambiente de computação da máquina virtual. Assim, por exemplo, se a memória virtual representada pela região 611 for acessada, uma PTE 631, que pode ser parte da tabela de página 283, pode correlacionar essa região 611 da memória virtual do convidado 311 a uma região 621 da memória física do convidado 312. Conforme descrito em detalhes acima, a memória física do host 312 pode ser apoiada por uma porção da memória virtual do host 321. Assim, a pilha de virtualização acima descrita 230 pode compreender uma estrutura de dados que pode incluir um entrada 651 que pode correlacionar a região 621 da memória física do convidado 312 a uma região 641 da memória virtual do host 321. Os processos em execução no dispositivo de computação host, incluindo a tabela de página acima descrita 250, podem então correlacionar a região 641 da memória virtual do host 321 para uma região 661 da memória física do host 322. Como também descrito em detalhes acima, a SLAT 220 pode compreender um conjunto hierarquicamente organizado de tabelas que podem ser atualizadas com uma entrada de tabela 691 que pode correlacionar a região 621, da memória física do convidado 312, para a região 661 da memória física do host 322. Mais especificamente,

mecanismos, como aqueles implementados pela pilha de virtualização 230, podem detectar a correlação entre a região 621 e a região 661 e podem atualizar, ou gerar, uma entrada de tabela, tal como a entrada 691, dentro da SLAT 220, conforme ilustrado pela ação 680, para incluir tal correlação. Desse modo, a memória física do host 312 pode ser apoiada pela memória virtual do host 321, enquanto continua a permitir que os mecanismos existentes, incluindo a SLAT 220, funcionem em sua maneira tradicional. Por exemplo, um acesso subsequente da região 611 da memória virtual do convidado 311, de um processo em execução dentro desse ambiente de computação de máquina virtual, pode implicar duas pesquisas de tabela de página de uma maneira tradicional: (1) a pesquisa de tabela de página realizada dentro o ambiente de computação de máquina virtual por referência à tabela de página 283, que pode compreender uma entrada, como a entrada 631, que pode correlacionar a região solicitada 611 com a região 621 correspondente na memória física do convidado 312, e (2) a pesquisa de tabela de página realizada no dispositivo de computação host por referência à SLAT 220, que pode compreender uma entrada, como a entrada 691, que pode correlacionar a região 621 na memória física do host 312 à região 661 na memória física do host 322, completando assim um caminho para os dados relevantes conforme armazenados na memória física do host.

[0080] Como também detalhado acima, as várias tabelas de página, tal como a tabela de página exemplar 283 no ambiente de computação de memória virtual, a tabela de página exemplar 250 no dispositivo de computação host e a SLAT 220 exemplar, podem compreender camadas de tabelas hierarquicamente organizadas, de modo que as entradas na camada hierarquicamente mais baixa de tabelas possam identificar "páginas pequenas" de memória, como esse termo é explicitamente definido neste documento, que pode ser a menor faixa de en-

dereços de memória que pode ser individual e discretamente mantida e operada pelos processos utilizando as respectivas tabelas de páginas. No sistema exemplar 600 mostrado na Figura 6, as múltiplas entradas de tabela de página (PTEs) 631, 671 e 691 podem estar nessa camada mais baixa de tabelas, de modo que as regiões 611, 621, 641 e 661 podem ser páginas pequenas de memória. Conforme detalhado acima, em uma arquitetura de microprocessador comum, tais páginas pequenas de memória, conforme esse termo é explicitamente definido neste documento, podem compreender 4 KB de endereços de memória contíguos.

[0081] De acordo com um aspecto, para aumentar a velocidade e eficiência de acesso à memória de um ambiente de computação de máquina virtual cuja memória física é apoiada por memória virtual de um dispositivo de computação host, a SLAT 220 pode manter correlações em camadas de tabelas hierarquicamente mais altas. Em outras palavras, a SLAT 220 pode correlacionar "páginas grandes" de memória ou "páginas enormes" de memória, conforme esses termos são explicitamente definidos neste documento. Em uma arquitetura de microprocessador comum, grandes páginas de memória, como esse termo é explicitamente definido neste documento, podem compreender 2 MB de endereços de memória contíguos e grandes páginas de memória, como esse termo é explicitamente definido neste documento, podem compreender 1 GB de endereços de memória contíguos. Mais uma vez, conforme indicado anteriormente, os termos "página pequena", "página grande" e "página enorme" são explicitamente definidos com referência à camada hierárquica, dentro de um conjunto hierarquicamente organizado de tabelas de páginas, e não são definidas com base em uma quantidade específica de memória, uma vez que tais quantidades de memória podem variar entre diferentes arquiteturas de microprocessador, arquiteturas de sistema operacional e semelhantes.

[0082] Voltando à Figura 7, um mecanismo para criar grandes entradas de página na SLAT 220 e, assim, aumentar a velocidade e eficiência de acesso à memória de um ambiente de computação de máquina virtual cuja memória física é apoiada por memória virtual de um dispositivo de computação host, é mostrada. Como na Figura 6, no sistema 700, mostrado na Figura 7, a memória virtual do convidado 311, a memória física do convidado 312, a memória virtual do host 321 e a memória física do host 322 são ilustradas como blocos retangulares cuja largura representa uma faixa de memória. Além disso, o acesso da região 611 pode prosseguir da maneira detalhada acima e mostrada na Figura 6. No entanto, em vez de gerar uma entrada de tabela de nível hierarquicamente mais baixo dentro da SLAT 220, os mecanismos descritos neste documento podem gerar uma entrada de tabela de nível superior, como a entrada de tabela exemplar 711. Conforme detalhado acima, tal entrada de tabela de nível superior pode correlacionar páginas grandes, ou páginas enormes, de memória. Por conseguinte, quando a pilha de virtualização 230 faz com que a entrada da tabela 711 seja inserida na SLAT 220, a entrada da tabela 711 pode correlacionar uma região de tamanho de página grande 761 da memória física do convidado 312, a uma região de tamanho de página grande 720 do host da memória física do convidado 322. Como será detalhado abaixo, no entanto, tais regiões de tamanho de página grande não são páginas grandes da perspectiva do gerenciador de memória em execução no dispositivo de computação host e utilizando a tabela de página 250 e, como resultado, essa página de grandes regiões dimensionadas não são, por exemplo, paginadas como uma única página grande ou, de outra forma, tratadas como uma única página grande indivisível pelo gerenciador de memória em execução no dispositivo de computação host.

[0083] Pelo contrário, de modo a manter as vantagens de eficiên-

cia, densidade e outras vantagens fornecidas pelos mecanismos detalhados acima que permitem que a memória física do host, tal como memória física do host exemplar 312, seja respaldada por memória virtual do host, tal como memória virtual do host exemplar 321, a tabela de página 250, em execução no host, pode compreender, não uma entrada de tabela de nível superior análoga à entrada de tabela de nível superior 711 na SLAT 220, mas pode simplesmente compreender entradas de tabela de nível inferior, como as entradas de tabela de nível inferior exemplares 671, 731 e 732, mostradas na Figura 7. Mais especificamente, um conjunto específico de entradas de tabela de nível inferior na tabela de página 250, tal como as entradas de tabela de nível inferior exemplares 671, 731, 732 e assim por diante, podem identificar faixas contíguas de memória física do host 322 de modo que a faixa contígua agregada de memória física do host 322 identificado por toda a sequência equivalente da tabela de nível inferior entradas na tabela de página 250 englobam a mesma faixa de memória que a única entrada de tabela de nível superior 711 na SLAT 220. Assim, como um exemplo, em uma arquitetura de microprocessador onde páginas pequenas são de 4 KB em tamanho e páginas grandes têm 2 MB de tamanho, 512 entradas de página pequena contígua na tabela de páginas 250 podem, em conjunto, identificar 2 MB contíguos da memória física do host 322.

[0084] Porque uma falha virtual pode ter sido gerada para menos do que uma região de memória de tamanho de página grande, apenas algumas específicas das entradas da tabela de nível inferior, como as entradas de tabela de nível inferior exemplares 671, 731 e 732, podem ser utilizadas, enquanto as entradas restantes da tabela de nível inferior podem permanecer marcadas como memória livre. Por exemplo, um aplicativo em execução dentro de um ambiente de computação de máquina virtual utilizando a memória física do host 312 pode inicial-

mente requerer apenas uma pequena quantidade de memória configurada por página. Em tal caso, a única entrada de tabela de nível superior 711 pode identificar uma região de tamanho de página grande da memória, como a região de tamanho de página grande 720, como sendo utilizada. No entanto, da perspectiva do gerenciador de memória do dispositivo de computação host, referindo-se à tabela de página 250, apenas uma única das entradas da tabela de nível inferior, como a entrada de tabela de nível inferior exemplar 671, pode ser indicada como sendo utilizada, sendo as restantes indicadas como disponíveis. Para evitar que outras entradas sejam tratadas como disponíveis, a pilha de virtualização 230 pode marcar as entradas de tabela de nível inferior restantes, tais como as entradas de tabela de nível inferior exemplares 731, 732, etc. como sendo usadas. Quando o aplicativo em execução dentro do ambiente de computação da máquina virtual, então, precisa utilizar quantidades adicionais de memória de tamanho de página pequena, a pilha de virtualização 230 pode utilizar o local de memória indicado anteriormente como sendo usado, mas que não foi, de fato, ainda sido utilizado pelo aplicativo em execução no ambiente de computação da máquina virtual, para satisfazer as necessidades de memória adicionais do aplicativo.

[0085] De acordo com um aspecto, o rastreamento da sequência equivalente de entradas de tabela de nível inferior na tabela de página 250, como a marcação acima mencionada, pode ser coordenado pela pilha de virtualização 230, conforme ilustrado pela ação 730 na Figura 7. Em tal instância, se o gerenciador de memória em execução no host faz alterações em uma ou mais dessas entradas de tabela de nível inferior, a pilha de virtualização 230 pode acionar as alterações apropriadas para as entradas correspondentes na SLAT 220, tal como a entrada da tabela de nível superior 711.

[0086] A sequência de entradas de tabela de nível inferior na tabe-

la de página 250, que é equivalente à entrada de tabela de nível superior 711 na SLAT 220, pode correlacionar uma sequência de faixas de memória virtual do host, tal como as faixas exemplares 641, 741 e 742, para hospedar faixas de memória física, tais como as faixas exemplares 661, 721 e 722, respectivamente. Consequentemente, uma região de tamanho de página grande da memória virtual do host 321, como a região de tamanho de página grande 751 exemplificativa, é correlacionada a uma região de tamanho de página grande da memória física do host 322, tal como a região de tamanho de página grande 720 exemplar, através da sequência de entradas de tabela de nível inferior na tabela de página 250, ou seja, as entradas de tabela de nível inferior exemplares 671, 731, 732, etc. A pilha de virtualização 230 pode manter estruturas de dados, tais como tabelas hierarquicamente organizadas, que podem coordenar locais de memória virtual do host para locais de memória física do convidado. Assim, a pilha de virtualização pode manter uma ou mais entradas de dados, tal como a entrada de dados exemplar 771, que pode correlacionar a região de tamanho de página grande 751 da memória virtual do host 321 a uma região de tamanho de página grande correspondente 761 da memória física do host 312. Embora a entrada de dados 771 seja ilustrada como uma única entrada, a pilha de virtualização 230 pode manter a correlação acima mencionada como uma única entrada, como múltiplas entradas, tais como por sub-porções correspondentes individualmente da região 761 com a região 751, incluindo sub-porções que são dimensionadas como páginas pequenas, ou combinações das mesmas. Para completar o ciclo, a entrada de tabela de nível superior acima mencionada 711 na SLAT 220 pode então correlacionar a região de tamanho de página grande 761, na memória física do convidado 312, à região de tamanho de página grande 720, na memória física do host 322. Mais uma vez, a partir da perspectiva da SLAT 220, tendo apenas

uma única entrada de tabela de nível superior 711, as regiões de tamanho de página grande 720 e 761 são páginas grandes que são tratadas unitariamente pela SLAT (de acordo com a única entrada de tabela de nível superior 711), enquanto, da perspectiva do gerenciador de memória em execução no host, a tabela de página 250 e, de fato, a pilha de virtualização 230, tais regiões não são unitárias, mas sim compostas de pequenas regiões de página individualmente gerenciáveis identificadas por, por exemplo, nível inferior entradas de tabela na tabela de página 250.

[0087] A disponibilidade de páginas pequenas de memória contíguas na memória física do convidado 312, equivalente a uma ou mais páginas de memória grandes ou enormes, como esses termos foram definidos neste documento, pode ser direta porque a memória física do convidado 312 é percebida, por processos em execução dentro do ambiente de computação de máquina virtual correspondente, como sendo memória física. Consequentemente, as descrições abaixo enfocam a contiguidade da memória física do host 322. Além disso, porque a memória física do convidado 312 é percebida como sendo memória física, permissões de acesso total à memória (como permissões de "leitura / gravação / execução") permanecem em toda a faixa da memória física do convidado 312 e, consequentemente, é improvável que haja descontinuidade das permissões de acesso à memória entre as entradas da tabela de nível inferior na tabela de páginas 250.

[0088] Com foco na tabela de páginas 250 do dispositivo de computação host, se uma ou mais regiões de tamanho de página grande da memória física do host 322 permanecerem sem uso, então, de acordo com um aspecto, quando uma falha virtual, conforme detalhado acima, é acionada em uma página pequena dentro de uma faixa de memória delineada em uma região de tamanho de página grande que permanece sem uso, as páginas pequenas remanescentes dentro

dessa região de tamanho de página grande, uma única entrada de tabela de nível superior, tal como a entrada da tabela de nível superior exemplar 711, pode ser gerada na SLAT 220.

[0089] No entanto, porque a página grande 720, correspondente à entrada de tabela de nível superior 711, é, da perspectiva da tabela de página 250, uma sequência de páginas pequenas, como as páginas pequenas exemplares 661, 721, 722, etc., devido à sequência de entradas de tabela de nível inferior, como as entradas de tabela de nível inferior exemplificativas 671, 731, 732, etc., os mecanismos de gerenciamento de memória utilizando tal tabela de página 250 podem tratar cada das páginas pequenas individualmente, o que pode incluir paginar essas páginas pequenas para um meio de armazenamento não volátil, tal como uma unidade de disco rígido. Em tal caso, de acordo com um aspecto, a entrada de tabela de nível superior 711, na SLAT 220, pode ser substituída por uma sequência equivalente de entradas de tabela de nível inferior. Mais especificamente, as entradas da tabela de nível inferior geradas na SLAT 220 podem ser contíguas e podem, em conjunto, fazer referência à mesma faixa de endereços de memória que a entrada da tabela de nível superior 711, com exceção daquelas específicas das entradas da tabela nível inferior, correspondentes às páginas que foram paginadas pelo gerenciador de memória, podem estar ausentes ou não foram criadas.

[0090] De acordo com outros aspectos, se uma página grande, ou página enorme, região dimensionada da memória física do host 322 estiver indisponível no momento em que uma falha virtual, conforme detalhado acima, é acionada, uma página grande disponível, ou página enorme, região dimensionada da memória física do host pode ser construída no momento da interceptação pela montagem de uma quantidade apropriada de páginas pequenas contínuas. Por exemplo, uma região com tamanho de página grande pode ser construída a par-

tir de páginas pequenas contíguas que estão disponíveis em uma "lista livre" ou outra enumeração semelhante de páginas de memória física do host disponíveis. Como outro exemplo, se páginas pequenas que estão sendo utilizadas por outros processos são necessárias para estabelecer a continuidade necessária para montar uma quantidade suficiente de páginas pequenas em uma região de tamanho de página grande, essas páginas pequenas podem ser obtidas de outros processos, e os dados nele contidos podem ser transferidos para outras páginas pequenas ou podem ser transferidos para o disco. Uma vez que essa região de tamanho de página grande é construída, o processamento da falha virtual pode prosseguir da maneira detalhada acima, incluindo a geração de uma entrada de tabela de nível superior, como a entrada de tabela de nível superior exemplar 711, na SLAT 220.

[0091] Em alguns casos, um gerenciador de memória em execução no dispositivo de computação host, ao utilizar a tabela de páginas 250, pode implementar mecanismos de plano de fundo ou oportunistas pelos quais as regiões de tamanho de página grande ou página enorme disponíveis são construídas a partir de uma quantidade suficiente de páginas pequenas contíguas. Na medida em que tais mecanismos resultam em uma região de tamanho de página grande disponível, por exemplo, quando uma falha virtual é acionada, o processamento pode prosseguir da maneira indicada acima. No entanto, na medida em que tais mecanismos ainda não concluíram a montagem de uma região de tamanho de página grande disponível, a construção de uma região de tamanho de página grande pode ser concluída no momento da interceptação com uma prioridade superior. Por exemplo, um processo de construção de uma região de tamanho de página tão grande pode receber uma prioridade de execução superior ou pode ser movido do plano de fundo para o primeiro plano ou pode ser executado de forma mais agressiva.

[0092] Alternativamente, se uma região de tamanho de página grande da memória física do host não estiver disponível quando uma falha virtual for acionada, o processamento pode prosseguir conforme detalhado acima com a utilização de múltiplas páginas pequenas. Posteriormente, quando uma região de tamanho de página grande se torna disponível, os dados de múltiplas páginas pequenas anteriormente utilizadas (devido à indisponibilidade de uma região de tamanho de página grande no momento) podem ser copiados para a página grande, e as páginas pequenas anteriormente utilizadas podem ser liberadas posteriormente. Em tal caso, as entradas de tabela de nível inferior na SLAT 220 podem ser substituídas por uma entrada de tabela de nível superior, como uma entrada de tabela de nível superior exemplar 711.

[0093] Além disso, os parâmetros sob os quais regiões de tamanho de página grande são construídas, seja no momento de interceptação ou oportunamente, podem ser variados de acordo com um equilíbrio entre o acesso à memória mais rápido, devido ao salto de camadas hierárquicas de tabelas durante uma pesquisa SLAT e o atraso inicial na construção de uma região de tamanho de página grande a partir de páginas pequenas. Um desses parâmetros pode ser uma quantidade de páginas pequenas contínuas suficientes para acionar a negociação, ou retirada de página, de outras páginas pequenas necessárias para completar a contiguidade em uma região inteira de tamanho de página grande. Por exemplo, se 512 páginas pequenas contíguas são necessárias para construir uma página grande e existem faixas contíguas de 200 páginas pequenas e 311 páginas pequenas, com uma página pequena entre elas sendo atualmente utilizada por outro processo, uma página pequena pode ser retirada da página ou negociada com outra página pequena disponível, de modo que todas as 512 páginas pequenas contíguas possam estar disponíveis para

serem construídas em uma região de tamanho de página grande. Nesse exemplo, a fragmentação das páginas pequenas disponíveis pode ser muito baixa. Em contraste, faixas contíguas de apenas 20 a 30 páginas pequenas que são continuamente interrompidas por outras páginas pequenas que estão sendo utilizadas atualmente por outros processos podem descrever uma faixa de páginas pequenas com uma fragmentação muito alta. Embora esse conjunto altamente fragmentado de páginas pequenas ainda possa ser negociado, retirado da página ou de outra forma reorganizado para gerar uma faixa contígua de páginas pequenas equivalente a uma região de tamanho de página grande, esse esforço pode levar muito mais tempo. Consequentemente, de acordo com um aspecto, um limite de fragmentação pode ser definido, o que pode delinear se os esforços forem realizados para construir uma região de tamanho de página grande a partir de faixas de páginas pequenas com fragmentação maior ou menor.

[0094] De acordo com um aspecto, a construção de uma região de tamanho de página grande pode tirar vantagem das utilizações existentes de páginas pequenas por processos cuja memória está sendo utilizada para apoiar a memória física do convidado de ambientes de computação de máquina virtual. Como um exemplo simples, se faixas contíguas de 200 páginas pequenas e 311 páginas pequenas já tiverem sido utilizadas pelo processo cuja memória apoia a memória física do convidado 312, e apenas uma página pequena entre essas duas faixas contíguas é necessária para estabelecer uma faixa contígua de páginas pequenas equivalente a uma página grande, então, uma vez que essa página pequena é liberada, tal como por ser retirada da página ou negociada, os únicos dados que precisam ser copiados podem ser os dados correspondentes àquela página pequena, com os dados restantes capazes de permanecer no lugar, mesmo que as entradas de tabela de nível inferior correspondentes na SLAT 220 possam ser

invalidadas e substituídas por uma única entrada de tabela de nível superior abrangendo a mesma faixa de memória.

[0095] Para evitar ter que reconstruir regiões de memória de tamanho de página grande a partir de faixas contíguas de páginas pequenas, um conjunto existente de entradas de tabela contíguas de nível inferior na tabela de página 250 que engloba, em agregado, uma faixa de memória equivalente como a única entrada de tabela de nível superior 711 na SLAT 220, pode ser bloqueada para que possam evitar a retirada de página. Assim, por exemplo, a pilha de virtualização 230 pode solicitar que as entradas da tabela de nível inferior 671, 731, 732, etc. sejam bloqueadas de modo que as faixas correspondentes de memória física do host 661, 721, 722, etc. possam reter os dados armazenados nelas, sem que seja retirada de página para o disco. De acordo com um aspecto, a decisão, tal como pela pilha de virtualização 230, para solicitar que tais entradas sejam bloqueadas pode ser dependente de vários fatores. Por exemplo, um tal fator pode ser uma frequência de utilização dos dados armazenados dentro dessas faixas de memória, com os dados utilizados com mais frequência resultando no bloqueio das entradas, e os dados utilizados com menos frequência resultando nas entradas sendo posteriormente desbloqueadas. Outros fatores, como uma quantidade de fragmentação de memória do sistema geral, uma necessidade de maior densidade de ambientes de computação de máquina virtual no dispositivo de computação host e outros fatores semelhantes podem igualmente ser referenciados na decisão de solicitar que entradas de tabela de página, suportando uma entrada de tabela de nível superior na SLAT, sejam bloqueadas.

[0096] Voltando à Figura 8, o diagrama de fluxo 800 mostrado aqui ilustra os mecanismos descritos acima como uma série de etapas exemplificativas. Inicialmente, na etapa 810, uma falha virtual, conforme detalhado acima, pode ocorrer. Subsequentemente, na etapa 815,

uma determinação pode ser feita quanto a se a falha virtual da etapa 810 faz referência à memória dentro de uma região de memória com tamanho de página grande, onde nenhuma outra memória dentro dessa região de tamanho de página grande foi anteriormente referenciada por uma falha virtual. Em particular, a memória física do host pode ser delineada em segmentos de tamanho de página grande. A primeira falha virtual que faz referência à memória dentro de uma região de tamanho de página grande pode resultar em uma determinação positiva na etapa 815, enquanto o processamento pode prosseguir para a etapa 835.

[0097] Na etapa 835, uma única entrada de tabela de nível superior pode ser gerada na SLAT, enquanto, correspondentemente, na etapa 840, múltiplas entradas de tabela de nível inferior podem ser utilizadas na tabela de página do host, onde uma quantidade relevante de múltiplas entradas de tabela de nível inferior é aquela que abrange a mesma faixa de memória que a única entrada de tabela de nível superior gerada na SLAT, ou seja, a faixa de memória correspondente à região delineada de tamanho de página grande de memória contendo o local de memória cujo acesso acionou a falha virtual na etapa 810. Conforme indicado anteriormente, porque a tabela de página do host compreende múltiplas entradas de tabela de nível inferior, em oposição a uma única entrada de tabela de nível superior, o gerenciamento de memória do dispositivo de computação host pode optar por retirada de página, por exemplo, copiando seu conteúdo para o disco, uma ou mais das regiões de memória, ou páginas, identificadas por tais entradas de tabela de nível inferior na tabela de página do host. Se essa retirada de página ocorrer, conforme determinado na etapa 845, o processamento pode prosseguir para a etapa 850 e a entrada da tabela de nível superior na SLAT pode ser substituída por múltiplas entradas de tabela de nível inferior correspondentes às regiões de memória re-

ferenciadas pelas entradas da tabela de nível inferior na tabela de página do host que não foram retiradas de página (e deixando de fora da SLAT as entradas da tabela de nível correspondentes às regiões de memória referenciadas pelas entradas da tabela de nível inferior na tabela de página do host que foram, na verdade, retiradas de página).

[0098] De acordo com um aspecto, se páginas pequenas da região de tamanho de página grande da memória fossem retiradas de página na etapa 845, e a entrada da tabela de nível superior na SLAT fosse invalidada e substituída por uma sequência correspondente de entradas de tabela de nível (com exceção de quaisquer entradas de tabela de nível inferior referenciando as páginas pequenas retiradas de página), então, opcionalmente, o processamento pode prosseguir para a etapa 855 para continuar a verificar se há regiões de memória de tamanho de página grande disponíveis que ficam livres. Conforme detalhado acima, os processos de gerenciamento de memória em execução no dispositivo de computação host, por exemplo, podem reconstruir oportunisticamente regiões de tamanho de página grande de memória disponível. Deveria tais regiões de tamanho de página grande de memória tornarem-se disponíveis, conforme determinado na etapa 855, o processamento pode prosseguir para a etapa 860, onde as entradas de tabela de nível inferior na SLAT podem ser substituídas por uma única entrada de tabela de nível superior na SLAT e, se apropriado, os dados podem ser copiados das regiões da memória referenciadas pelas entradas anteriores da tabela de nível inferior para as regiões da memória agora referenciadas pela única entrada da tabela de nível superior (se as duas regiões da memória forem diferentes).

[0099] Voltando para a etapa 815, se a falha virtual da etapa 810 não foi a primeira a fazer referência à memória dentro de uma região de tamanho de página grande previamente delineada, o processamento pode prosseguir para a etapa 820. Na etapa 820, uma determinação

pode ser feita para determinar se uma página grande deve ser criada no momento da interceptação, como na maneira detalhada acima. Mais especificamente, se as etapas 835 e 840 foram realizadas anteriormente para uma região de memória de tamanho de página grande, e a SLAT agora compreende uma única entrada de tabela de nível superior, enquanto a tabela de página compreende entradas de tabela de nível inferior equivalentes referenciando sub-porções da região de memória referenciada pela única entrada de tabela de nível superior na SLAT, então um acesso de memória subsequente direcionado a quaisquer locais de memória dentro da região de memória referenciada pela única entrada de tabela de nível superior na SLAT não acionaria um falha virtual, conforme detalhado acima. Assim, se uma falha virtual for acionada na etapa 810 e, na etapa 815, for determinado que a falha virtual não é a primeira direcionada para locais de memória dentro de uma região de memória de tamanho de página grande previamente delineada, então a falha virtual desencadeada na etapa 810 foi provavelmente desencadeada porque as etapas 835 e 840 foram realizadas anteriormente para locais de memória dentro dessa região de tamanho de página grande, mas, subsequentemente, uma ou mais das páginas pequenas que compreendem essa região de memória de tamanho de página grande foram paginadas, conforme determinado na etapa 845 e, conseqüentemente, a etapa 850 foi realizada anteriormente. Assim, na etapa 820, pode-se determinar se uma região de memória de tamanho de página grande deve ser construída no momento da interceptação. Se tal região de tamanho de página grande deve ser construída, então o processamento pode prosseguir para a etapa 830 e uma quantidade apropriada de páginas pequenas contíguas pode ser montada, conforme detalhado acima. O processamento pode então prosseguir com a etapa 835 como antes. Por outro lado, se, na etapa 830, nenhuma construção sob demanda de regiões de

tamanho de página grande de memória disponível for solicitada, o processamento pode prosseguir para utilizar entradas de tabela de nível inferior na tabela de página do host e na SLAT de acordo com os mecanismos descritos em detalhes acima com referência às Figuras 4 e 5.

[00100] As descrições acima incluem, como um primeiro exemplo, um método para aumentar a velocidade de acesso da memória do computador, o método compreendendo: detectar um primeiro acesso à memória, a partir de um primeiro processo em execução em um ambiente de computação de máquina virtual, direcionado a uma primeira faixa de memória; gerar, como uma pré-condição para concluir o primeiro acesso à memória, uma primeira entrada em uma tabela de tradução de endereço de segunda camada hierarquicamente organizada, a tabela de tradução de endereço de segunda camada hierarquicamente organizada correlacionando endereços de memória física do host a endereços de memória física do convidado, a primeira entrada sendo pelo menos um nível hierárquico acima de um nível hierarquicamente mais baixo de tabelas, de modo que uma segunda faixa de memória, identificado pela primeira entrada, pode ser identificado sem referência a qualquer tabela no nível hierarquicamente mais baixo de tabelas, a segunda faixa de memória sendo maior do que a primeira faixa de memória; e em resposta à geração da primeira entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada, marcando como usada uma primeira pluralidade de entradas em uma tabela de página hierarquicamente organizada, a tabela de página hierarquicamente organizada correlacionando os endereços de memória física do host para os endereços de memória virtual do host, a primeira pluralidade de entradas referenciando, em agregado, a mesma segunda faixa de memória que a primeira entrada na tabela de tradução de endereços de segunda camada hierarquicamente organi-

zada, em que as entradas da primeira pluralidade de entradas estão em um nível hierarquicamente mais baixo de tabelas; em que os endereços de memória física do convidado são percebidos como sendo endereços para a memória física por processos em execução no ambiente de computação de máquina virtual; em que os endereços de memória física do host são endereços para a memória física atual de um dispositivo de computação host que hospeda o ambiente de computação da máquina virtual; em que os endereços de memória virtual do host são endereços para a memória virtual fornecida por um gerenciador de memória em execução no dispositivo de computação host, o gerenciador de memória utilizando a tabela de página hierarquicamente organizada para fornecer a memória virtual; e em que a memória física do convidado, identificada pelos endereços de memória física do convidado, é apoiada por uma porção da memória virtual do host, identificada por uma porção dos endereços da memória virtual do host.

[00101] Um segundo exemplo é o método do primeiro exemplo, compreendendo ainda: detectar um segundo acesso à memória a partir do primeiro processo em execução no ambiente de computação da máquina virtual, o segundo acesso à memória sendo direcionado para uma terceira faixa de memória que difere da primeira faixa de memória; e satisfazer o segundo acesso à memória por referência a um segundo subconjunto da primeira pluralidade de entradas na tabela de páginas hierarquicamente organizada; em que o primeiro acesso à memória foi satisfeito por referência a um primeiro subconjunto da primeira pluralidade de entradas na tabela de páginas hierarquicamente organizadas, o primeiro subconjunto compreendendo entradas diferentes, na tabela de páginas hierarquicamente organizadas, do segundo subconjunto.

[00102] Um terceiro exemplo é o método do primeiro exemplo, compreendendo ainda: detectar, após o primeiro acesso à memória

ser concluído, que um primeiro subconjunto da primeira pluralidade de entradas na tabela de página hierarquicamente organizada teve dados, que foram originalmente armazenados em endereços de memória física do host correspondentes, subsequentemente retirados de página para um meio de armazenamento não volátil; invalidar, em resposta à detecção, a primeira entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada; e gerar, no lugar da primeira entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada, uma segunda pluralidade de entradas na tabela de tradução de endereço de segunda camada hierarquicamente organizada, a segunda pluralidade de entradas referenciando, em agregado, pelo menos algumas da mesma segunda faixa de memória que a primeira entrada, em que as entradas da segunda pluralidade de entradas estão em um nível hierarquicamente mais baixo de tabelas.

[00103] Um quarto exemplo é o método do terceiro exemplo, em que a segunda pluralidade de entradas faz referência a porções da segunda faixa de memória que foram acessadas anteriormente pelo primeiro processo e não foram retiradas de página.

[00104] Um quinto exemplo é o método do primeiro exemplo, compreendendo ainda: montar uma primeira pluralidade de regiões contíguas de tamanho de página pequena da memória física do host em uma única região de tamanho de página grande da memória física do host.

[00105] Um sexto exemplo é o método do quinto exemplo, em que a montagem ocorre subsequentemente à detecção do primeiro acesso à memória e antes de gerar a primeira entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada.

[00106] Um sétimo exemplo é o método do quinto exemplo, compreendendo ainda: gerar uma segunda entrada na tabela de tradução

de endereço de segunda camada hierarquicamente organizada, a segunda entrada sendo pelo menos um nível hierárquico acima do nível hierarquicamente mais baixo de tabelas de modo que uma terceira faixa de memória, identificada pela segunda entrada, possa ser identificada sem referência a qualquer tabela no nível hierarquicamente mais baixo de tabelas, a terceira faixa de memória referenciando a única região de tamanho de página grande da memória física do host em que a primeira pluralidade das pequenas regiões contíguas de tamanho de página da memória física do host foi montada; e em resposta à geração da segunda entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada, marcar como usada uma segunda pluralidade de entradas na tabela de página hierarquicamente organizada, a segunda pluralidade de entradas referenciando a primeira pluralidade das regiões contíguas de tamanho de página pequena da memória física do host que foi montada em uma única região de tamanho de página grande da memória física do host.

[00107] Um oitavo exemplo é o método do sétimo exemplo, compreendendo ainda: gerar uma segunda entrada na tabela de tradução de endereços da segunda camada hierarquicamente organizada, a segunda entrada sendo pelo menos um nível hierárquico acima do nível hierarquicamente mais baixo de tabelas de modo que uma terceira faixa de memória, identificada pela segunda entrada, possa ser identificada sem referência a qualquer tabela no nível hierarquicamente mais baixo de tabelas, a terceira faixa de memória referenciando a única região de tamanho de página grande da memória física do host na qual a primeira pluralidade das regiões contíguas de tamanho de página pequena da memória física do host foi montada; e em resposta à geração da segunda entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada, marcar como usada uma segunda pluralidade de entradas na tabela de página hierarqui-

camente organizada, a segunda pluralidade de entradas referenciando a primeira pluralidade de regiões contíguas de tamanho de página pequena da memória física do host que foram montadas na única região de tamanho de página grande da memória física do host.

[00108] Um nono exemplo é o método do sétimo exemplo, compreendendo ainda: copiar dados de um segundo conjunto de uma ou mais regiões de tamanho de página pequena da memória física do host para pelo menos uma parte da primeira pluralidade das regiões contíguas de tamanho de página pequena; e invalidar uma segunda pluralidade de entradas na tabela de tradução de endereço de segunda camada hierarquicamente organizada, a segunda pluralidade de entradas compreendendo: (1) um primeiro subconjunto de entradas que fazem referência ao segundo conjunto de uma ou mais regiões de tamanho de página pequena da memória física do host e (2) um segundo subconjunto de entradas que fazem referência a pelo menos alguma da primeira pluralidade de regiões contíguas de tamanho de página pequena da memória física do host, em que as entradas da segunda pluralidade de entradas estão em um nível hierarquicamente mais baixo de tabelas; em que a segunda entrada gerada na tabela de tradução de endereço de segunda camada hierarquicamente organizada é utilizada no lugar da segunda pluralidade de entradas que foram invalidadas.

[00109] Um décimo exemplo é o método do quinto exemplo, em que a montagem da primeira pluralidade de regiões contíguas de tamanho de página pequena da memória física do host compreende a cópia de dados de alguma da primeira pluralidade de regiões contíguas de tamanho de página pequena para outras regiões de tamanho de página pequena da memória física do host que diferem da primeira pluralidade de regiões contíguas de tamanho de página pequena.

[00110] Um décimo primeiro exemplo é o método do décimo exem-

plo, em que a cópia dos dados de algumas da primeira pluralidade de regiões contíguas de tamanho de página pequena para outras regiões de tamanho de página pequena é realizada apenas se uma fragmentação da primeira pluralidade de regiões contíguas de tamanho de página pequena está abaixo de um limite de fragmentação.

[00111] Um décimo segundo exemplo é o método do primeiro exemplo, compreendendo ainda: impedir a paginação da segunda faixa de memória.

[00112] Um décimo terceiro exemplo é o método do décimo segundo exemplo, em que o impedimento da paginação da segunda faixa de memória é realizado apenas se uma ou mais porções da segunda faixa de memória forem acessadas com mais frequência do que um limite de frequência de acesso.

[00113] Um décimo quarto exemplo é o método do décimo segundo exemplo, compreendendo ainda a remoção do impedimento de paginação da segunda faixa de memória se uma ou mais porções da segunda faixa de memória forem acessadas com menos frequência do que um limite de frequência de acesso.

[00114] Um décimo quinto exemplo é o método do primeiro exemplo, em que a segunda faixa de memória tem 2 MB de tamanho.

[00115] Um décimo sexto exemplo é um dispositivo de computação que compreende: uma ou mais unidades de processamento central; memória de acesso aleatório (RAM); e uma ou mais mídias legíveis por computador, compreendendo: um primeiro conjunto de instruções executáveis por computador, que, quando executadas pelo dispositivo de computação, fazem com que o dispositivo de computação forneça um gerenciador de memória referenciando uma tabela de página hierarquicamente organizada para traduzir os endereços de memória virtual do host para hospedar endereços de memória física que identifiquem locais na RAM; um segundo conjunto de instruções executáveis

por computador, as quais, quando executadas pelo dispositivo de computação, fazem com que o dispositivo de computação forneça um ambiente de computação de máquina virtual, em que processos em execução dentro do ambiente de computação de máquina virtual percebem endereços de memória física do convidado como sendo endereços para a memória física, e em que mais memória física do convidado, identificada pelos endereços de memória física do convidado, é apoiada por uma porção da memória virtual do host, identificada por uma porção dos endereços da memória virtual do host; e um terceiro conjunto de instruções executáveis por computador, que, quando executadas pelo dispositivo de computação, fazem com que o dispositivo de computação: detecte um primeiro acesso à memória, a partir de um primeiro processo em execução no ambiente de computação da máquina virtual, direcionado a um primeira faixa de memória; gerar, como uma pré-condição para concluir o primeiro acesso à memória, uma primeira entrada em uma tabela de tradução de endereço de segunda camada hierarquicamente organizada, a tabela de tradução de endereço de segunda camada hierarquicamente organizada correlacionando os endereços de memória física do host aos endereços de memória física do convidado, a primeira entrada sendo pelo menos um nível hierárquico acima de um nível hierarquicamente mais baixo de tabelas, de modo que uma segunda faixa de memória, identificada pela primeira entrada, possa ser identificada sem referência a qualquer tabela no nível hierarquicamente mais baixo de tabelas, a segunda faixa de memória sendo maior do que a primeira faixa de memória; e em resposta à geração da primeira entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada, marcar como usada uma primeira pluralidade de entradas na tabela de página hierarquicamente organizada, a primeira pluralidade de entradas referenciando, em agregado, a mesma segunda faixa de memória como a primeira

entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada, em que as entradas da primeira pluralidade de entradas estão em um nível hierarquicamente mais baixo de tabelas.

[00116] Um décimo sétimo exemplo é o dispositivo de computação do décimo sexto exemplo, em que o terceiro conjunto de instruções executáveis por computador compreende outras instruções executáveis por computador, que, quando executadas pelo dispositivo de computação, fazem o dispositivo de computação: detectar, após a conclusão do primeiro acesso à memória, que um primeiro subconjunto da primeira pluralidade de entradas na tabela de páginas hierarquicamente organizadas teve dados, que foram originalmente armazenados em endereços de memória física do host correspondentes, subsequentemente retirados de página para um meio de armazenamento não volátil; invalidar, em resposta à detecção, a primeira entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada; e gerar, no lugar da primeira entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada, uma segunda pluralidade de entradas na tabela de tradução de endereço de segunda camada hierarquicamente organizada, a segunda pluralidade de entradas referenciando, em agregado, pelo menos algumas da mesma segunda faixa de memória que a primeira entrada, em que as entradas da segunda pluralidade de entradas estão em um nível hierarquicamente mais baixo de tabelas.

[00117] Um décimo oitavo exemplo é o dispositivo de computação do décimo sexto exemplo, em que o terceiro conjunto de instruções executáveis por computador compreende outras instruções executáveis por computador, que, quando executadas pelo dispositivo de computação, fazem o dispositivo de computação: reunir uma primeira pluralidade de regiões contíguas de tamanho de página pequena da

memória física do host em uma única região de tamanho de página grande da memória física do host; em que a montagem ocorre subsequentemente à detecção do primeiro acesso à memória e antes da geração da primeira entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada.

[00118] Um décimo nono exemplo é o dispositivo de computação do décimo sexto exemplo, em que o terceiro conjunto de instruções executáveis por computador compreende outras instruções executáveis por computador, que, quando executadas pelo dispositivo de computação, fazem o dispositivo de computação: evitar a paginação da segunda faixa de memória.

[00119] Um vigésimo exemplo é um ou mais meios de armazenamento legíveis por computador compreendendo instruções executáveis por computador, que quando executadas, fazem um dispositivo de computação: detectar um primeiro acesso à memória, a partir de um primeiro processo em execução em um ambiente de computação de máquina virtual, direcionado a uma primeira faixa de memória; gerar, como uma pré-condição para concluir o primeiro acesso à memória, uma primeira entrada em uma tabela de tradução de endereço de segunda camada hierarquicamente organizada, a tabela de tradução de endereço de segunda camada hierarquicamente organizada correlacionando endereços de memória física do host a endereços de memória física do convidado, a primeira entrada sendo pelo menos um nível hierárquico acima de um nível hierarquicamente mais baixo de tabelas, de modo que uma segunda faixa de memória, identificada pela primeira entrada, possa ser identificada sem referência a qualquer tabela no nível hierarquicamente mais baixo de tabelas, a segunda faixa de memória sendo maior do que a primeira faixa de memória; e em resposta à geração da primeira entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada, marcar como usada

uma primeira pluralidade de entradas em uma tabela de página hierarquicamente organizada, a tabela de página hierarquicamente organizada correlacionando os endereços de memória física do host para endereços de memória virtual a primeira pluralidade de entradas referenciando, em agregado, a mesma segunda faixa de memória que a primeira entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada, em que as entradas da primeira pluralidade de entradas estão em um nível hierarquicamente mais baixo de tabelas; em que os endereços de memória física do convidado são percebidos como sendo endereços para a memória física por processos em execução dentro do ambiente de computação de máquina virtual; em que os endereços de memória física do host são endereços para a memória física atual de um dispositivo de computação host que hospeda o ambiente de computação da máquina virtual; em que os endereços de memória virtual do host são endereços para a memória virtual fornecida por um gerenciador de memória em execução no dispositivo de computação host, o gerenciador de memória utilizando a tabela de páginas hierarquicamente organizada para fornecer a memória virtual; e em que a memória física do convidado, identificada pelos endereços de memória física do convidado, é apoiada por uma porção da memória virtual do host, identificada por uma porção dos endereços da memória virtual do host.

[00120] Como pode ser visto a partir das descrições acima, foram descritos mecanismos pelos quais o acesso à memória através de uma SLAT pode ser acelerado. Tendo em vista as muitas variações possíveis do assunto aqui descrito, reivindicamos como nossa invenção todas as modalidades que possam estar dentro do escopo das seguintes reivindicações e seus equivalentes.

REIVINDICAÇÕES

1. Método para aumentar a velocidade de acesso à memória do computador, **caracterizado** pelo fato de que compreende:

detectar um primeiro acesso à memória, a partir de um primeiro processo em execução em um ambiente de computação de máquina virtual, direcionado a uma primeira faixa de memória;

gerar, como uma pré-condição para completar o primeiro acesso à memória, uma primeira entrada em uma tabela de tradução de endereço de segunda camada hierarquicamente organizada, a tabela de tradução de endereço de segunda camada hierarquicamente organizada correlacionando endereços de memória física do host a endereços de memória física do convidado, a primeira entrada sendo pelo menos um nível hierárquico acima de um nível hierarquicamente mais baixo de tabelas, de modo que uma segunda faixa de memória, identificada pela primeira entrada, pode ser identificada sem referência a qualquer tabela no nível hierarquicamente mais baixo de tabelas, a segunda faixa de memória sendo maior que a primeira faixa de memória; e

em resposta à geração da primeira entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada, marcando como usada uma primeira pluralidade de entradas em uma tabela de página hierarquicamente organizada, a tabela de página hierarquicamente organizada correlacionando os endereços de memória física do host a endereços de memória virtual do host, a primeira pluralidade de entradas referenciando, em agregado, a mesma segunda faixa de memória que a primeira entrada na tabela de tradução de endereços de segunda camada hierarquicamente organizada, em que as entradas da primeira pluralidade de entradas estão em um nível hierarquicamente mais baixo de tabelas;

em que os endereços de memória física do convidado são

percebidos como sendo endereços para a memória física por processos em execução dentro do ambiente de computação de máquina virtual;

em que os endereços de memória física do host são endereços para a memória física atual de um dispositivo de computação host que hospeda o ambiente de computação da máquina virtual;

em que os endereços de memória virtual do host são endereços para a memória virtual fornecida por um gerenciador de memória em execução no dispositivo de computação host, o gerenciador de memória utilizando a tabela de página hierarquicamente organizada para fornecer a memória virtual; e

em que a memória física do convidado, identificada pelos endereços de memória física do convidado, é apoiada por uma porção da memória virtual do host, identificada por uma porção dos endereços da memória virtual do host.

2. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que compreende ainda:

detectar, após a conclusão do primeiro acesso à memória, que um primeiro subconjunto da primeira pluralidade de entradas na tabela de páginas hierarquicamente organizadas teve dados, que foram originalmente armazenados em endereços de memória física do host correspondentes, subsequentemente retirados de página para um meio de armazenamento não volátil;

invalidar, em resposta à detecção, a primeira entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada; e

gerar, no lugar da primeira entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada, uma segunda pluralidade de entradas na tabela de tradução de endereço de segunda camada hierarquicamente organizada, a segunda plurali-

dade de entradas referenciando, em agregado, pelo menos algumas das mesmas segundas faixa de memória como a primeira entrada, em que as entradas da segunda pluralidade de entradas estão em um nível hierarquicamente mais baixo de tabelas.

3. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que compreende ainda:

montar uma primeira pluralidade de regiões contíguas de tamanho de página pequena da memória física do host em uma única região de tamanho de página grande da memória física do host.

4. Método, de acordo com a reivindicação 3, **caracterizado** pelo fato de que compreende ainda:

gerar uma segunda entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada, a segunda entrada sendo pelo menos um nível hierárquico acima do nível hierárquico mais baixo de tabelas, de modo que uma terceira faixa de memória, identificada pela segunda entrada, pode ser identificada sem referência a qualquer tabela no nível hierarquicamente mais baixo de tabelas, a terceira faixa de memória referenciando a única região de tamanho de página grande da memória física do host na qual a primeira pluralidade das regiões contíguas de tamanho de página pequena do host físico a memória foi montada; e

em resposta à geração da segunda entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada, marcar como usada uma segunda pluralidade de entradas na tabela de página hierarquicamente organizada, a segunda pluralidade de entradas referenciando a primeira pluralidade das pequenas regiões contíguas de tamanho de página da memória física do host, que foram reunidas na única região de tamanho de página grande da memória física do host.

5. Método, de acordo com a reivindicação 4, **caracterizado**

pelo fato de que compreende ainda:

copiar dados de uma segunda pluralidade de regiões de tamanho de página pequena da memória física do host para pelo menos uma porção da primeira pluralidade de regiões de tamanho de página pequena contígua, a segunda pluralidade de regiões de tamanho de página pequena estando pelo menos parte descontínua; e

invalidar uma segunda pluralidade de entradas na tabela de tradução de endereço de segunda camada hierarquicamente organizada que faz referência à segunda pluralidade de regiões de tamanho de página pequena da memória física do host, em que as entradas da segunda pluralidade de entradas estão em um nível hierarquicamente mais baixo de tabelas;

em que a segunda entrada gerada na tabela de tradução de endereço de segunda camada hierarquicamente organizada é utilizada no lugar da segunda pluralidade de entradas que foram invalidadas.

6. Método, de acordo com a reivindicação 4, **caracterizado** pelo fato de que compreende ainda:

copiar dados de um segundo conjunto de uma ou mais regiões de tamanho de página pequena da memória física do host para pelo menos uma porção da primeira pluralidade das regiões contíguas de tamanho de página pequena; e

invalidar uma segunda pluralidade de entradas na tabela de tradução de endereço de segunda camada hierarquicamente organizada, a segunda pluralidade de entradas compreendendo ambos: (1) um primeiro subconjunto de entradas que fazem referência ao segundo conjunto de uma ou mais regiões de tamanho de página pequena da memória física do host e (2) um segundo subconjunto de entradas que fazem referência a pelo menos alguma da primeira pluralidade de regiões contíguas de tamanho de página pequena da memória física do host, em que as entradas da segunda pluralidade de entradas es-

tão em um nível hierarquicamente mais baixo de tabelas;

em que a segunda entrada gerada na tabela de tradução de endereço de segunda camada hierarquicamente organizada é utilizada no lugar da segunda pluralidade de entradas que foram invalidadas.

7. Método, de acordo com a reivindicação 3, **caracterizado** pelo fato de que a montagem da primeira pluralidade de regiões contíguas de tamanho de página pequena da memória física do host compreende a cópia de dados de algumas da primeira pluralidade de regiões contíguas de tamanho de página pequena para outras regiões de tamanho de página pequena da memória física do host que difere da primeira pluralidade de regiões contíguas de tamanho de página pequena.

8. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que compreende ainda:

impedir a paginação da segunda faixa de memória.

9. Mídias de armazenamento legíveis por computador, **caracterizadas** pelo fato de que compreende instruções executáveis por computador, as quais, quando executadas, fazem com que um dispositivo de computação execute as etapas como definidas na reivindicação 1.

10. Dispositivo de computação, **caracterizado** pelo fato de que compreende:

uma ou mais unidades centrais de processamento;

memória de acesso aleatório (RAM); e

uma ou mais mídias legíveis por computador, compreendendo:

um primeiro conjunto de instruções executáveis por computador, as quais, quando executadas pelo dispositivo de computação, fazem com que o dispositivo de computação forneça um gerenciador de memória que faz referência a uma tabela de páginas hierarquica-

mente organizada para traduzir endereços de memória virtual do host em endereços de memória física do host que identificam locais na RAM;

um segundo conjunto de instruções executáveis por computador, que, quando executadas pelo dispositivo de computação, fazem com que o dispositivo de computação forneça um ambiente de computação de máquina virtual, em que os processos em execução dentro do ambiente de computação de máquina virtual percebem endereços de memória física do convidado como sendo endereços para memória física, e em que mais memória física do convidado, identificada pelos endereços de memória física do convidado, é apoiada por uma porção da memória virtual do host, identificada por uma porção dos endereços da memória virtual do host; e

um terceiro conjunto de instruções executáveis por computador, que, quando executadas pelo dispositivo de computação, fazem com que o dispositivo de computação:

detectar um primeiro acesso à memória, a partir de um primeiro processo em execução no ambiente de computação da máquina virtual, direcionado a uma primeira faixa de memória;

gerar, como uma pré-condição para completar o primeiro acesso à memória, uma primeira entrada em uma tabela de tradução de endereço de segunda camada hierarquicamente organizada, a tabela de tradução de endereço de segunda camada hierarquicamente organizada correlacionando os endereços de memória física do host aos endereços de memória física do convidado, a primeira entrada sendo pelo menos um nível hierárquico acima de um nível hierarquicamente mais baixo de tabelas, de modo que uma segunda faixa de memória, identificada pela primeira entrada, possa ser identificada sem referência a qualquer tabela no nível hierarquicamente mais baixo de tabelas, sendo a segunda faixa de memória maior do que a primei-

ra faixa de memória; e

em resposta à geração da primeira entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada, marcar como usada uma primeira pluralidade de entradas na tabela de página hierarquicamente organizada, a primeira pluralidade de entradas referenciando, em agregado, a mesma segunda faixa de memória que a primeira entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada, em que as entradas da primeira pluralidade de entradas estão em um nível hierarquicamente mais baixo de tabelas.

11. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que compreende ainda:

detectar um segundo acesso à memória do primeiro processo em execução no ambiente de computação da máquina virtual, o segundo acesso à memória sendo direcionado para uma terceira faixa de memória que difere da primeira faixa de memória; e

satisfazer o segundo acesso à memória por referência a um segundo subconjunto da primeira pluralidade de entradas na tabela de páginas hierarquicamente organizada;

em que o primeiro acesso à memória foi satisfeito por referência a um primeiro subconjunto da primeira pluralidade de entradas na tabela de páginas hierarquicamente organizada, o primeiro subconjunto compreendendo entradas diferentes, na tabela de páginas hierarquicamente organizadas, do que o segundo subconjunto.

12. Método, de acordo com a reivindicação 2, **caracterizado** pelo fato de que a segunda pluralidade de entradas faz referência a porções da segunda faixa de memória que foram acessadas anteriormente pelo primeiro processo e não foram paginadas.

13. Método, de acordo com a reivindicação 3, **caracterizado** pelo fato de que a montagem ocorre subsequentemente à detecção

do primeiro acesso à memória e antes de gerar a primeira entrada na tabela de tradução de endereço de segunda camada hierarquicamente organizada.

14. Método, de acordo com a reivindicação 7, **caracterizado** pelo fato de que a cópia dos dados de algumas da primeira pluralidade de regiões contíguas de tamanho de página pequena para outras regiões de tamanho de página pequena é realizada apenas se uma fragmentação da primeira pluralidade de regiões contíguas de tamanho de página pequena for abaixo de um limite de fragmentação.

15. Método, de acordo com a reivindicação 8, **caracterizado** pelo fato de que compreende ainda remover a prevenção de paginação da segunda faixa de memória se uma ou mais porções da segunda faixa de memória forem acessadas com menos frequência do que um limite de frequência de acesso.

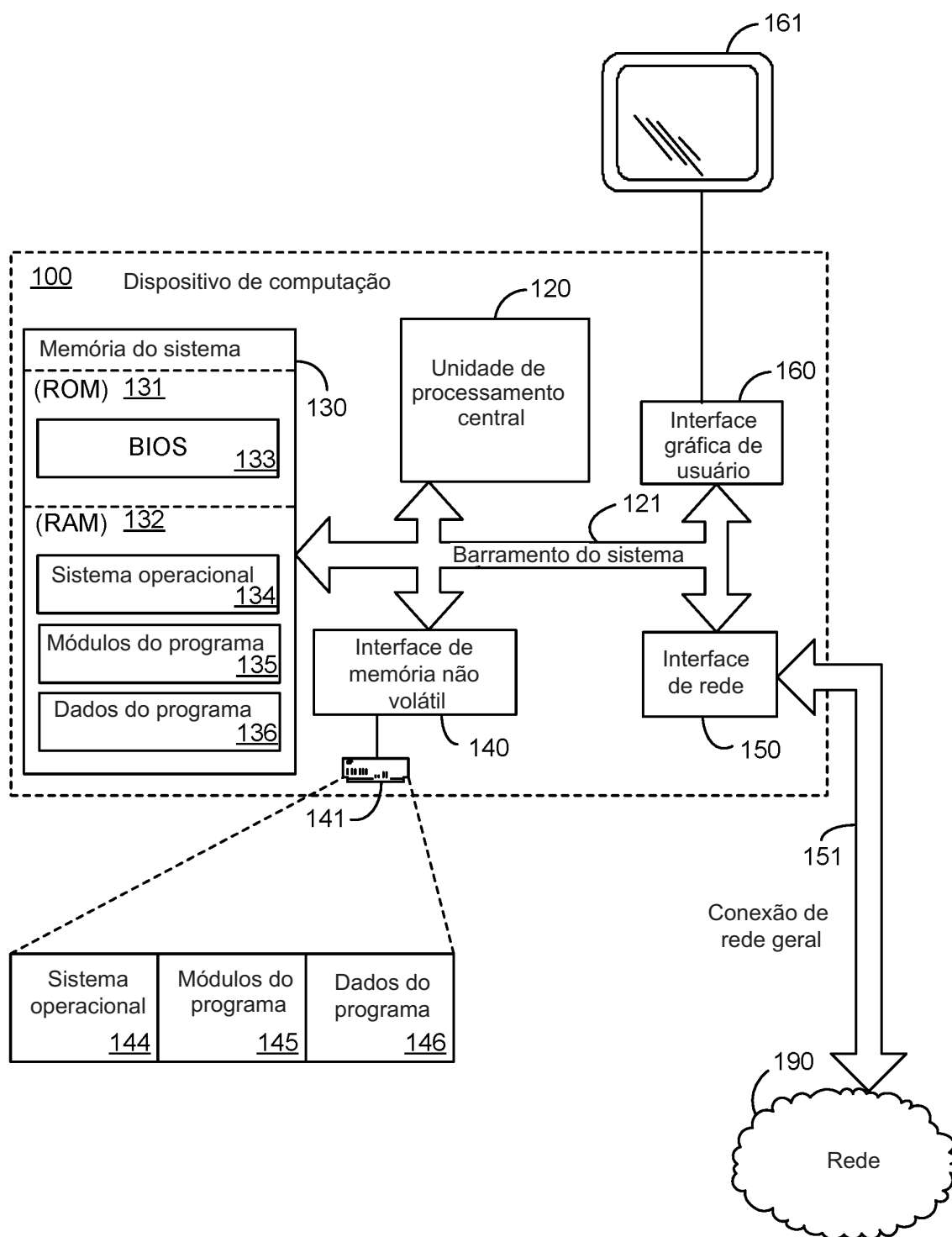


FIG. 1

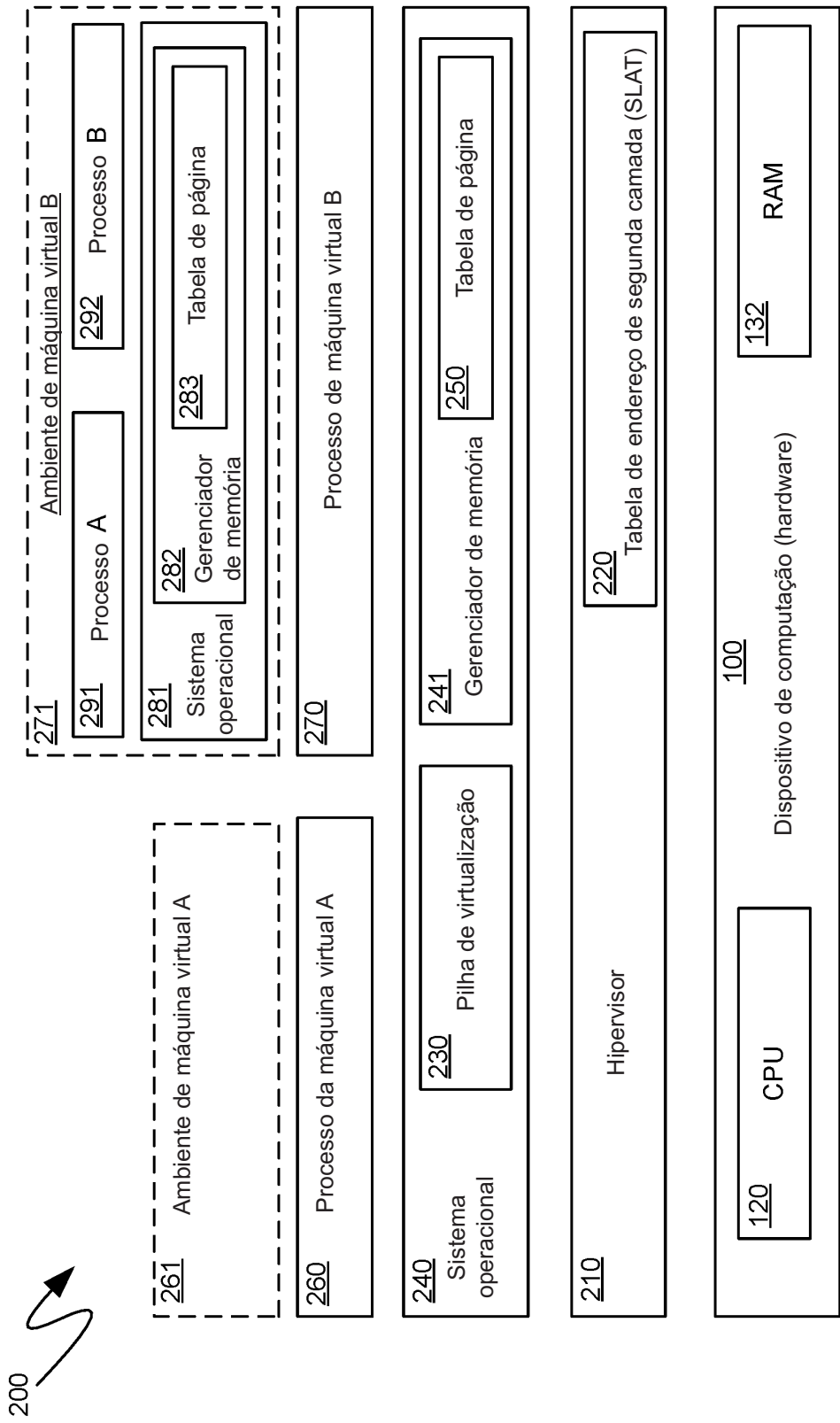


FIG. 2

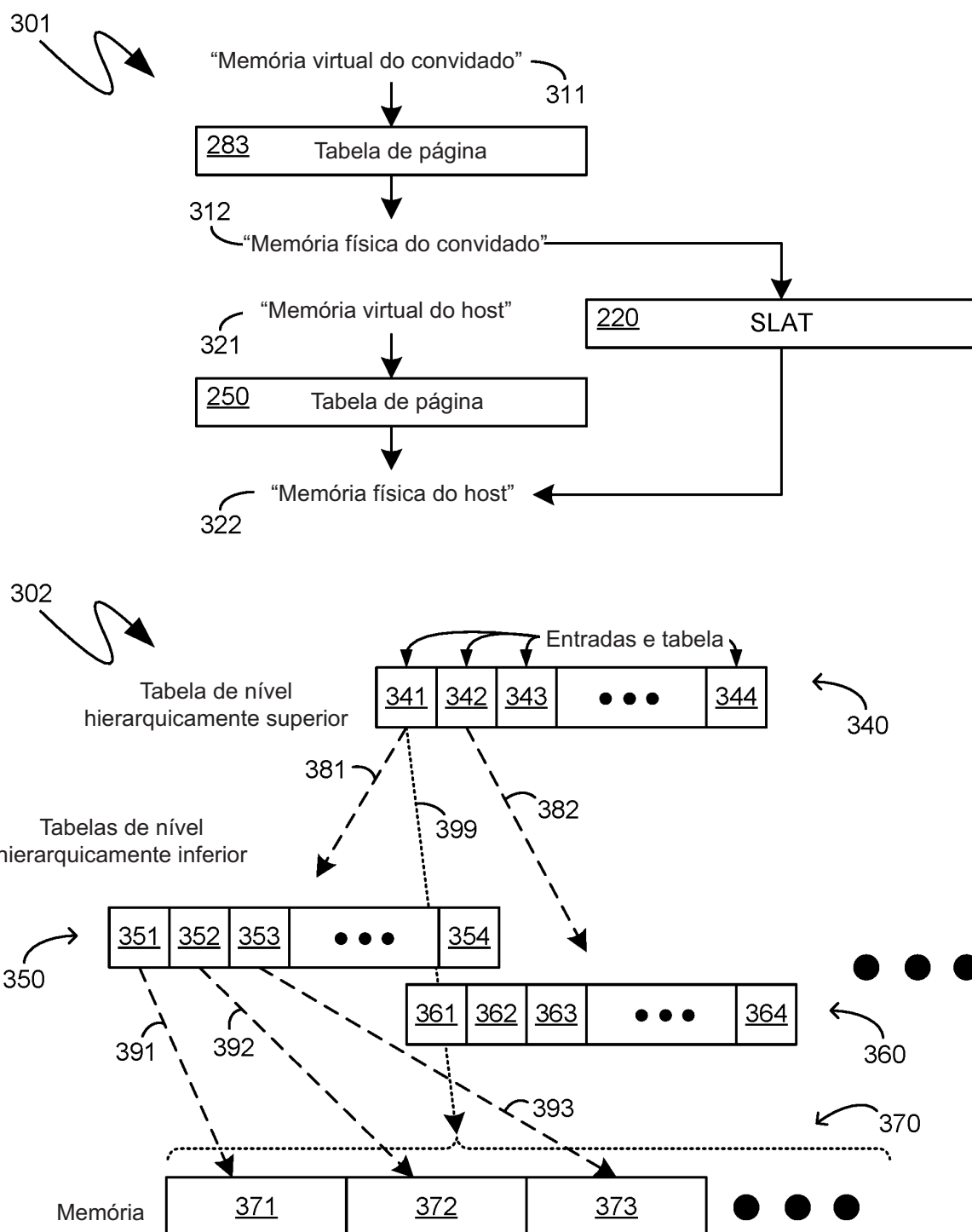


FIG. 3

400

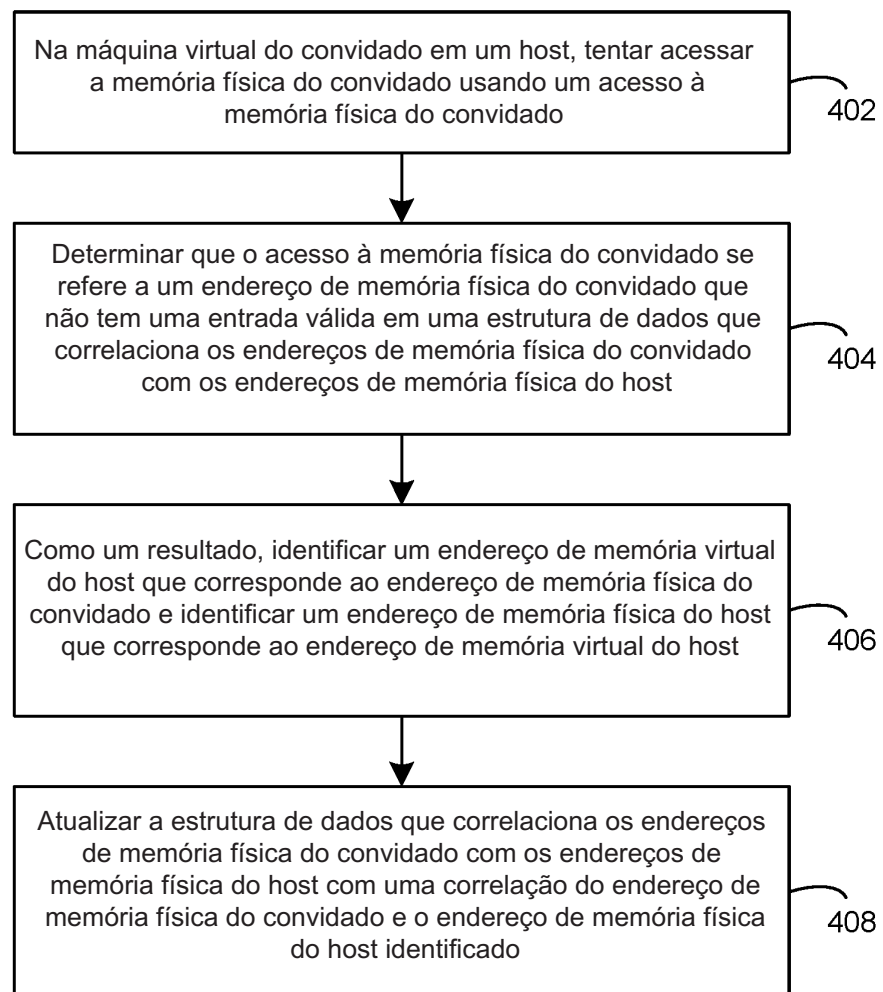


FIG. 4

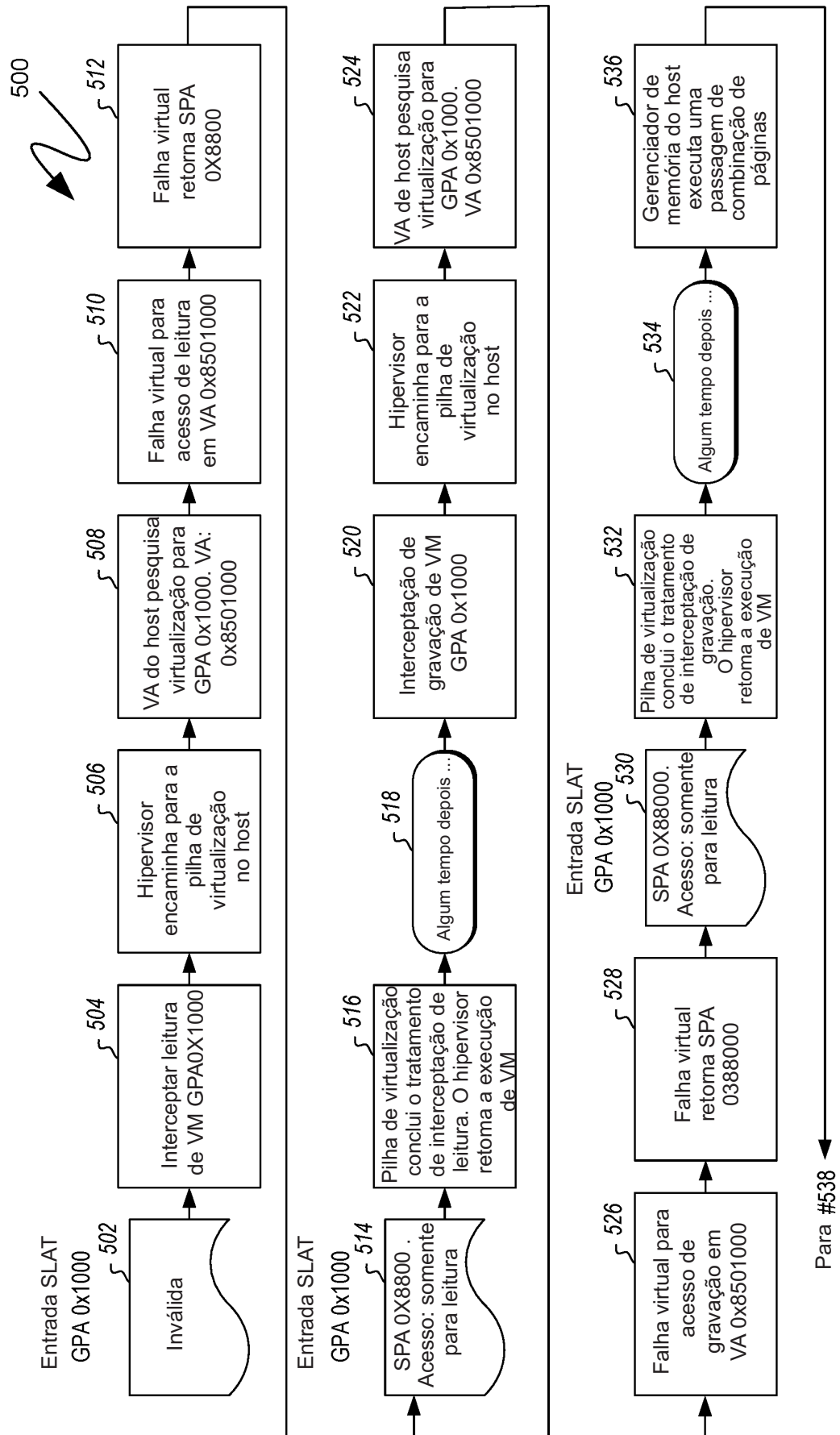


FIG. 5

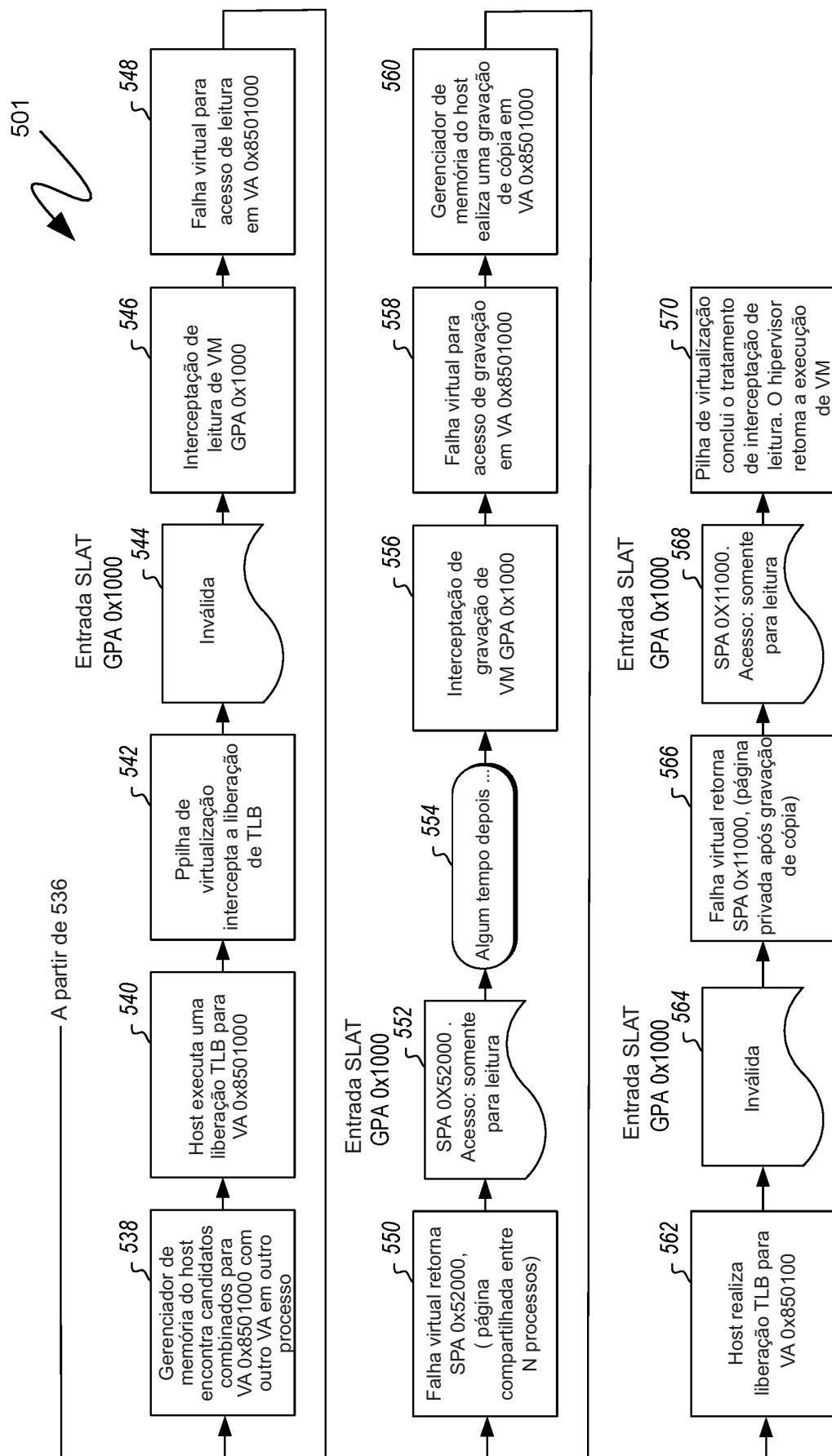


FIG. 5 (continuada)

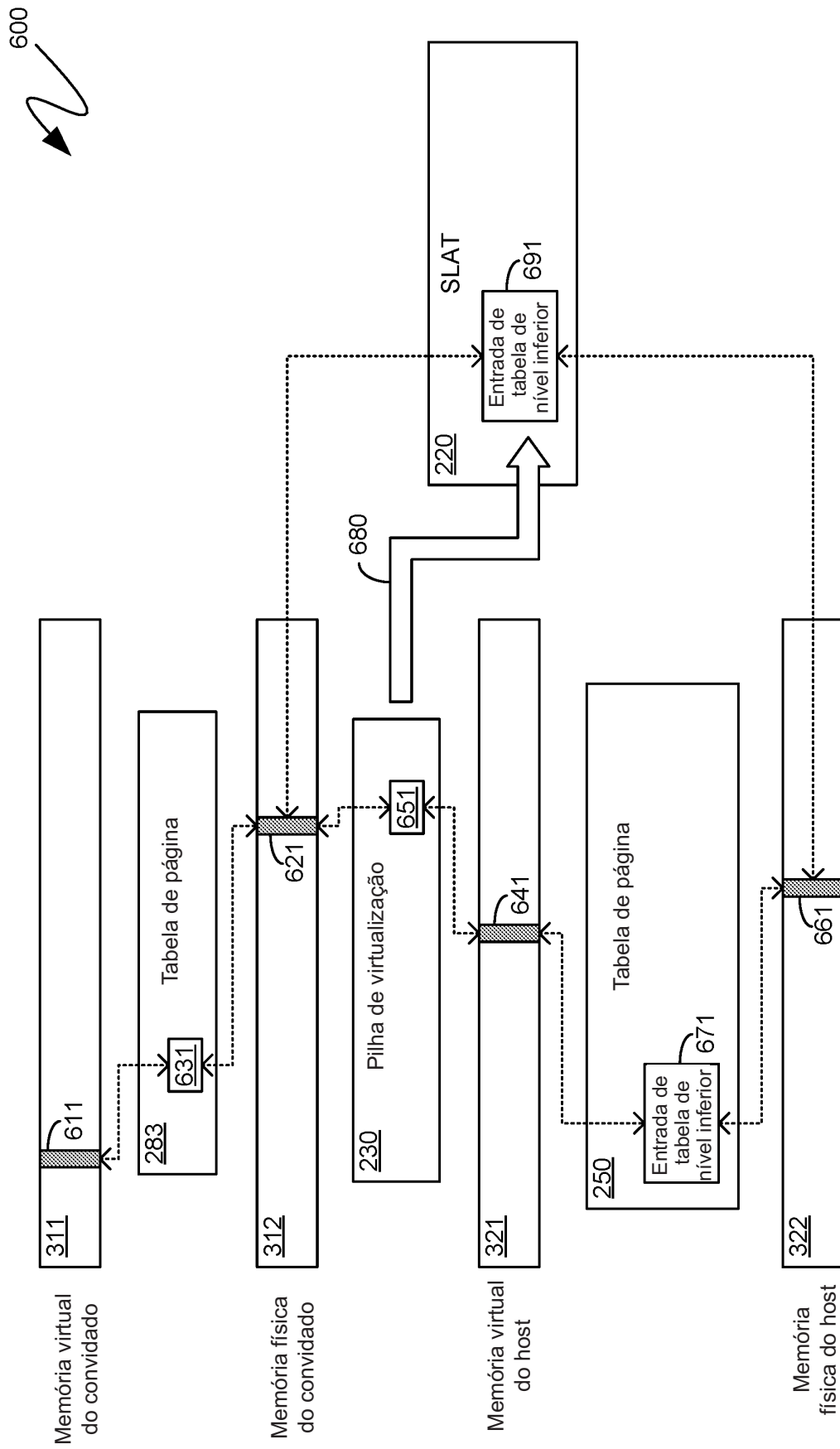


FIG. 6

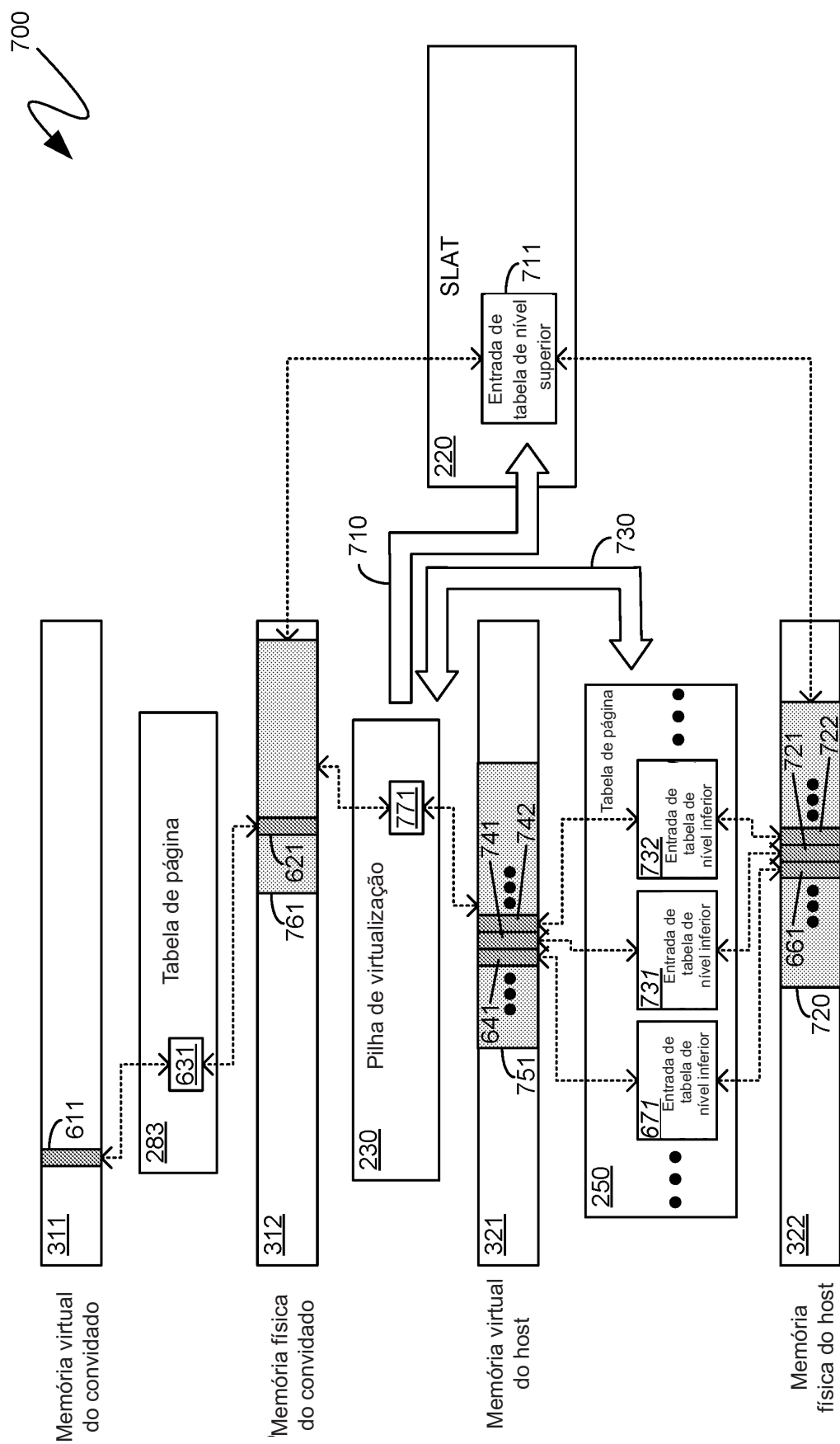


FIG. 7

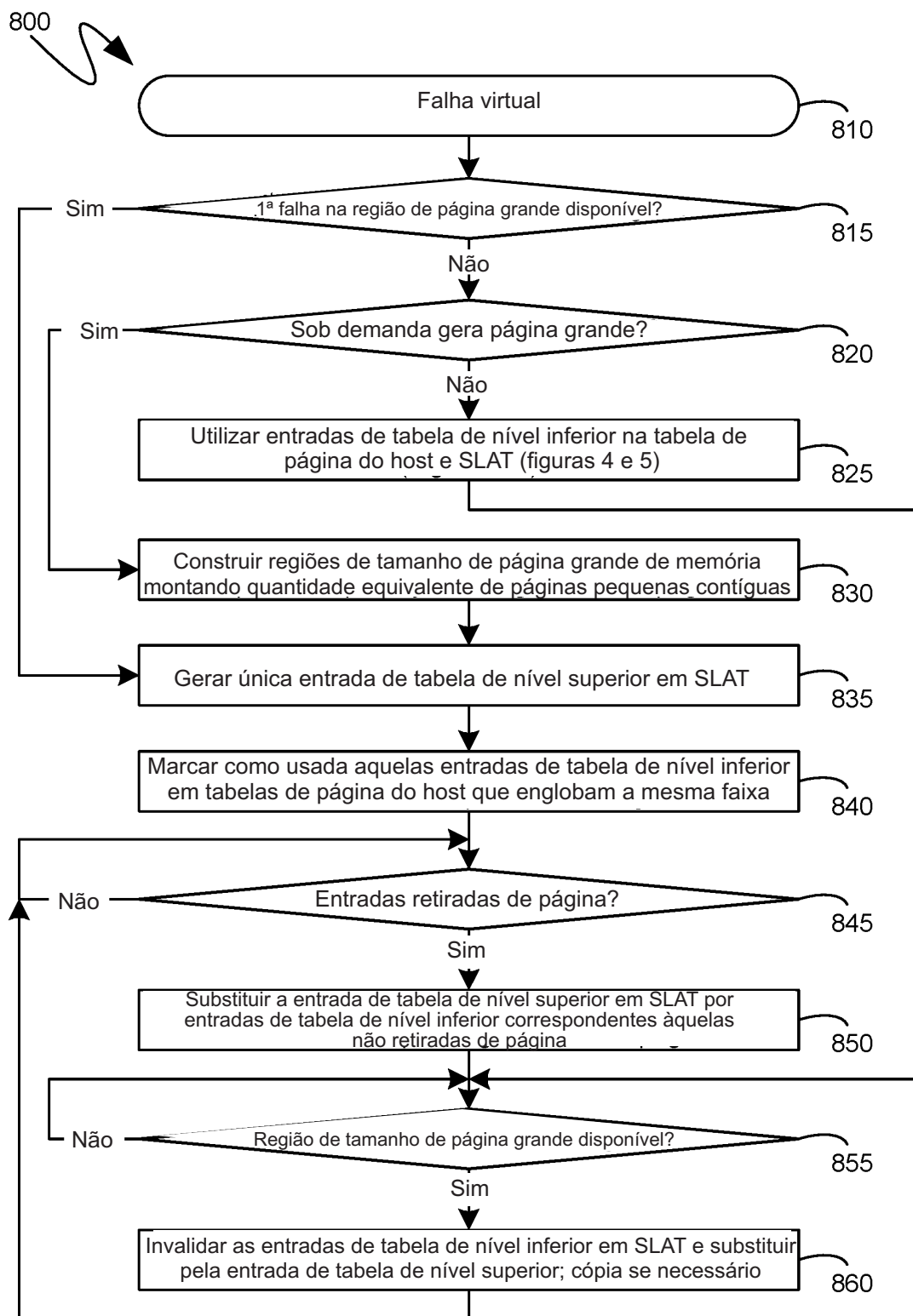


FIG. 8

RESUMO

Patente de Invenção: **"ACESSO MAIS RÁPIDO DA MEMÓRIA DA MÁQUINA VIRTUAL APOIADO POR UMA MEMÓRIA VIRTUAL DO DISPOSITIVO DE COMPUTAÇÃO HOST"**.

Para aumentar a velocidade com que os níveis hierárquicos de uma Tabela de Endereços de Segunda Camada (SLAT) são percorridos como parte de um acesso à memória onde a memória física do convidado de um ambiente de máquina virtual é apoiada por memória virtual atribuída a um ou mais processos em execução em um dispositivo de computação host, um ou mais níveis hierárquicos de tabelas dentro da SLAT podem ser ignorados ou de outra forma não referenciados. Embora a SLAT possa ser preenchida com correlações de memória em níveis hierarquicamente superiores de tabelas, a tabela de páginas do dispositivo de computação host, suportando o fornecimento de memória virtual do dispositivo de computação host, pode manter um conjunto contíguo correspondente de correlações de memória no nível de tabela hierarquicamente mais baixo, permitindo assim que o dispositivo de computação host pague, ou manipule de outra forma, pedaços menores de memória. Se tal manipulação ocorrer, a SLAT pode ser preenchida novamente com correlações de memória no nível de tabela hierarquicamente mais baixo.