



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

(11) PI 0304297-9 B1

(22) Data do Depósito: 30/09/2003

(45) Data de Concessão: 11/10/2016



(54) Título: MÉTODO PARA GERENCIAMENTO PROATIVO DE MEMÓRIA

(51) Int.Cl.: G06F 12/08

(30) Prioridade Unionista: 04/10/2002 US 60/416.115, 20/12/2002 US 10/325.591

(73) Titular(es): MICROSOFT TECHNOLOGY LICENSING, LLC

(72) Inventor(es): STUART SECHREST, MICHAEL R. FORTIN, CENK ERGAN, Inventor(a), MEHMET IYIGUN

"MÉTODO PARA GERENCIAMENTO PROATIVO DE MEMÓRIA"

REMISSÃO RECÍPROCA A PEDIDOS AFINS

[001] A presente invenção reivindica o benefício do pedido provisório de patente Norte-Americana Nº 60/416.115, depositado em 4 de outubro de 2002 e incorporado no presente documento à guisa de referência.

CAMPO DA INVENÇÃO

[002] A invenção refere-se, de um modo geral, ao gerenciamento de memória de computador.

ANTECEDENTES DA INVENÇÃO

[003] Os sistemas contemporâneos de computação são equipados com quantidades significativas de memória de acesso aleatório, ou RAM, atualmente sob a ordem de 256 megabytes até alguns gigabytes. Entretanto, os métodos e mecanismos de gerenciamento de memória atuais foram realizados há muitos anos atrás, quando a RAM era uma fonte de computação relativamente escassa. Para proporcionar grandes quantidades de memória virtual a processos de programas, tal mecanismo de gerenciamento de memória depende de técnicas, tais como paginação sob demanda, que transferem essencialmente páginas de dados a partir da RAM para uma unidade de disco quando uma certa quantidade de espaço da RAM fosse necessária por outro processo e se aqueles dados transferidos ainda fossem novamente necessários por seu processo correspondente, lendo os dados de volta a partir do disco para a RAM.

[004] Com tal gerenciamento de memória convencional, as páginas necessárias ficam frequentemente ausentes da memória, mesmo que a quantidade total de código e dados que é mencionada em padrões de uso típico seja razoavelmente estável com relação ao tempo e seja pequena com relação ao dimensionamento de memória atual. Isto é essencialmente um resultado de dois fatores, isto é, que o sis-

tema sofre transições que podem forçar as páginas necessárias a partir da memória e/ou mudam rapidamente o conjunto de páginas necessárias, e erosão, em que páginas que não estão em uso ativo são sobrepostos por outros dados.

[005] Os exemplos de tais transições incluem inicialização a partir de um estado desligado, que mudam de um usuário para outro, retornam de um modo de inatividade (*standby*) ou de hibernação, utilizam uma aplicativo consumidor de memória (*memory hungry application*), (por exemplo, jogar um jogo e então prossegue de volta ao sistema de uso típico, e retorna ao computador após um longo retardo. É desnecessário dizer que, além de operação rápida normal, os clientes desejam muito sistema e uma partida rápida no programa que sigam tais transições. Entretanto, com o gerenciamento de memória atual, isto não é possível no momento. Por exemplo, testes mostraram que, após um retorno da hibernação, um disparo de aplicativo pode levar de nove a onze segundos, mesmo que grandes quantidades de memória não sejam utilizadas neste momento. Isto deve-se ao fato de que a maior parte do tempo de disparo é consumida em transferências de disco, com a CPU bloqueada, que aguarda o código necessário e páginas de dados a serem lidos a partir do disco, que como resultado de paginação de demanda é essencialmente realizada uma página por vez. Em suma, após tais transições, o conjunto de páginas de maior utilidade não estão frequentemente em memória, que resultam em paginação de demanda que proporciona uma experiência desagradável ao usuário devido à lentidão de tempos de transferência de entrada/saída(I/O).

[006] Embora algumas transições envolvam evicção deliberada de páginas de memória, e outras resultam em demanda repentina para memória, ainda outras resultam em páginas que tornam-se inativas e são trocadas por discos. Mais particularmente, as páginas de memória que estão em conjuntos de trabalho podem ser aparadas e colocadas

dentro de um cache de páginas que não estão em uso, referidas no presente documento como um conjunto de páginas em espera (por exemplo, uma lista inativa em um sistema baseado na plataforma Windows®), juntamente com as páginas de arquivos que estão fechados. Para reduzir o número de discos de leituras e gravações, o gerenciamento de memória monitora quais as páginas de memória no conjunto de páginas inativas foram menos acessadas recentemente, e quando a memória é necessária, seleciona-se aquelas páginas para transferência para disco, essencialmente em tipos de fila primeiro a entrar, primeiro a sair (FIFO).

[007] Embora esta técnica trabalhe na medida em que reduz-se em transferências de disco de redução, porque isto resulta em páginas recentemente mais usadas pelos processos que são aqueles que permanecem em memória, aplicativos modernos possuem código e requerimentos de dados amplos. Como resultado, em curto prazo, as páginas podem ser chamadas de volta a partir deste conjunto de páginas inativas, entretanto, por um prazo mais longo, com gerenciamento de memória convencional, a fila simples (FIFO) do conjunto de página inativa resulta em erosão, na qual páginas possivelmente valiosas (porém não utilizadas recentemente) são sobrepostas por páginas menos valiosas (porém mais utilizadas recentemente). Consequentemente, é necessário um número significativo de transferências I/O até e a partir dos discos. Mesmo em sistemas potentes estas transferências de I/O atuam como um estágio de estrangulamento, onde o sistema subutiliza grandes quantidades de memória disponível, deixando a CPU inativa por grande quantidade de tempo.

[008] Em suma, os programas modernos requerem grande quantidade de memória de acesso aleatório e rápidas capacidades de processamento, que os dispositivos de computação modernos oferecem. Entretanto, a tecnologia atual de gerenciamento de memória não cor-

responde-se de forma adequada a esses recursos conjuntamente, pois o gerenciamento de memória atual foi desenvolvido em um momento em que a memória estava escassa, e foi então projetado com a visão geral de que o uso de disco substancial era inevitável, ainda que não incômodo devido ao fato de esses aplicativos serem pequenos e a realização de multitarefas ser rara. Como tal, o gerenciamento de memória convencional não tenta limitar o uso de disco, além de manter em memória as páginas recentemente mais acessadas. Em computação moderna, isto resulta em uma quantidade de significativa de uso de disco, e desse modo proporciona um estágio de estrangulamento que prejudica tanto o desempenho real como o percebido.

SUMÁRIO DA INVENÇÃO

[009] Resumidamente, a presente invenção é dirigida a uma arquitetura de gerenciamento de memória aperfeiçoada, que compreende sistemas, métodos e mecanismos que proporcionam um sistema de gerenciamento de memória proativo, resiliente e de auto-sintonização. O sistema de gerenciamento de memória funciona para proporcionar melhoras de desempenho real e percebido em gerenciamento de memória, ao carregar e manter em memória dados que são provavelmente necessários, antes dos dados serem realmente necessários. As páginas de dados podem ser priorizadas com um valor/pontuação de acordo com vários fatores, que incluem suas utilidades, frequências de uso, tempo de último uso, facilidade de transferência de dados e outra, informação baseada no contexto. Os mecanismos funcionam para pré-buscar e/ou manter os dados mais valiosos em memória. Se os dados precisarem ser trocados ou sobrepostos, os dados podem ser automaticamente trazidos de volta de uma maneira resiliente, não devido à demanda real, porém devido à demanda esperada.

[0010] Ao preencher a memória com dados apropriados antes daquelas páginas serem necessárias, o sistema de gerenciamento de

memória reduz ou elimina substancialmente as operações de transferência de disco sob demanda, e então reduz ou elimina os estágios de estrangulamentos I/O em muitos cenários de cliente significativos. Para esse fim, as implementações da presente invenção compreendem diversos mecanismos dirigidos ao monitoramento do histórico de uso de memória, análise de uso de memória, renovar a memória com páginas altamente valiosas (por exemplo, altamente utilizadas), eficiência de pré-busca, e gerenciamento de disco ativo.

[0011] Para obter a informação necessária para designar um valor para uma página, um rico conjunto de informações sobre cada página é monitorado através de técnicas de rastreamento e registro, de preferência até um período de tempo relativamente longo. Os dados avaliados são processados transformando-se em informações gerenciáveis, que são mantidas e acessadas como parte de um processo de tomada de decisão que reequilibra os conteúdos de memória em resposta aos eventos relacionados com a memória e/ou sobre uma base temporizada de forma que os dados mais valiosos sejam mantidos e/ou pré-buscados dentro da memória com relação aos dados menos valiosos. O valor de prioridade permite que um conjunto de página inativa (por exemplo, uma lista inativa em um sistema baseado na plataforma Windows®), de páginas de memória seja priorizado, tal que as páginas no conjunto de página inativa possam ser de maior prioridade em termos de valor com relação a outras páginas, independente de quando referenciadas pela última vez, e desta maneira mantidas em memória com relação a outras páginas. A pré-busca resiliente e proativa carrega páginas valiosas oriundas do disco para o conjunto de página inativa, em uma ordem determinada pela prioridade. A carga é de preferência realizada em uma baixa prioridade, transferência em segundo plano I/O para impedir que interfira em uma operação normal do usuário, embora esta também possa ser realizada em uma operação em

primeiro plano.

[0012] Como resultado da presente invenção, a utilização da CPU é significativamente aumentada sem interferir no uso normal de sistema. Para muitos usuários de computador, todo o conjunto de páginas de memória realmente usado é pequeno o bastante para manter em memória, sendo que resulta em inúmeras falhas de demanda (para ler os dados necessários do disco) reduzidas a zero ou quase zero.

[0013] Em uma implementação, um mecanismo de rastreamento observa atividade relacionada à memória, e uma parte registrante efetua o registro dos resultados das observações. Um integrador combina a parte registrante que resulta com os prévios registros dentro de um base de dados de informação de página. Um serviço de extração acessa os dados no base de dados de informação de página para desenvolver planos de cenário a partir do mesmo, e os planos de cenário são analisados para priorizar as páginas dentro de uma lista. A lista é então classificada de acordo com o valor calculado para cada página, que foi baseado nas observações de uso de memória e possivelmente outros fatores.

[0014] Uma parte de reequilíbrio, disparada por um evento (por exemplo, sistema ou tempo gerado) trabalha com um planejador I/O para adicionar páginas em um conjunto de página inativa priorizado, em uma ordem determinada por cada valor calculado de página. O conjunto de página inativa priorizado inclui inúmeros subconjuntos que correspondem aos valores para cada página. Estes valores podem mudar com o tempo e/ou baseados no uso real, e desta maneira as páginas podem mover-se de forma lógica entre os subconjuntos. À medida em que necessita-se de memória, se sob demanda ou para pré-busca em segundo plano, os dados de prioridade mais baixa no subconjunto são paginados primeiro. Desta maneira, as páginas mais valiosas, conforme determinado por cálculos e observações de uso

atual, bem como contexto e outras informações, são colocadas automaticamente em memória em uma maneira de não-interferência, e tende a permanecer nesta com relação às páginas menos valiosas.

[0015] Outras vantagens tornar-se-ão óbvias a partir da seguinte descrição detalhada quando tomadas em conjunto com os desenhos, nos quais:

BREVE DESCRIÇÃO DOS DESENHOS

[0016] A Figura 1 é um diagrama de blocos que representa, de um modo geral um sistema de computador dentro do qual a presente invenção pode ser incorporada;

As Figuras 2 e 3 compreendem diagramas de blocos que representam uma relação básica entre os componentes do sistema de acordo com um aspecto da presente invenção;

As Figuras 4A e 4B compreendem um diagrama de blocos que representa uma arquitetura geral adequada para realizar gerenciamento proativo de memória de acordo com um aspecto da presente invenção;

A Figura 5A é um diagrama de blocos que mostra uma maneira na qual um conjunto de página inativa pode manter dados de página;

A Figura 5B é um diagrama de blocos que representa uma maneira alternativa na qual um conjunto de página inativa priorizado pode manter dados de página em subconjuntos priorizados, de acordo com um aspecto da presente invenção; e

A Figura 6 é um diagrama de blocos que representa um sistema de restrição para compelir atividades em segundo plano para limitar interferência com aplicativos do usuário, de acordo com um aspecto da presente invenção.

DESCRIÇÃO DETALHADA

AMBIENTE DE OPERAÇÃO EXEMPLIFICATIVO

[0017] A Figura 1 ilustra um exemplo de um ambiente de sistema de computação adequado 100 sobre o qual a invenção pode ser im-

plementada. O ambiente de sistema de computação 100 é somente um exemplo de um ambiente de computação adequado e não pretende sugerir qualquer limitação para o escopo de uso ou funcionalidade da invenção. O ambiente de computação 100 não deve ser interpretado como se tivesse qualquer dependência ou requerimento com relação a qualquer um nem combinação de componentes ilustrados no ambiente de operação exemplificativo 100.

[0018] A invenção é operacional com inúmeros outros propósitos gerais ou configurações ou ambientes de sistema de computação com propósito especial. Exemplos de ambientes, sistemas de computação bem conhecidos, e/ou configurações que podem ser adequados para uso com a invenção incluem, porém sem caráter limitativo: computadores pessoais, computadores de servidor, dispositivos de mão ou portáteis, dispositivos de mesa gráfica, sistemas de multiprocessador, sistemas baseados em microprocessador, aparelhos decodificadores, eletrônica de consumo programável, PCs de rede, microcomputadores, computadores de quadro principal, ambientes de computação distribuídos que incluam qualquer um dos sistemas ou dispositivos acima, e similares

[0019] A invenção pode ser descrita no contexto geral das instruções executáveis por computador, tais como módulos de programa, que são executados por um computador. Geralmente, os módulos de programa incluem rotinas, programas, objetos, componentes, estruturas de dados, e assim por diante, que realizam tarefas particulares ou implementam tipos de dados abstratos. A invenção pode ser também exercida em ambientes de computação distribuídos onde as tarefas são realizadas por dispositivos de processamento remoto que são conectados através de uma rede de comunicações. Em um ambiente de computação distribuído, os módulos de programa podem ser localizados em meios de armazenamento de computador local e/ou remoto

que incluem dispositivos de armazenamento de memória.

[0020] Com referência à Figura 1, um sistema exemplificativo para implementar a invenção inclui um dispositivo de computação de propósito geral na forma de um computador 110. Os componentes do computador 110 podem incluir, porém não limitados a, uma unidade de processamento 120, uma memória de sistema 130, e um barramento de sistema 121 que acopla vários componentes de sistema que incluem a memória de sistema para a unidade de processamento 120. O barramento de sistema 121 pode ser qualquer um dos diversos tipos de estruturas de barramento que inclua um barramento de memória ou controlador de memória, um barramento periférico, e um barramento local que utilize uma variedade de arquiteturas de barramento. À guisa de exemplo, e sem caráter limitativo, tais arquiteturas incluem barramento (ISA) Industry Standard Architecture, barramento (MCA) Micro Channel Architecture, barramento (EISA) Enhanced ISA, barramento local (VESA) Video Electronics Standards Association, e barramento (PCI) Peripheral Component Interconnect porém também conhecido como barramento Mezzanine.

[0021] O computador 110 inclui tipicamente uma variedade de meios legíveis por computador. O meio legível por computador pode ser qualquer meio disponível que possa ser acessado pelo computador 110 e inclui tanto meio volátil como não-volátil e meio removível e não-removível. À guisa de exemplo, e sem caráter limitativo, o meio legível por computador pode compreender meio de armazenamento de computador e meio de comunicação. O meio de armazenamento de computador inclui meio volátil e não-volátil, removível e não-removível implementado em qualquer método ou tecnologia para armazenamento de informações tais como instruções legíveis por computador, estruturas de dados, módulos de programa ou outros dados. O meio de armazenamento de computador inclui, porém não limitado a, RAM,

ROM, EEPROM, memória instantânea ou outra tecnologia de memória, CD-ROM, discos versáteis digitais (DVD) ou outro armazenamento de disco óptico, cassetes magnéticos, fitas magnéticas, armazenamento de disco magnético ou outros dispositivos de armazenamento magnético, ou qualquer outro meio que possa ser usado para armazenar as informações desejadas e que possa ser acessado pelo computador 110. Os meios de comunicação tipicamente concretizam instruções legíveis por computador, estruturas de dados, módulos de programa ou outros dados em um sinal de dados modulados tal como onda portadora ou outro mecanismo de transporte e incluem qualquer meio de distribuição de informações. O termo "sinal de dados modulados" significa um sinal que possui um ou mais conjuntos de características do mesmo ou alterados de tal maneira para codificar informações no sinal. À guisa de exemplo, e sem caráter limitativo, os meios de comunicação incluem meios cabeados tais como rede com fio ou conexão direta com fio, e meios sem fio tais como acústico, RF, infravermelho e outro meios sem fio. As combinações de qualquer um destes acima deve estar também incluído dentro do escopo do meio legível por computador.

[0022] A memória de sistema 130 inclui meio de armazenamento de computador na forma de memória volátil e/ou não-volátil tal como memória somente de 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 informações entre os elementos dentro do computador 110, tal como durante a inicialização, é tipicamente armazenado em ROM 131. A RAM 132 contém tipicamente dados e/ou módulos de programa que são imediatamente acessíveis e/ou atualmente operadas através da unidade de processamento 120. À guisa de exemplo, e sem caráter limitativo, a Figura 1 ilustra um sistema operacional 134, programas aplicativos 135, outros módulos

de programa 136 e dados de programa 137.

[0023] O computador 110 pode também incluir outros meios de armazenamento de computador removíveis/não removíveis, voláteis/não voláteis. Somente à guisa de exemplo, A Figura 1 ilustra uma unidade de disco rígido 141 que lê a partir de ou grava para um meio magnético não-removível, não-volátil, uma unidade disco magnético 151 que lê a partir de ou grava para um disco magnético removível, não-volátil 152, e uma unidade de disco ótico 155 que lê a partir de ou grava para um disco ótico removível, não-volátil 156 tal como um CD ROM ou outro meio ótico. Outros meios de armazenamento de computador removíveis/não-removíveis, voláteis/não-voláteis que podem ser utilizados no ambiente de operação exemplificativo incluem, porém não limitados a, cassetes de fita magnética, placas de memória instantânea, discos versáteis digitais, fita de vídeo digital, RAM de estado sólido, ROM de estado sólido, e similares. A unidade de disco rígido 141 é tipicamente conectada ao barramento de sistema 121 através de uma interface de memória removível tal como interface 140, e unidade de disco magnético 151 e unidade de disco ótico 155 são tipicamente conectadas ao barramento de sistema 121 através de uma interface de memória removível, tal como interface 150.

[0024] As unidades e seus meios de armazenamento de computador associados, discutidos acima e ilustrados na Figura 1, proporcionam armazenamento de instruções legíveis por computador, estruturas de dados, módulos de programa e outros dados para o computador 110. Na Figura 1, por exemplo, a unidade de disco rígido 141 é ilustrada como um sistema operacional de armazenamento 144, programas aplicativos 145, outros módulos de programa 146 e dados de programa 147. Nota-se que estes componentes tanto podem ser iguais como diferentes do sistema operacional 134, programas aplicativos 135, outros módulos de programa 136, e dados de programa 137. O

sistema operacional 144, programas aplicativos 145, outros módulos de programa 146, e dados de programa 147 são dados diferentes números no presente documento para ilustrar que, em um mínimo, eles são cópias diferentes. Um usuário pode ingressar comandos e informações no computador 110 através de dispositivos de entrada tais como uma mesa gráfica, ou digitador eletrônico 164, um microfone 163, um teclado 162 e um dispositivo de indicação 161, comumente referido como mouse, cursor trackball, mesa sensível ao toque. Outros dispositivos de entrada não mostrados na Figura 1 podem incluir um punho de comando (joystick), sensor de jogo, prato de satélite, dispositivo de varredura ou similares. Estes e outros dispositivos são frequentemente conectados à unidade de processamento 120 através de uma interface de entrada com o usuário 160 que é acoplada ao barramento de sistema, porém pode ser conectada através de outra interface e estruturas de barramento, tais como porta paralela, porta de jogos ou um barramento serial universal (USB). Um monitor 191 ou outro tipo de dispositivo de vídeo é também conectado ao sistema de barramento 12 através de uma interface, tal como interface de vídeo 190. O monitor 191 pode ser também integrado com um painel de tela sensível ao toque ou similar. Nota-se que o monitor e/ou painel de tela sensível ao toque pode ser fisicamente acoplado em um alojamento no qual o dispositivo de computação 110 é incorporado, tal como em um computador pessoal do tipo mesa gráfica. Ademais, computadores tal como o dispositivo de computação 110 pode também incluir outros dispositivos de saída periféricos tais como auto falantes 195 e impressora 196, que podem ser conectados através de uma interface periférica de saída 194 ou similar.

[0025] O computador 110 pode operar em um ambiente de rede que utiliza conexões lógicas para um ou mais computadores remotos, tal como um computador remoto 180. O computador remoto 180 pode

ser um computador pessoal, um servidor, um roteador, um PC de rede, um dispositivo de rede não hierárquica ou outro nó de rede comum, e inclui tipicamente muitos ou todos os elementos descritos acima com relação ao computador 110, embora somente um dispositivo de armazenamento de memória 181 seja ilustrado na Figura 1. As conexões lógicas mostradas na Figura 1 incluem uma rede de área local (LAN) 171 e uma rede remota (WAN) 173, porém pode incluir também outras redes. Tais ambientes de rede são comuns em escritórios, redes de computador empresarial corporativa, intranet e a Internet. Por exemplo, na presente invenção, o sistema de computador 110 pode compreender máquina-fonte a partir da qual dados são emigrados, e o computador remoto 180 pode compreender a máquina de destino. Nota-se entretanto que as máquinas fonte e de destino não precisam ser conectadas através de uma rede ou qualquer outro meio, porém em vez disso, os dados podem ser emigrados através de qualquer meio capaz de ser gravado pela plataforma-fonte e lido pela plataforma ou plataformas de destino.

[0026] Quando usado em um ambiente de rede LAN, o computador 110 conecta-se ao LAN 171 através de uma interface de rede ou adaptador 170. Quando usado em um ambiente de rede WAN, o computador 110 inclui tipicamente um modem 172 ou outro meio para estabelecer comunicações com relação à WAN 173, tal como a Internet. O modem 172, que pode ser interno ou externo, pode ser conectado ao barramento de sistema 121 através da interface de entrada com o usuário 160 ou outro mecanismo apropriado. Em um ambiente de rede, os módulos de programa mostrados com relação ao computador 110, ou partes dos mesmos, podem ser armazenados no dispositivo de armazenamento de memória remota. À guisa de exemplo, e sem caráter limitativo, A Figura 1 ilustra programas aplicativos remotos 185 à medida em que estão presentes no dispositivo de memória 181. Será

avaliado que as conexões de rede mostradas são exemplificativas e outros meios de estabelecer um link de comunicação entre os computadores podem ser utilizados.

GERENCIAMENTO PROATIVO DE MEMÓRIA

[0027] Um aspecto da presente invenção é, em parte, geralmente dirigido em direção a uma arquitetura de gerenciamento de memória melhorada que compreende sistemas, métodos e mecanismos que resultam em melhora de desempenho atual e percebido através de aspectos significativos de interface com usuário, programas aplicativos, e software de sistema operacional. Para essa finalidade, como representado na Figura 2, a presente invenção proporciona um sistema de gerenciamento de memória proativa, resiliente e de auto-sintonização 200 que monitora as atividades e os trabalhos relacionadas de memória de um usuário para assegurar, pelo menos a um nível relativamente alto de satisfação, que os dados necessários estejam presentes em memória de acesso aleatório 202 (ou simplesmente "memória" como usado no presente documento) antes dos dados serem necessários, em vez de lidos dentro da memória 202 mediante a demanda de um disco 204 (ou outro, mecanismo de armazenamento secundário equivalente, tais como uma memória instantânea, uma fonte de dados de rede, uma unidade de fita e/ou virtualmente qualquer tipo de armazenamento de dados). Em outras palavras, o gerenciamento de memória proativa e resiliente traz potencialmente dados úteis para dentro da memória e tenta manter estes dados em memória. Se os dados úteis precisarem ser trocados ou sobrepostos, os dados podem ser automaticamente trazidos de volta de maneira resiliente, não devido à demanda atual, porém de preferência devido à demanda esperada.

[0028] Para essa finalidade, a presente invenção impulsiona grandes quantidades de memória atualmente disponíveis para manter em

memória os conjuntos de trabalho de processos potencialmente ativos, e manter a conservação de registro, estruturas de dados, e algoritmos que o sistema utiliza para executar o gerenciamento de memória proativa e resiliente. Espera-se que tal conservação de registro, estruturas de dados, e outros algoritmos torne-se mais e mais favorável com o tempo, entretanto as implementações atuais resultaram em reduções significativas em transferências I/O sob demanda.

[0029] Como pode ser facilmente avaliado, ao ter a memória 202 preenchida com dados apropriados (por exemplo, dispostos em páginas) antes destas páginas serem necessárias, o sistema de gerenciamento de memória 200 da presente invenção reduz ou elimina substancialmente operações de transferência de disco mediante demanda, e desta maneira reduz ou elimina estágios de estrangulamento I/O em muitos cenários de cliente significativo. Nota-se que conforme usado no presente documento, o termo "página" pode referir-se à menor quantidade de dados que o sistema de gerenciamento de memória 200 manuseia como uma unidade, (por exemplo, 4096 bytes), entretanto não há intenção de limitar a presente invenção a qualquer quantidade fixa particular de dados, e assim a "página" de dados pode ser alguma outra quantidade fixa, ou ainda pode ser uma quantidade variável de dados (por exemplo, tão pequena como um byte ou múltiplo do mesmo), ou até transferências de fluxo. Desta maneira, um sistema de gerenciamento de memória que pode ler /gravar/ quantidades arbitrárias de fluxo de dados, em vez de uma quantidade de forma fixa ou múltipla do mesmo, por exemplo, continua equivalente à leitura e gravação de uma ou mais páginas.

[0030] Conforme descrito abaixo, as implementações da presente invenção compreendem diversos mecanismos, que incluem aqueles dirigidos em direção a monitoramento de histórico uso de memória, análise de uso de memória, de renovação com páginas altamente vali-

osas (por exemplo, altamente utilizadas), gerenciamento de disco eficiente, e ativo. Em geral, estes mecanismos trabalham juntos como o sistema de gerenciamento de memória proativa 200, para assegurar que os dados necessários estarão em memória antes de serem requeridos. Para esse fim, algoritmos, que sustentam estruturas de dados, e métodos estendem gerenciamento de memória para operar de forma proativa, ao tirar vantagem de vários tipos de dados de uso de memória e outras informações que podem ser adquiridas continuamente e/ou por um quadro de tempo relativamente longo. Nota-se, entretanto, que a presente invenção não requer qualquer combinação de tais mecanismos, porém pode proporcionar inúmeros benefícios e melhor relacionados com a memória com poucos e/ou mecanismos alternativos. Por exemplo, ao determinar um valor indicativo de que as páginas são mais prováveis de serem usadas que outras, e deixa-se as páginas mais valiosas em memória, as reduções em transferências I/O podem ser obtidas sem qualquer pré-busca em segundo plano de páginas em memória, e mesmo que as informações usadas para determinar o valor daquelas páginas sejam adquiridas intermitentemente e/ou somente com relação a um período de tempo relativamente curto.

[0031] Ademais, métodos e mecanismos de gerenciamento de memória da presente invenção podem melhorar o uso de memória ao serem ativos não só em períodos quando a memória está cheia e troca de decisões precisam ser feitas, porém em outros níveis de uso de memória, que inclui quando a memória está relativamente vazia. Por exemplo, a frequência em que uma página carregada é acessada pode ser calculada independente se o sistema está sob pressão de memória. Além disso, em vez de gerenciar somente o que está atualmente em memória, a presente invenção gerencia a memória de forma global, incluindo o que está atualmente em disco, e vice-versa.

[0032] A presente invenção representa o estado de memória de

inúmeras maneiras, que inclui ao adicionar as informações que são atualmente mantidas em torno de cada página de memória. Tais informações novas podem incluir a frequência de cada página de uso com o tempo, em vez de simplesmente informações implícitas e explícitas com base no tempo de último uso. Implementações exemplificativas mantêm também informações sobre páginas que não estão atualmente presentes em memória. Conforme descrito abaixo, as estruturas de dados são utilizadas para organizar a memória livre que segura ou pode segurar dados potencialmente úteis, e são usadas para controlar quando a memória é feita disponível para usos alternativos.

[0033] Para obter as informações, um rico conjunto de informações sobre cada página é monitorado através de técnicas de rastreamento e registro, de preferência, com relação a um período de tempo relativamente longo. Ao observar as diversas páginas usadas por processos quando ativas com relação a um período relativamente longo, o código que é acessado e os arquivos de dados que são lidos são gravados, proporcionando um cálculo significativo do valor de uma página para, assim, estabelecer conjuntos de trabalho potenciais. Estes dados calculados são mantidos e acessados como parte de um processo de tomada de decisão que reequilibra os conteúdos de memória em resposta a eventos relacionados com a memória e/ou de outra maneira (Por exemplo, de forma periódica). Em outras palavras, o monitoramento de dados ricos resulta em informações que quando analisadas permitem que mais páginas valiosas sejam mantidas e/ou pré-buscadas dentro da memória com relação a páginas menos valiosas, se o valor for determinado por dados mais prováveis de serem necessários, facilidade de transferência de dados, contexto e/ou outros critérios.

[0034] Em geral, a presente invenção mantém informações ricas não tradicionais por página de memória para obter um valor relativo

para cada página, tanto para identificar páginas de alto valor como páginas de baixo valor, quer a página esteja atualmente em memória quer não ou em outro armazenamento. Essas ricas informações (incluindo dados de padrão/frequência/agrupamento/e contexto) são mantidas em memória (e também persistidos) para páginas em memória bem como para páginas que não estão mais em memória (por exemplo, estão atualmente sobre um disco rígido ou flexível, mídia de CD, mídia de DVD, rede, memória instantânea e assim por diante) ou nunca estiveram em memória. Estas informações são usadas para pré-carregar (pré-buscar) páginas valiosas que não estão em memória, dentro da memória, e/ou para manter páginas mais valiosas com relação a páginas menos valiosas. Uma maneira é isto ser realizado através de um mecanismo de complemento de conjunto de trabalho que continua a trabalhar em entradas de conjunto que contêm as informações de uso ricas para páginas que não estão mais em memória porém são válidas no espaço de endereço para o conjunto de trabalho. Outra maneira é através de um conjunto de página inativa estendida, (por exemplo, um cache de páginas que não estão em uso, tal como uma lista inativa em um sistema baseado em Windows®), que é construído baseado no mecanismo histórico de página de arquivo, onde mais descritores de página física e entradas são alocados que há memória física, e estas são usadas para manter informações padrão de uso, frequência, agrupamento e/ou contexto sobre páginas que deixam a memória.

[0035] Diversas estatísticas são monitoradas para derivar padrões de referência e/ou frequência de uso que determina o indício da necessidade de uma página, tal como uma conta de período utilizado, tempo de criação, tempo de último acesso e assim por diante, conforme descrito abaixo. Estes tipos de estatísticas podem ser mantidos em memória por entradas de página ou faixas/grupos que podem ser usa-

dos para derivar padrões de referência, frequência de uso e assim por diante. Tais estatísticas podem ser entradas mantidas e/ou objetos para arquivos, objetos de mapeamento de memória ou outros grupos de página. Ademais, máquinas de estado podem ser dispostas por página ou grupo de página que possui estados além de bits de idade, tais como para cobrir quando uma página for removida da memória e trazida de volta. Estas informações variadas podem ser usadas para priorizar páginas em memória em subconjuntos, e proteger páginas valiosas contra repropostas, tais como quando os programas fazem referência a grandes quantidades de páginas uma ou duas vezes. A priorização pode ser explícita, e não inferidas a partir da posição em uma lista. A prioridade pode ser periódica ou ocasionalmente recalculada partir de padrões de uso, frequência, contexto, agrupamento capturados e assim por diante.

[0036] Ademais, o custo para trazer uma página particular para dentro da memória pode ser um fator para determinar o valor da mesma, por exemplo, se a página for retrocedida ao arquivo de página, em um arquivo pequeno, ou entre páginas que são sequencialmente acessadas/movidas podem ser usadas para estabelecer este fator de custo. Tal avaliação pode ser feita por página, porém também pode ser feita mais compacta, tal como ao manter algumas ou todas informações ricas somente em um subconjunto de páginas, obter e/ou manter estatísticas e contas de grupo de páginas, para objetos de arquivo, para seção/mapeamentos, e assim por diante. Ademais, os dados de estado podem ser obtidos e utilizados como parte das informações de página estendida.

[0037] Por exemplo, ao observar o contexto, fatores que não estão normalmente sob consideração em decisões de gerenciamento de memória podem ser levados em conta, incluindo estado da potência e estado do disco. Os eventos observados são também levados em con-

ta, para tirar ações apropriadas para rápida recuperação a partir de inatividade/retorno, hibernação/retorno, e execução/terminação de programas que são conhecidos por possuírem grande impacto de memória.

[0038] Além de gravar tais informações de estado, outros mecanismos estão disponíveis para facilitar o gerenciamento de memória proativa da presente invenção. Por exemplo, os programas aplicativos podem declarar que certos dados são menos (ou mais) úteis que outros dados, por exemplo, dados necessários para lidar com um erro raro podem ser declarados como não indicados para serem usados novamente, onde tais dados são melhores candidatos a serem atribuídos ou ponderados com um valor inferior. Um usuário também pode indicar certos dados de prioridade maiores ou menores, por exemplo, um usuário que raramente usa um arquivo de ajuda de aplicativo pode proporcionar manualmente uma configuração para indicar a baixa prioridade da mesma para o aplicativo e/ou o sistema operacional, ou vice-versa, por exemplo, sempre se possível manter os dados de ajuda em memória. Para esse fim, as interfaces de programação de aplicativo (ou funções similares) para aplicativos podem ser proporcionadas para permitir que os programas contribuam para o gerenciamento de memória de sistema, tal como ao especificar que as páginas (por exemplo, uma região de páginas) não são mais valiosas, ou que outro conjunto de páginas é valioso. Os programas aplicativos podem também especificar contextos e operações que o gerenciador de memória pode usar para monitorar e valores de grupo de páginas referenciados naqueles programas. Desta maneira, os programas aplicativos podem marcar operações que o programa aplicativo acredita ser importantes com relação ao uso de memória, tal como exibir um diálogo de Abrir/Salvar Arquivo ou mostrar facetas de tarefa.

[0039] Os valores de página também podem ser usados para me-

lhorar o gerenciamento de memória de outras formas. Por exemplo, quando um programa efetua grandes quantidades de páginas de arquivo que são utilizadas uma ou duas vezes, somente páginas similares são repropostas, e as páginas de maior valor são protegidas. Além disso, a compactação de certas páginas é também possível, tal como ao possuir páginas de valor baixo e/ou intermediário compactadas em memória em vez de trocadas por disco. Por exemplo, um mecanismo aplicativo ou de descoberta/de partida pode indicar que deve ser desejável compactar algum bloco amplo de (por exemplo, valor baixo ou intermediário) dados em vez de gravá-los em disco para espaço livre.

[0040] As técnicas de rastreamento e registro que são usadas para proporcionar novas habilidades para gravar dados de uso de memória que passam entre estes atualmente conhecidos. Novos algoritmos podem minar estes dados e gerar instruções ou planos de cenário para pré-buscar dados a partir de disco de maneira eficiente, e para manter dados em memória baseados em priorização de valor. Por exemplo, devido às ricas informações, um conjunto de página inativa de páginas de memória pode ser priorizado, em vez de simplesmente dispostos em uma fila do tipo FIFO onde não há prioridade quantificada. Com a presente invenção, o conjunto de página inativa pode ser de maior prioridade em termos de valor com relação a outras páginas, independente de quando referenciadas pela última vez, e desta maneira mantém em memória com relação a outras páginas, que incluem estas na memória de conjunto de trabalho. Por exemplo, as páginas de um certo processo ativo podem provavelmente nunca ser usadas novamente, mesmo que essas páginas tenham sido recentemente usadas, onde tais páginas são de valor menor que as páginas que não foram usadas por um período muito longo porém potencialmente reutilizáveis.

[0041] Ademais, além de dados de estado observados e resulta-

dos calculados tais como frequência de uso real, tempo de último uso e outros resultados calculados, outros critérios podem ser usados para determinar o valor de uma página. Por exemplo, o conceito de uma penalidade de substituição pode ser usado para determinar o valor de páginas em memória, por exemplo, dados dispersos de forma randômica (não contíguos em disco) são mais caros para transferir para e a partir de disco do que os dados sequenciais contíguos, e desta maneira estes podem ser mais valiosos para manter os dados dispersos de forma randômica em memória, e transferir os dados sequenciais, tais como se uma grande quantidade de memória for requerida de repente. Desta maneira, a utilidade de uma página pode ser determinada pelo custo de transferência I/O da mesma, juntamente com o rastreamento histórico de uso e máquina e outro contexto da mesma tais como considerações de estado de disco e potência. Ainda outros fatores podem ser usados, tais como o peso de uma entidade (por exemplo, aplicativo ou usuário) podem atribuir a certos dados.

[0042] Em uma implementação, o uso de memória de programas é monitorado em uma máquina, por exemplo, por usuário, e a utilidade potencial de páginas tanto em disco como em memória são categorizadas. A partir deste monitoramento e categorização, um orçamento de memória pode ser determinado, e baseado no orçamento, um mecanismo de segundo plano I/O tenta preencher a memória com as páginas de mais alta utilidade disponíveis, como determinado pelo valor quantificado de cada página. As páginas de memória desta maneira podem ser resilientes, por exemplo, removidas se necessário porém automaticamente re-armazenadas quando o espaço torna-se novamente disponível. Nota-se que esta abordagem, que conta com paginação de demanda re-ativa (com agrupamento limitado ou pré-buscado) para trazer as páginas do disco somente no momento da primeira vez de uso, e as mantém de maneira FIFO. Com a presente

invenção, quando as páginas estiverem residentes, as estruturas de memória mantêm as páginas em memória sobre uma base priorizada, que pode variar com o tempo à medida em que estas são usadas, tal que as páginas de maior utilidade permanecem em memória a menos que e até a memória ser necessária para outras atividades atuais.

[0043] Para identificar as páginas de alto valor (bem como baixo), e/ou dados de padrão de acesso são propagados e mantidos em um armazenamento persistente não-residente, onde estas informações (por exemplo, que inclui dados de padrão, frequência, contexto e/ou agrupamento) são persistidos através de partidas. Nota-se que além de gravar os dados de página para arquivos, estas também são mantidas, que inclui interpretar os dados, construir novos valores, e/ou extrair os dados para descartar dados redundantes ou desnecessários, tal que o que for salvo nos armazenamentos persistentes compreende não só dados de uso de página, como também dados de padrões discernidos. Conforme descrito abaixo, o valor de página e dados de valor de página e padrão de acesso que são mantidos e usados pelo gerenciamento de memória são capturados por rastreamento e/ou leitura de referências de página, juntamente com outros contextos e eventos (como descrito abaixo). Acompanhando a presente invenção, a operação de rastreamento e amostragem vai além de simplesmente manter as contas e uma quantidade pequena de estado por página em memória. Para esse fim, os mecanismos de rastreamento e leitura incluem rastreamento de bits acessados de PTE (entrada de tabela de página) que efetua o registro quando os bits acessados forem reinicializados. Nota-se que ao calcular e manter a taxa em que os dados de acesso são conduzidos e reinicializados limita-se a quantidade de dados que está efetuada em registro, por exemplo, para páginas frequentemente referenciadas, as informações não são efetuadas em registro em todos os acessos, porém de preferência a taxa em que os bits de acesso

PTE são inicializados pode ser mantida. Ademais, as falhas de páginas (falhas irrecuperáveis e recuperáveis/falhas de transição) e pedidos de arquivo de gravação e leitura podem ser postos em registro. Os mecanismos para conjuntos de trabalho de processo instantâneos podem também obter dados de valor de página, como podem os mecanismos para rastrear a página à medida em que eles deixam a memória ou à medida em que são aparados dos conjuntos de trabalho.

[0044] Com estas informações, as páginas podem ser agrupadas e as informações extraídas, tanto para páginas no armazenamento persistente como em memória, através de seus períodos de último acesso, padrões de uso, frequência de uso, localidade física e virtual, quando as páginas são aparadas, e assim por diante. Estes agrupamentos podem então serem usados para trocar de forma eficiente as regiões de páginas em um período, e podem ser usadas no esquema no destino, tal como em arquivo de página. Por exemplo, se poucas páginas forem usadas, o grupo todo pode ser trazido de forma eficiente. Os mecanismos podem realizar "agrupamento inteligente" ao utilizar o arquivo persistente/histórico de página, tal como trazer todo o arquivo em um período, e assim por diante. As páginas podem ser adicionalmente agrupadas através de processos que as utiliza, com relação à parte da existência de um processo (por exemplo, disparo, desligamento), que as páginas são utilizadas, pelo qual as páginas falham irrecuperavelmente dentro de um período (por exemplo, operações de limite de disco). Outras formas de agrupamento de páginas incluem páginas referenciadas em torno de troca de tarefa, clique de mouse, hesitação, foco e outra entrada de usuário, uso baseado no tempo tal como nos períodos do dia e/ou dias da semana particulares, através dos processos de um usuário específico, e outros contextos, que incluem contextos especificados em aplicativos.

[0045] Em geral, o gerenciamento de memória de acordo com a

presente invenção pode monitorar e adotar medidas com base em um rico conjunto de eventos e contextos, que incluem comutação de tarefa, criação de janela, diálogo, mudança de foco, hesitações e cliques de mouse, e outros eventos de entrada de usuário. Outros eventos e contextos incluem disparo e desligamento aplicativo, inicialização de sistema, inatividade/hibernação, retorno, troca de usuários, vídeo, áudio e/ou eventos de reprodução de mídia (por exemplo, início e fim), deslocamento de tempo de vídeo, conexão remota para o sistema (que inclui área remota de trabalho, protocolo telnet, e assim por diante, e contextos especificados de aplicativos.

[0046] De acordo com um aspecto da presente invenção, as páginas mais valiosas não só são mantidas em memória com relação a páginas menos valiosas, como as páginas mais valiosas podem ser carregadas dentro da memória (pré-buscadas) com relação a páginas menos valiosas (a memória é automaticamente ocupada) mesmo quando não há demanda para estas páginas. Esta ocupação e/ou re-equilíbrio pode ser feito diversas vezes e de diversas maneiras, incluindo após a re-inicialização, após uma grande quantidade de memória ser liberada, ou mesmo um tanto de forma contínua ou pelo menos regular, de preferência conforme uma operação de segundo plano. Prefere-se que o mecanismo de ocupação usado pelo gerenciador de memória não seja intruso, e baseado nos mecanismos de prioridade de inatividade I/Os e de detecção de inatividade. Conforme descrito abaixo, isto não é só um algoritmo de programa de execução, mas pode operar de forma acelerada ao observar padrões normais de entrada I/O, e fundo de programa de execução I/O quando não for provável a sobreposição ou devido a uma busca por I/O normal. Os mecanismos de detecção de inatividade (por exemplo, enfileirado com uma tarefa inativa) determinam quando o sistema está inativo ao examinar a entrada de usuário, CPU e disco, para realizar o equilíbrio de memória de

maneira reservada. Entretanto, a ocupação de primeiro plano da memória é também valiosa de vez em quando, e desta maneira algumas ou todas as pré-cargas da memória podem ser altas em prioridade.

[0047] Durante a pré-busca, as operações I/O podem ser desta maneira gerenciadas para proporcionar eficiência através de mecanismos que restringem a interferência entre os aplicativos ativos e os programas ao realizar trabalho de segundo plano. Por exemplo, as transferências I/O podem ser classificadas para reduzir os períodos de busca, e as transferências I/O podem ser priorizadas de modo que as operações de segundo plano (por exemplo, pré-busca) não retardem as operações de primeiro plano e não interfiram no uso de sistema normal do usuário. Ademais, os mecanismos pode ser proporcionados para categorizar os pedidos I/O e passar essas informações por pilha de acionador, de modo que os mecanismos de baixo nível estejam disponíveis para programar a execução I/O com a prioridade apropriada. Isto permite que as categorias de I/O de baixa prioridade não interfiram nas categorias de alta prioridade, que podem exigir que evitem possíveis inversões de prioridade quando a dada operação for requerida várias vezes sob categorias diferentes. As (entradas /saídas) I/O de categorias diferentes podem ser despachadas utilizando algoritmos que buscam reconhecer padrões predominantes de uso e minimizar as interferências.

[0048] Um dos aspectos da operação I/O é impulsionar os recursos sob utilizados sem prejudicar o desempenho de outra atividade de sistema e de usuário, entretanto podem ocorrer problemas na leitura de páginas de arquivo de alta utilidade do disco sem interferir nos aplicativos de primeiro plano do usuário. Um tal problema é o requerimento para pré-buscar a partir de arquivos que o usuário exclusivamente já abriu, tais como arquivos de registro e índice de cache de arquivos de internet temporários. Em tal situação na qual o arquivo já está exclusi-

vamente aberto, o arquivo não pode ser aberto normalmente para pré-busca a partir do mesmo. De forma semelhante, se o mecanismo de gerenciamento de memória abrir com êxito um arquivo para pré-busca, e um aplicativo que exija acesso exclusivo para o arquivo tentar abri-lo, o aplicativo falharia, resultando no que poderia ser percebido como erros de aplicativo aleatórios. Este problema pode ser desviado ao chamar um API especial para abrir o arquivo de maneira atípica na qual o sistema de arquivo não atualiza as informações compartilhadas para aquele arquivo.

[0049] Outro problema com relação a I/O é dirigido para emissão de pedidos I/O de maneira que não interfira nos pedidos de maior prioridade I/O sobre o sistema. Isto pode ser realizado, em parte ao não iniciar uma baixa prioridade I/O se o disco não estiver inativo por tempo suficiente (por exemplo, 75 ms) desde o último I/O normal. Entretanto, isto resulta em um problema de inversão de prioridade, que pode ser solucionado ao enviar uma leitura provisória com um sinalizador especial abaixo de uma pilha de I/O. Nota-se que o envio de um pacote de leitura provisória é uma maneira de comunicar-se com a pilha de I/O para informá-la da inversão de prioridade, que trabalha mesmo quando o pedido real for violado para muitos sub-pedidos que são dispersados na pilha tal que todos estes não possam ser atingidos através do manejo do pedido normal. Quando o acionador de filtro vê a correspondência IRP (pacote de pedido I/O), o mesmo promove a prioridade de todos I/Os relevantes do mesmo que sobrepõe-se com esta faixa, e a leitura provisória IRP é então completada sem realizar nenhum trabalho real. Entretanto, mesmo com este ajuste, que emite um ampla lista de paginação de baixa prioridade I/Os pode bloquear um encadeamento de alta prioridade. Desta maneira, isto pode ser evitado ao emitir somente pouca paginação de pré-busca de baixa prioridade I/Os, para um arquivo único em um período, que reduz grandemente o

potencial para inversão de prioridade, e encurta sua duração. Com estes aperfeiçoamentos, uma vez que as páginas de arquivo de alta utilidade sobre o disco e as páginas de baixa prioridade em memória são identificadas, os pedidos podem ser imediatamente enfileirados sem preocupa-rem-se com a interferência na atividade atual do usuário.

[0050] Como resultado deste aspecto da invenção, a utilização da CPU é significativamente aumentada sem interferir no uso normal de sistema. Para muitos usuários típicos, todo o conjunto de páginas retornadas de arquivo realmente usadas é pequeno o bastante para manter-se em memória. As páginas necessárias podem ser pré-carregadas após à inicialização, e também re-armazenamento se ocorrer constantemente uma alta demanda temporária para memória e estas páginas tiverem de ser trocada por disco. Nota-se que se o conjunto de trabalho agregado de memória necessária for muito grande para um dado sistema e aplicativo, os subconjuntos do conjunto de trabalho podem ser mantidos em memória, ao mesmo tempo em que a presente invenção irá se esforçar para proporcionar I/O eficiente para o equilíbrio das páginas.

[0051] Entretanto, ao testar muitos cenários de clientes comuns, foi mostrado que o número total de páginas referenciadas mesmo em uma carga de trabalho relativamente complexa ajusta-se dentro de quantidades contemporâneas de memória. Por exemplo, conforme determinado através de uma variedade de experiências, as páginas necessárias para algumas cargas de trabalho muito significativas foram capazes de se ajustar dentro da memória de máquinas com 384 megabytes de RAM, e muitas em 256 megabytes de RAM. Devido ao fato da memória ser maior que o necessário, a presente invenção torna a mesma viável para pré-carregar as páginas apropriadas e eliminar as falhas de página, e desse modo executar os aplicativos de usuário em utilização total de CPU em resposta à entrada de usuário.

[0052] Como resultado da eliminação (ou quase eliminação) de falhas de página, a presente invenção permite que o disco seja gerenciado de forma muito mais eficiente do que feito presentemente. Por exemplo, em muitos casos é possível eliminar uma situação na qual um retorno do estado de inatividade é retardado por rotação ascendente de disco. Desta maneira, um benefício significativo de gerenciamento de memória estendida e proativa da presente invenção é que dispositivos portáteis, tais como dispositivos de computação de mesa gráfica, portáteis e outros dispositivos podem executar por longos períodos de tempo sem a leitura a partir do disco. Ao manter o disco a partir de rotação descendente quando sobre potência de bateria representa um avanço de economias de potência significativas para usuários móveis, como um disco que é girado de forma descendente por períodos estendidos aumenta a vida da bateria em máquinas portáteis, e aumenta a vida do disco em geral.

[0053] Sem contar, leituras de disco, a única razão para manter um disco em rotação é satisfazer as gravações. Entretanto, a atividade de gravação típica compreende gravações repetidas para os mesmos setores de disco, e inúmeros setores distintos gravados por longos períodos são tipicamente simples. Como resultado, um cache de setor de disco simples, que utiliza memória restaurada de bateria ou outro NVRAM não-volátil pode geralmente manejar gravações, como representado na Figura 3, onde uma quantidade relativamente pequena de memória de acesso aleatório não-volátil (como representado pela caixa pontilhada) opcional (NVRAM) 302 pode ser parte de um sistema de armazenamento não-volátil total 300 que permite que o disco permaneça em rotação descendente quando algumas gravações não podem ser evitadas ou são desejáveis, por exemplo, quando um usuário salva um documento. Em uma implementação, NVRAM de sistemas testados capacitados de 32 megabytes para reter tais gravações de

limite de disco por períodos relativamente longos, durante os quais o disco foi capaz de ser girado de forma descendente, ao mesmo tempo em que realiza tarefas comuns como ler e-mail, navegar na rede, editar e salvar documentos, e assim por diante. Nota-se que a memória de acesso aleatório 302 pode ser gerenciada de forma proativa através de um sistema de gerenciamento de armazenamento proativo opcional 304 (que pode ser um tanto similar ao sistema de gerenciamento de memória proativa 200 descrito no presente documento) para trocar as páginas armazenadas de menor valor a partir de NVRAM 302 por disco 204 quando o disco estiver girando, para liberar espaço para quando o disco for posteriormente girado de forma descendente

[0054] Embora girar um disco de forma descendente imponha uma penalidade severa no primeiro acesso de disco requerido, o gerenciamento cuidadoso de cache de setor e de memória irá capacitar o impedimento de retardo síncronos na maioria das situações. Isto não será sempre possível, entretanto, e desta maneira o sistema operacional terá de fazer decisões estratégicas sobre quando é mais vantajoso parar e reiniciar os discos, e em alguns casos, futuros aplicativos devem ser capazes de ajudar em reinicializações de discos de ponte.

[0055] Para resumir, em geral, a presente invenção é dirigida com respeito às técnicas de gerenciamento de memória que reduzem substancialmente a necessidade de utilizar o componente mais lento, o disco, como um fator durante o uso de sistema normal. A presente invenção, através de uma nova abordagem resiliente e proativa para gerenciamento de memória, é disposta para distribuir compreensividade consistente em operações significativas relacionadas com a memória mesmo após a inicialização, troca rápida de usuário, ou saída de um grande aplicativo tal como um programa de jogo. Os retornos confiadamente rápidos da inatividade não atrasaram em rotação ascendente de disco também resultam da presente invenção, bem como tempos

de resposta reduzidos por operações grandes e caras que são atualmente limitadas a disco devido à paginação de demanda. Outros benefícios resultam do armazenamento em cache de memória de auto-sintonização e esquema de arquivo automático sobre o disco que adapta-se aos padrões de uso de sistema e proporciona desempenho total melhorado. Ainda outros benefícios incluem problemas de mídia drasticamente reduzidos, ao projetar a memória e a largura de banda de disco usadas por aplicativos de mídia, proteção de atividades de segundo plano tal como dispositivos de varredura anti-vírus, índices de arquivo e programas de procedimento de identificação que podem incapacitar a compreensividade interativa, e a vida da bateria, baixo ruído, e baixa vibração para dispositivos móveis, que resultam da conservação de discos em rotação descendente por longos períodos de tempo. As capacidades de fluxo melhoradas para aplicativos de mídia e do tipo e-home que possuem múltiplos fluxos de segundo plano também resultam da presente invenção.

[0056] A seção seguinte descreve diversas implementações exemplificativas, que devem ser observadas somente algumas maneiras de utilizar as informações que podem ser obtidas para gerenciar memória

IMPLEMENTAÇÕES EXEMPLIFICATIVAS

[0057] Pelo menos uma implementação exemplificativa da presente invenção descrita abaixo foi implementada em um sistema modificado executando um sistema operacional Microsoft Corporation's Windows® XP SP1 de execução e o sistema de arquivo Windows® NFTS. Entretanto, não há intenção de limitar a presente invenção a nenhum sistema operacional ou tipo de gerenciamento de memória, porém ao contrário, a presente invenção tende a operar e proporcionar benefícios com qualquer computador que tenha a memória gerenciada por um sistema de gerenciamento de memória, devido ao fato de a pre-

sente invenção incluir um esquema de gerenciamento de cache inteligente e especulativo. Além disso, embora uma implementação exemplificativa seja disposta com uma unidade de disco como um armazenamento secundário a partir desta a memória é carregada, deve estar prontamente óbvio que o armazenamento secundário não é necessariamente uma unidade de disco, e inclui alternativas tais como memória instantânea, fonte de dados de rede, unidade de fita ou virtualmente qualquer tipo de armazenamento de dados, e de fato um sistema de computador que é completamente sem disco pode tirar muitas vantagens da presente invenção.

[0058] De acordo com um aspecto da presente invenção e como geralmente representado na implementação exemplificativa 400 das Figuras 4A e 4B, são desenvolvidos diversos mecanismos que implementam a abordagem da presente invenção para gerenciamento de memória proativa, resiliente e de auto-sintonização. Tais mecanismos proporcionam o rastreamento e registro de páginas referenciadas, a proteção de páginas mais valiosas da erosão, e a busca e rearmazenamento destas páginas para transições seguintes de memória. Estes mecanismos executam como parte do núcleo quando necessário, porém pode incluir também serviços (desprivilegiados) de modo de usuário.

[0059] Conforme descrito acima, para rastrear e registrar o uso de memória e outros dados de contexto, a presente invenção acumula informações para obter dados ricos, que incluem quais as páginas acessadas para atividades frequentes do usuário atual. Por exemplo, cada acesso à páginas através de um processo simples, ou por todo o sistema em um período de tempo especificado, pode ser registrado. Para esse fim, como representado na Figura 4A, em uma implementação de gerenciamento de memória exemplificativa 400, um mecanismo de rastreamento 402, ou rastreador, rastreia referências de páginas ao

monitorar a atividade relacionadas com a página 404, que inclui monitoramento de bits de acesso que corresponde às páginas que foram acessadas, e examinar como as páginas são trazidas para dentro da memória através de falhas.

[0060] Conforme também representado na Figura 4A, o rastreamento de informações é persistido por uma parte registrante 406 dentro de estruturas de dados (por exemplo, que incluem um ou mais registros prévios 408 e registros brutos 410, descritos abaixo) que identificam cada página como um par de deslocamento de arquivo, ou, para páginas privadas tais como pilhas, um par de endereço de conjunto virtual de trabalho. Mais particularmente, em uma implementação, o mecanismo de rastreamento 402 registra os eventos de memória virtual relacionados com o processo em armazenamentos temporários de núcleo (não mostrados), a partir dos quais os registros brutos 410 são posteriormente montados. Outra peça útil de informações que pode ser obtida pelo mecanismo de rastreamento 402 monitora se a página (quando referenciada) esteve residente em memória ou teve de ser falhada de forma irrecuperável a partir do disco. Os eventos de memória virtuais incluem acessos iniciais em páginas particulares de um espaço de endereço e contadores de acesso que registram a proporção em que a página está sendo re-acessada. As informações também estão incluídas no estado da máquina, e os contextos reconhecidos ativam-se no momento do rastreamento. Para cada página, as informações adicionais podem ser mantidas, tais como em que os disparos recentes de um cenário cada tal página foi referenciada, e informações de estado de sistema similares.

[0061] Desta maneira, uma base importante da abordagem de gerenciamento de memória da presente invenção é efetuar registro de eventos de gerenciamento de memória observados no sistema, isto é, representações de acessos em partes de espaços de endereço virtual

através de processos particulares. Pode não ser possível registrar todos os acessos, e desta maneira na implementação descrita o trabalho é realizado na granularidade de página, para registrar os primeiros acessos e informações sobre a frequência de acessos subsequentes.

[0062] A parte registrante 406 opera obtendo os rastreamentos de núcleo e formatando-os como registros brutos 410. As responsabilidades da parte registrante incluem garantir que os limites de recuso sejam respeitados para que os registros não ocupem o disco do usuário. Os rastreamentos são desta maneira registrados nos registros brutos 410 que representam o padrão de acesso para uma instanciação particular de um processo com relação a algum intervalo razoavelmente curto. Por exemplo, os registros devem ser coletados para uma inicialização simples ou retorno de hibernação, para o período que segue algum evento de usuário, tal como criar o painel de controle, pelo período que um aplicativo seja o aplicativo de primeiro plano, para algum intervalo de tempo não fixos em um evento particular, e assim por diante.

[0063] Em uma implementação descrita, os registros incluem um cabeçalho, que pode conter contexto, estado de memória, e informações de tempo, e incluem uma tabela de <fonte, deslocamento, falha irrecuperável> entradas, onde a fonte é o arquivo ou espaço de endereço de processo acessado em um dado deslocamento, e a falha irrecuperável indica se os dados foram lidos a partir do disco, que proporciona um cálculo aproximado de custo.

[0064] Além de rastrear o uso de página para diversos contextos, as informações sobre padrões de acesso e existência de páginas individuais que estão em memória também podem ser mantidas, ao proporcionarem a habilidade para tomar decisões melhores quanto à utilidade de páginas dentro de conjuntos de trabalho. Por exemplo, os sistemas operacionais contemporâneos mantêm somente dois bits de

informações de idade por página, pela qual até o gerenciador de memória, todas as páginas que não foram referenciadas recentemente (por exemplo, nos últimos poucos minutos) tendem a parecer iguais. Na presente invenção, as informações para as quais o sistema gerenciador de memória tem de acessar pode incluir inúmeros períodos (por exemplo, minutos) nos quais cada página foi utilizada, bem como os períodos em que cada página foi criada e acessada por último. No desenvolvimento da presente invenção, os mecanismos mantêm estas informações sobre as páginas mesmo quando estas páginas forem paginadas para disco. Em outras palavras, a página não só é monitorada enquanto em memória, como estas informações são mantidas ao mesmo tempo em que a página está sobre o disco. Esta informação adicional proporciona uma imagem significativamente mais precisa de como a página está sendo acessada com relação à duração da mesma.

[0065] O mecanismo de rastreamento 402 pode trabalhar em múltiplos rastreamentos ativos simultaneamente. Estes rastreamentos podem incluir rastreamentos contínuos, que por exemplo podem ser montados um após o outro até estes atingirem um tamanho particular (por exemplo, 100.000 registros) ou até estes representarem uma duração particular (por exemplo, um intervalo de cinco minutos). Nota-se que um mecanismo de rastreamento de implementação possui um overhead relativamente muito baixo, (por exemplo, 0,1% de CPU com um processador de 450 MHz, somado com poucos megabytes de memória para efetuar registro de um dia de uso de página), que habilita tal rastreamento contínuo quanto a que páginas estão sendo referenciadas, sem sobrecarregar o sistema. O rastreamento também pode ser realizado como uma operação do tipo amostragem, intermitente, tal como automaticamente acionada ou em resposta a pedidos através de uma ferramenta GUI para testar ou treinar propósitos, que

permite que a coleta de informações de página seja associada com atividades e transições particulares, tais como disparo de aplicativo, troca rápida de usuário ou retorno de inatividade.

[0066] Nota-se que em vez de iniciar o rastreamento/ registro a partir do temporário para um usuário/sistema, alguma quantidade de pré-treinamento pode ser realizada (por exemplo, para outros usuários e sistemas). Isto permite a cópia de algum arquivo e valor de página e base de dados pré-construídos no tempo de configuração, ou outros períodos adequados, tais como quando o programa aplicativo é instalado. Esta solução estática ajuda a acelerar a linha de aprendizado que pode ser de outra maneira requerido a qualquer momento um novo usuário, novo sistema, nova versão de sistema operacional e/ou novo programa. Desta maneira, a presente invenção proporciona vantagens através de carga de memória seletiva baseada no valor ou toda carga de memória, onde o valor é determinado pelo menos em parte na pré-observação, seja através de histórico de uso de monitoramento, e/ou através de trino (que inclui simulação, análise de código, execuções em máquinas diferentes e assim por diante). Para manter os dados, um repositório persistente de informações que pertence ao uso de página o valor é construído, utilizado e armazenado, que pode estar em qualquer arquivo, que inclui arquivos executáveis, arquivos de dados de propósito especial e de documentos e base de dados. Ademais, nota-se que tal conhecimento pré-obtido também proporciona reduções significativas em transições I/O sob demanda mesmo sem realizar o acúmulo dinâmico de informações através de registro e rastreamento. Por exemplo, as transições I/O sob demanda podem ser reduzidas ao proporcionar simplesmente arquivos de grupos de página pré-construída, por exemplo, quando um usuário falha na primeira página para exibir o Painel de Controle, outras páginas relacionadas são trazidas para dentro da memória de forma eficiente a partir do disco.

[0067] Através do mecanismo de rastreamento 402 e do processo adicional descrito no presente documento, o sistema de gerenciamento de memória constrói e mantém planos de cenário (por exemplo, arquivos) 422 (Figura 4B) que contêm os dados que podem ser usados para estabelecer um ou mais valores calculáveis por páginas de arquivo, tal como frequência de uso. Para essa finalidade, diversos componentes podem processar os dados de forma adequada para tomar decisões de gerenciamento de memória, que inclui um integrador 412 que combina essencialmente os registros 408 e registros brutos prévios 410 em um base de dados de informações de página 414.

[0068] O integrador 412 representado na Figura 4A é um serviço, que pode executar em um processo de nível de usuário, (à medida em que o núcleo não é requerido), que lê os registros brutos 410, processa-os e integra as informações que estes contêm com informações de página para contextos similares já armazenados no sistema (por exemplo, nos registros prévios 408). Inicialmente não haverá informações prévias, porém o sistema irá atingir rapidamente um estado onde os rastreamentos brutos servem primeiramente para atualizar e aprimorar as informações já conhecidas a partir de prévias ocorrências de um contexto particular. Um integrador contínuo 412 que foi implementado mantém um vetor de bit histórico para cada página mencionada em rastreamentos para uma dada classe de contexto. Para integrar cada nova instância, o vetor de bit histórico é deslocado e um novo bit é adicionado representando o período de tempo mais recente. As páginas que aparecem no novo registro possuem um "1" como o novo bit, enquanto outras possuem um "0". As páginas que aparecem pela primeira vez são adicionadas a um base de dados de informação de página 414, enquanto as páginas que não estão sendo utilizadas recentemente podem ser removidas do base de dados 414.

[0069] Desta maneira, através de um integrador 412, os registros

brutos 410 que cobrem eventos ou intervalos particulares são integrados com registros prévios 408 que contêm informações conhecidas sobre prévias instâncias de eventos similares, tais como prévias execuções do mesmo programa, para proporcionar o base de dados de informação de página 414. O base de dados de informação de página 414 causa essencialmente um colapso de informações contidas nos registros, para reduzir a quantidade de memória requerida para monitorar estes dados, ao mesmo tempo em que reorganiza as informações de modo que os eventos similares sejam apropriadamente agrupados, e em arquivos e páginas entre um registro bruto 410 e registros prévios 408 é identificada.

[0070] Conforme representado nas Figuras 4A e 4B, as informações no base de dados de informação de página 414 podem ser extraídas (por exemplo, através de um serviço de extração 420) para produzir os planos de cenário 422, que indicam a provável composição de futuros espaços de endereço virtual e a probabilidade de acesso a páginas dentro de espaço virtual em contextos particulares. Conforme descrito abaixo, estes planos de cenário 422 são usados por um re-balanceador 424 para prever os futuros acessos para o sistema, e permitir que o re-balanceador 424 crie novos planos de gerenciamento de memória em resposta a eventos observados pelo sistema. Por exemplo, os planos de cenário podem ser construídos a partir de um conjunto fixo de arquivos de informação de página que compreende informações quanto a cenários conhecidos como importantes para o sistema, tal como iniciar e desligar o sistema, e retornar da hibernação. As mensagens particulares são usadas para estabelecer contextos. Devem-se adicionar informações a estes dados a partir de cenários selecionados por um usuário, por exemplo, através de uma ferramenta gráfica interativa, e os cenários diferentes podem ser dadas prioridades de padrão que podem ser modificadas através da ferramenta

interativa.

[0071] Quando os planos de cenário 422 mudam, um priorizador de página de cenário 426 (por exemplo, um serviço de modo de usuário incorporado ou de outra maneira associado ao serviço de extração 420) processa os planos de cenário 422 para gerar uma lista de páginas que deve ser trazida para dentro da memória e protegida. Esta regeneração também pode ser acionada baseada em um tempo máximo transcorrido uma vez que finalmente realizada, tal como para garantir que os planos sejam regenerados pelo menos uma vez a cada quinze minutos. Em uma implementação, para executar a priorização, cada página é atribuída uma pontuação de prioridade calculada. Quando as pontuações são determinadas, as páginas são então classificadas através de suas pontuações calculadas para construir uma lista de página priorizada 428.

[0072] Um fator em pontuação, uso baseado na frequência, é determinado ao contar quantas vezes as páginas foram usadas em intervalos de tempo recentes, por exemplo dentro da última hora, últimas oito horas, último dia, última semana, e assim por diante, de acordo com os arquivos de cenário de dados. Para esse fim, o priorizador de página 426 continua a monitorar a frequência de uso de página por diversas aberturas de tempo, tais como através de depósitos de uma hora, oito horas, um dia, três dias, uma semana e três semanas. Baseado no número de vezes que uma página foi usada em cada um desses depósitos, uma página pode ser atribuída uma pontuação, tal como entre 0 e 127.

[0073] Cada plano de cenário pode também monitorar em que os últimos número de execuções (por exemplo, trinta e dois) a página foi usada, com rótulos de tempo para cada disparo. Uma verificação mais pesada é dada para os usos recentes da página. Ademais, as pontuações das páginas que estão em planos de cenário conhecidos, tal co-

mo um plano de cenário mantido pelo Start Menu, podem ser aumentadas para concedê-las uma maior prioridade.

[0074] A pontuação das páginas pode também levar em consideração o contexto em que uma página é observada ao ser utilizada. Desta maneira, as pontuações podem ser dependendo do estado da máquina, os aplicativos que estão em execução, ou ações de usuário recentes. Tais abordagens sofisticadas para prioridade de página irão ser provavelmente aperfeiçoadas com o tempo. Nota-se que devido ao fato de cada plano de cenário manter as informações históricas e de tempo com relação ao último número de execuções (por exemplo, trinta e dois) em que as páginas foram acessadas, é possível determinar de forma frequente como uma página de arquivo particular foi acessada através dos cenários, com relação a outras páginas para quais informações foram registradas.

[0075] Para gerar eventos, como representado na Figura 4B, proporciona-se um observador 430. O observador 430 compreende de preferência um encadeamento no núcleo de sistema operacional que monitora o estado de sistema, que inclui mudanças pronunciadas no uso de memória física. Ademais, as chamadas explícitas podem ser feitas para acionar a avaliação de estado quando certas ações forem proporcionadas, tal como iniciar desligamento ou hibernação. Ademais, os aplicativos podem chamar dentro do núcleo para notificar o sistema de mudanças de estado potencialmente interessantes no qual um rótulo de escolha de aplicativo pode ser aplicado.

[0076] Em uma implementação, os eventos rastreados incluem disparos de aplicativos e reconhecem mensagens de janela passadas no núcleo. O gancho de mensagem de janela permite que o sistema detecte indicações de que os componentes de sistema ou aplicativos podem estar realizando uma operação significativa relacionada com a memória, tal como o disparo do Menu de Inicialização, ou a criação de

diálogos ou janelas para Abrir Arquivo, Impressão, ou Painel de Controle. As heurísticas podem ser usadas para evitar o rastreamento redundante de eventos frequente ou firmemente compactados. Em resposta a esses eventos, o núcleo monitora o uso de página para um processo individual ao utilizar os bits acessados em entradas de tabela de página para páginas. Estes bits são esclarecidos quando o rastreamento começa e checados no final do rastreamento. Qualquer remoção ou reinicialização de bit acessado durante o rastreamento é também registrado. As páginas que são acessadas através de Arquivo de Leitura e falhas irre recuperáveis nos quais uma leitura de disco é realizada são registradas.

[0077] Quando observa-se uma mudança no estado de sistema ou uma chamada indica um evento importante, um evento 432 é gerado e o estado de sistema pode ser atualizado por uma avaliação de estado e um serviço de atualização 434. O rastreamento pode ser solicitado ou terminado, e/ou os dados de estado (e outras informações de memória de sistema) 436 passados para o mesmo, e o re-equilíbrio pode ser acionado, como representado na Figura 4B. O estado inclui contextos ativos bem como a distribuição atual de memória entre os conjuntos de trabalho, a lista-zero, a lista suja e o conjunto de página inativa. As informações de memória de sistema incluem dimensionamentos de endereço virtual e dimensionamentos de conjunto de trabalho, dados instantâneos de processo, falhas recuperáveis e irre recuperáveis, bits de acesso de memória sustentados em hardware, as informações de uso por página derivadas pelo sistema operacional, e qualquer outro dado calculado ou observável, contextos e estados que podem ser usados para determinar os valores de página, conforme descrito acima.

[0078] Conforme geralmente descrito acima, tanto de forma contínua como no final do rastreamento, o núcleo produz um armazena-

mento temporário de rastreamento, que inclui listas das páginas específicas necessárias a partir de diversos arquivos, bem como uma variedade de contadores de gerenciamento de memória interno, tal como quantas páginas livres/zero estavam no sistema e a constituição de um conjunto de página inativa. Conforme também descrito acima, os armazenamentos temporários de rastreamento são coletados por um serviço de nível de usuário e usados para atualizar os arquivos de cenário, por exemplo, mantidos em um diretório de pré-busca. Os arquivos de cenário podem ser identificados baseados em um contexto que foi rastreado, tal como `AppLaunch.WMPlayer.exe-...pfx`, ou `Shell.StartMenuCreate...pfx`.

[0079] Desta maneira, os eventos de sistema provocam uma re-avaliação do estado de sistema, e podem acionar um re-equilíbrio de conteúdos de memória. Os contextos são indicados por marcadores que são disparados quando um código de sistema particular é executado, bem como especificado por aplicativos através de APIs ou similares. Os aplicativos podem proporcionar Uis através destes um usuário pode especificar e marcar por si só, os contextos individualizados. Outros eventos de sistema que podem provocar a re-avaliação incluem grandes alocações e desalocações de memória, o disparo de processos, ou detecção de que o usuário caiu em inatividade ou retornou de trabalho. Desta maneira, o gerenciador de memória e o re-balanceador 424 podem tirar ação dos gatilhos/eventos/ contextos, tal como para priorizar que as páginas são mais prováveis de serem necessárias em um contexto atual e dá-los um valor maior com relação a outros, e para pré-carregar a memória baseada nesta priorização. Estes contextos podem ser usados pelo processador de rastreamento para categorizar e agrupar os dados de uso de página rastreados. Quando identificadas, as páginas e grupos são re-dispostos em memória, tal como para trocá-los, traz-se todas as páginas relacionadas ao mesmo tempo

quando uma delas falha, e assim por diante.

[0080] O re-balanceador de memória 424 faz interface entre o modo de usuário e os elementos de modo de núcleo do política de pré-busca do sistema de gerenciamento de memória. Conforme descrito abaixo, o re-balanceador 424 primeiramente conta com o núcleo para identificar e ordenar as páginas em um conjunto de página inativa, para identificar as páginas de baixo valor e proteger as páginas de maior valor. Ademais para a proteção das páginas mais valiosas, uma primeira proposta do re-balanceador de memória 424 é ocupar a memória com as páginas que o priorizador de página de cenário 426 escolheu, ao substituir as páginas de menor valor pelas páginas de maior valor, como determinado pelas pontuações de prioridade. Para esse fim, o re-balanceador de memória 424 avalia o priorizador de página de cenário 426 e os contadores de conteúdo de memória de núcleo determinam se é necessário tirar alguma ação. Nota-se que os contadores de conteúdo de memória do núcleo dão informações sobre quantas páginas livres/zero estão disponíveis, bem como estatísticas para cada subconjunto de página inativa (descrito abaixo) e tal como quantas páginas foram re-propostas devido à pressão de memória

[0081] O re-balanceador 424 desta maneira observa a disposição atual da memória, e pode solicitar rotinas para originar conjuntos de trabalho a serem postos em ordem. A partir desta informação, o re-balanceador 424 então determina o orçamento de memória para ser usada para armazenamento de dados pré-buscados. Um conjunto de páginas que inclui-se no controle de gerenciamento especulativo é determinada ao identificar as páginas que proporcionam baixa utilidade, tais como páginas com conteúdos zerados além de certos limites estabelecidos pelo sistema. As páginas de baixa utilidade podem também compreender páginas que possuem conteúdos válidos que não estão em conjuntos de trabalho, e que não estão sendo usadas em um

período relativamente longo. Através de um priorizador de página de cenário 426, o re-balanceador 424 usa desta maneira as informações nos planos de cenário 422 para estabelecer um conjunto mais ideal de conteúdos para a memória, dentro do orçamento da mesma. Pode-se questionar (por exemplo, através de APIs) para determinar quantos conteúdos estão em memória e pode-se então modificar os planos 422.

[0082] Se houver páginas livres ou de baixo valor, o re-balanceador 424 pede para o núcleo de sistema ocupá-las com páginas de maior valor que utilizam I/O de pré-busca de baixa prioridade até as páginas de maior valor serem trazidas para dentro da memória. Para esse fim, uma vez que o re-balanceador 424 estabelece uma lista de páginas de acordo com o orçamento, a lista de páginas desejadas para carregar (e paginar externamente) são passadas para um planejador I/O 440. Estas instruções I/O são de preferência classificadas na fila de disco para minimizar as buscas, que resultam em uma medição(throughput) de disco muito maior. Se o re-balanceador 424 não for capaz de trazer tantas páginas para dentro da memória quanto gostaria, o re-balanceador 424 se lembra onde elas se encontram em sua lista, e quando as páginas tornam-se disponíveis, continua a pré-busca a partir daquele ponto.

[0083] Quando o priorizador de página de cenário 426 constrói uma nova lista de páginas, ou quando qualquer uma das páginas (resilientes) pré-buscadas que foram trazidas para dentro da memória forem re-propostas devido à pressão de memória, o re-balanceador de memória 424 inicia a pré-busca a partir do início da lista. Antes do re-balanceador 424 pré-buscar o novo lote de páginas, esclarece-se a preferência em páginas previamente pré-buscadas para torna-las disponíveis.

[0084] Nota-se que o re-balanceador de memória 424 pode ser

programado para ser relativamente moderado com relação a quais as páginas deseja-se trazer para dentro da memória, tal como evitar a interferência na atividade do usuário. Para esse fim, o re-balanceador é ativado para realizar operações I/O de baixa prioridade através do planejador de I/O 440, e pode evitar a tomada de ação quando o sistema está sob pressão de memória. Por exemplo, o re-balanceador 424 pode ser ativado somente para substituir as páginas que não foram usadas por um período relativamente longo, e que não foram usadas com frequência, e pode não tirar ação alguma se não tiver alguma quantidade limite de memória disponível no sistema (por exemplo, 16 megabytes) de alguma quantidade menor (por exemplo, 6 megabytes) as páginas são livres, zero ou de outra baixa utilidade.

[0085] O planejador I/O 440 trabalha para produzir conteúdos de memória desejados ao distribuir as I/Os, porém é forçado pela necessidade de evitar interferência ou outra I/O conduzida pelo sistema. Uma maneira disto ser realizado é aguardar por períodos de atividade relativamente ópticas I/O e interromper a tarefa em tamanhos de transferência um tanto pequenos. O sistema pode, entretanto, proporcionar esta classificação de não-interferência ao implementar um esquema de prioridade. Desta maneira o re-balanceador 424, por exemplo, de forma periódica e/ou quando acionado, decide quais as páginas devem estar na memória, baseado no uso antecipado, eficiência e assim por diante, conforme descrito acima, e o planejador I/O controla as transferências I/O para ajustar os conteúdos de memória como desejado, de maneira que tenta minimizar a interferência em outras operações I/O.

[0086] Conforme representado na Figura 5A, em gerenciamento de memória tradicional, as páginas a serem paginadas primeiramente adicionadas a uma lista de página inativa 500, que é uma lista FIFO simples. Ao ser uma lista FIFO, todas as páginas adicionadas 500 são

tratadas igualmente, sem considerar suas importâncias. Isto resulta em situações onde uma página que foi uma página valiosa é re-proposta antes de outra página que é muito improvável de ser usada, porém acontece de cair posteriormente na lista 500.

[0087] De acordo com um aspecto da presente invenção e conforme representado na Figura 5B, para aperfeiçoar esse esquema de maneira que proteja as páginas mais valiosas, um conjunto de página inativa priorizada, ou cache 502 é proporcionado que leva-se em conta os valores relativos das páginas, como determinado pela pontuação de prioridade baseada no rastreamento. Conforme representado na Figura 5B, tal conjunto de página inativa priorizada compreende subconjuntos 502_0 - 502_{15} , cada dos quais contém uma ou mais páginas atribuídas a uma prioridade particular. Nota-se que somente poucos subconjuntos (dezesseis) são mostrados na Figura 5B, entretanto como pode ser facilmente avaliado, que qualquer número prático de tais subconjuntos é viável, e os subconjuntos podem ser de vários tamanhos. O subconjunto 502_0 é um subconjunto desprotegido, que atua como o conjunto de página inativa existente 500 da Figura 5A; as páginas que não são adequadamente rastreadas podem passar nesta lista. Quando uma página precisa ser re-proposta a partir do conjunto de página inativa priorizada 502, a página é tirada do subconjunto não vazio, de prioridade mais baixa.

[0088] Desta maneira, quando um arquivo é fechado ou um conjunto de trabalho é posto em ordem, as páginas do mesmo vão para o conjunto de página inativa 502, com uma determinação feita no que diz respeito a qual subconjunto as páginas devem ser colocadas baseado nas informações de uso estendido, por exemplo, armazenados na entrada do base de dados PFN do gerenciador de memória para a página. Ademais para existência de página e campos de frequência de acesso, há campos que permitem a definição de prioridade da página

baseado nas informações do base de dados de cenário e do contexto atual do sistema.

[0089] O encadeamento do programa de gerenciamento de memória do sistema re-equilibra ocasionalmente (por exemplo, ao solicitar o re-balanceador 424, Figura 4B) o conjunto de página inativa priorizada 502, tal como uma vez a cada segundo. As páginas que foram colocadas em subconjuntos de alta prioridade que não são subsequentemente usadas por longos períodos são migradas para subconjuntos de menor prioridade.

[0090] Em uma modalidade, certas páginas são colocadas no subconjunto de maior prioridade 502₁₅, e aquelas páginas jamais são re-propostas. Isto proporciona um mecanismo para bloquear estar páginas em memória mesmo sob pressão de memória. Nota-se que este mecanismo é mais forte que o VirtualLock, devido ao fato dos processos que podem precisar destes poderem ir embora e voltar (e podem ainda, na realidade, ter que ser criados).

[0091] Desta maneira, ao acompanhar a presente invenção, estes mecanismos habilitam a seleção de páginas que são desejáveis para manterem-se em memória. Ao distribuir prioridades relativas a cenários, as páginas mais importantes não são geralmente re-propostas antes das páginas menos importantes. Ademais, uma determinação pode ser feita de modo que um conjunto de páginas de pouco valor ou nenhum valor possa ser substituído por páginas mais valiosas a partir do disco.

[0092] Em uma implementação, o sistema de gerenciamento de memória divide o conjunto de página inativa em dezesseis subconjuntos, que o mesmo utiliza para classificar diferentes classes de páginas e, ao ordenar os subconjuntos, para conceder um grau maior de proteção para algumas classes de páginas. Conforme descrito acima, uma página que contém dados úteis (tais como uma página de arquivo, ou

uma página de pilha para um processo de execução) porém não mais em algum conjunto de trabalho 450 (Figura 4B) é colocada em um subconjunto particular no conjunto de página inativa priorizada 502 baseada no conjunto de preferência no qual através do re-balanceador de memória 424, e/ou baseado no padrão de uso do mesmo enquanto este estiver em memória. Conforme descrito acima, os depósitos de tempo podem ser usados para monitorar o uso de páginas, por exemplo, em unidades de dez segundos, juntamente com uma conta do número de períodos que a página foi acessada desde a sua criação e o último período que a página foi acessada para taxar a idade da mesma.

[0093] Periodicamente, tal como uma vez a cada segundo, o re-balanceador 424 passa por uma parte do conjunto de página inativa 502 e re-avalia em qual subconjunto uma página particular deve estar. Em geral, uma página que não está em uso por um longo período é movida dentro de um subconjunto de menor prioridade apropriado.

[0094] A tabela 1 mostra como uma implementação determina o subconjunto apropriado para uma página. Se um aplicativo precisar de memória e não houver páginas livres ou zero disponíveis, as páginas de conjunto de página inativa serão re-propostas a começar pelo subconjunto de menor índice.

TABELA 1

PRIORIZAÇÃO DE SUBCONJUNTOS INATIVOS

Índice de Subconjunto	Descrição
15	Para páginas bloqueadas em memória (opcional)
14	Páginas acessadas em 40 + períodos
13	Páginas acessadas em 40 + períodos porém não usadas por 4 + horas
12	Páginas pré-buscadas por gerenciamento de memória

	proativa
11	Páginas acessadas em 20-40 períodos
10	Páginas acessadas em 10-20 períodos
9	Novas páginas de conjunto de página inativa com dados insuficientes
8	Páginas acessadas em ≤ 4 períodos, porém não usadas nos últimos 5 minutos
7	Páginas acessadas em ≤ 8 períodos, porém não usadas nos últimos 10 minutos
6	Páginas acessadas em ≤ 16 períodos, porém não usadas nos últimos 20 minutos
5	Páginas em agrupamento pré-buscadas por gerenciamento de memória proativa
4	Páginas acessadas em ≤ 2 períodos, porém não usadas uma hora antes
3	Páginas acessadas em ≤ 2 períodos, porém não usadas nas últimas 4 horas
2	Páginas acessadas em ≤ 16 períodos, porém não usadas no dia anterior
1	Páginas acessadas em ≤ 2 períodos, porém não usadas no dia anterior
0	Páginas não acessadas nos últimos 3 dias

[0095] Ademais, as páginas de conjunto de trabalho que não forem usadas por vinte minutos são expulsas da memória de conjunto de trabalho 450 dentro do conjunto de página inativa 502 para torná-las disponíveis para gerenciamento priorizado. Conforme representado na Figura 4B, permite-se que um gerenciador de conjunto de trabalho 452 gerencie as páginas que são recentemente usadas, como em sistemas de tempo compartilhado tradicionais, visto que o gerenciamento de conjunto de página inativa através do re-balanceador de memória

424 gerencia as páginas por períodos mais longos de tempo.

[0096] Conceitualmente, como indicado pela Tabela 1, nesta implementação exemplificativa, o sistema de gerenciamento de subconjunto utiliza categorias baseadas no último tempo de uso e baseadas na frequência para gerenciamento de conjunto de página, referido como uma peneira de tempo e escada de re-utilização, respectivamente. O conceito de peneira de tempo (como concretizado nos subconjuntos 502₀-502₈) e escada de re-utilização (como concretizado nos subconjuntos 502₁₀-502₁₄) é que as páginas que não foram recentemente usadas devem ser re-propostas a partir do conjunto de página inativa 502 antes de outras páginas, porém aquelas proteções extras devem ser as páginas produzidas que são usadas frequentemente, mesmo que não necessariamente muito recente. As páginas no subconjunto de página inativa 502 são desta maneira organizadas como uma série de subconjuntos, e aquelas páginas serão periodicamente movidas a partir de subconjuntos protegidos até subconjuntos menos protegidos, ou vice-versa, baseado na falta de uso ou uso durante o tempo das mesmas. Assim, ao assumir que posteriormente não há necessidade de uma página não-usada, a página não-usada irá mover-se a partir de um subconjunto "não usado em 5 minutos" até um subconjunto "não usado em 10 minutos", até um subconjunto "não usado em 30 minutos", e assim por diante, até este chegar no subconjunto "não usado em 3 dias" (ou que for menor). As páginas que foram usadas um tanto de forma frequente com relação a algum intervalo razoável, entretanto, irão pular a parte inicial da cascata, ou podem ser mantidas em algum subconjunto por um período de tempo maior.

[0097] O que as categorias de peneira de tempo e escada de re-utilização realizam é separar as páginas que foram usadas somente por um breve período de tempo das páginas que foram usadas com mais frequência, e então priorizam estas páginas dentro de subconjun-

tos de respectivas categorias das mesmas. A peneira de tempo opera em um gradiente recentemente menos usado, enquanto a escada de re-utilização opera em um gradiente baseado na frequência. Naturalmente, os gradientes não precisam ser limitados a nenhuma frequência ou duração, e de fato, podem variar baseados em condições reais de modo que as páginas sejam movidas entre os subconjuntos e removidas do conjunto de página inativa à medida em que apropriadas para o dado sistema.

[0098] Nota-se que as páginas que são lidas a partir do disco pela primeira vez têm de ser colocadas em subconjuntos do conjunto de página inativa 502. As classes de páginas incluem páginas pré-buscadas (subconjunto 502₁₂), páginas em agrupamento com falhas comuns (subconjunto 502₉) e páginas interpoladas dentro de pré-buscas através de agrupamento (subconjunto 502₅). Estas classes de páginas são priorizadas nesta ordem, com páginas pré-buscadas colocadas no meio de páginas de escada de re-utilização, as páginas pré-buscadas interpoladas colocadas no meio de páginas de peneira de tempo e as páginas de falha comum colocadas no meio de páginas de peneira de tempo e escada de re-utilização.

[0099] Em suma, as páginas de maior utilidade desejadas em memória são trazidas para dentro de subconjuntos inativos para protegê-las da pressão normal de memória, que resulta em melhora significativa de desempenho com decisões apropriadas na identificação e priorização de cenários de chave. Por exemplo, em uma compreensividade consistente de sistema MB 256 foi proporcionado em muitas operações comuns na camada shell, os componentes e aplicativos, mesmo após as transições tais como partida, troca rápida de usuário e pressão de memória a partir de um aplicativo amplo. Um rápido recomeço a partir da inatividade, (por exemplo, inferior a dois segundos), sem atrasos significativos de I/O de discos, foi consistentemente atin-

gido. Ademais, com estes mecanismos, uma política de auto-sintonização e um conjunto de regras que podem processar de forma contínua as informações de cenário podem ser implementados para determinar quais as páginas devem estar em memória.

[00100] Conforme compreendido a partir da descrição acima, as melhoras de eficiência da pré-busca de maneira tempestiva não podem proporcionar experiência quase instantânea ao usuário, devido ao tamanho do desvio de dados requeridos e um grande número de posições de disco que precisam ser acessadas. Em vez de, proporcionar uma experiência significativamente melhorada, a presente invenção reconhece que as páginas para cenários frequentes e favoritos do usuário precisam ser antecipadamente trazidas para dentro da memória. Desta maneira, uma vez que o gerenciador de programa foi identificado um conjunto de páginas que devem ser trazidas para dentro da memória, estas páginas devem ser pré-buscadas, ao utilizar os períodos de inatividade do disco. Entretanto, é um recurso altamente exaustivo que os enfileirar um grande número (por exemplo, centenas) de I/Os de paginação assíncrona especulativa; por exemplo, se um aplicativo de primeiro plano precisar de algo do disco naquele momento, o pedido do aplicativo será bloqueado na fila de disco, que resulta potencialmente em tempos muito lentos (por exemplo, segundos) de incompreensibilidade. Somente a detecção de inatividade não é confiável para evitar tal situação, uma vez que o aplicativo de usuário possa sempre tirar a falha exatamente quando decide-se enfileirar as I/Os especulativas para o que mostra ser um disco inativo.

[00101] Conforme descrito acima, a presente invenção evita este problema ao proporcionar suporte para I/O priorizada, como se a pré-busca especulativa for distribuída para uma fila de disco de baixa prioridade, não irá causar imensos atrasos para pedidos de paginação a partir do aplicativo de primeiro plano que são enfileirados em uma prio-

ridade maior.

[00102] Além disso, se as I/Os de alta prioridade forem distribuídas uma de cada vez, estas podem ser intercaladas com I/Os de pré-busca de baixa prioridade. Os padrões resultantes de buscas irão impactar de forma significativa a largura de banda do disco e prejudicar os tempos de resposta. Como resultado, para limitar de fato o overhead de I/O de pré-busca especulativa, o suporte para I/O priorizada pode precisar ser combinado com detecção de inatividade rápida e de auto-sintonização que aprende a partir destas falhas e padrões de troca. Quando o suporte I/O priorizado for combinado com os mecanismos para identificar e priorizar a memória referenciada, o sistema torna-se capaz de proteger a operação de um processo a partir de interferência através de outros processos do sistema. Tradicionalmente isto foi experimentado ao trocar a prioridade da CPU de processo, porém esta priorização não foi transportada para filas de disco ou gerenciamento de memória, pelo menos não em sistemas operacionais de propósito geral.

[00103] Um sistema de restrições é geralmente representado na Figura 6, onde um quadro de restrições 602 restringe as atividades de segundo plano tal como scanners anti-vírus, indexadores de arquivo e programas de procedimento de identificação, para limitar a interferência nos aplicativos do usuário. O quadro de restrições 602 também pode proteger os aplicativos de mídia a partir de disco e interferência de memória, que é uma causa para problemas de fluxo de vídeo e áudio, bem como garantir a compreensividade interativa aos aplicativos de primeiro plano mesmo sob cargas pesadas de sistema.

[00104] A integração de aperfeiçoamentos de gerenciamento de memória com suporte de I/O priorizado proporciona uma base para um sistema altamente responsivo e prognosticável. Nota-se que o quadro de restrições 602 por si só não proporciona grandes garantias de re-

curso finamente parceladas (por exemplo, abaixo de 100 ms) que um gerenciador de recurso sofisticado pode proporcionar, porém limita os efeitos negativos de interferência para tempos de resposta, tal que o sistema irá sempre mostrar-se responsivo.

[00105] Conforme descrito acima, a pré-busca é um mecanismo com o qual um conjunto de páginas pode ser lido de forma eficiente a partir do disco dentro da memória e colocado em um subconjunto inativo particular. A pré-busca funciona ao enfileirar as I/Os de paginação assíncrona para o conjunto de páginas que estão faltando na memória. Na presente invenção, a pré-busca de páginas de diretório e de outros metadados de arquivo, bem como páginas de arquivo de página, é sustentada. Com o mecanismo de pré-busca aperfeiçoado, a presente invenção pode trazer para dentro da memória de forma eficiente qualquer página identificada nos rastreamentos, que incluem uma página de arquivo, uma página (pilha, conjunto de núcleo) privada em arquivo de página, página de registro, ou metadados de sistema de arquivo tal como (tabela de arquivo mestre) MFT ou diretório.

[00106] Como pode ser visto a partir do precedente, os diversos mecanismos de gerenciamento de memória proativa proporcionam uma experiência melhor ao usuário para máquinas novas com amplas memórias, que incluem dispositivos móveis tais como mesas gráficas, bem como computadores de mesa e portáteis, ao garantir que os dados estarão antecipadamente quase sempre em memória. Os métodos e mecanismos são proativos, lidam com gerenciamento de memória global (tanto em memória como em disco) ao obter ricas informações, e gerenciar a memória em todos os níveis de uso. A eficiência de I/O é também proporcionada, juntamente com categorias de I/O, para tentar otimizar as transferências de I/O ao mesmo tempo em que minimiza a interferência. Um conjunto de página ativa, gerenciado de forma mais inteligente, facilita a proteção de páginas mais valiosas com relação às

páginas menos valiosas, também fatorando no conceito de estimação de utilidade. O rastreamento determina um uso de memória melhor, juntamente com contextos, que incluem contextos definidos de usuário, definidos de aplicativo e definidos de sistema. Os métodos e sistemas de gerenciamento de memória abrangentes da presente invenção proporcionam desta maneira vantagens e benefícios significativos necessários em computação contemporânea.

[00107] Ao mesmo tempo em que a invenção é suscetível a diversas modificações e construções alternativas, certas modalidades ilustrativas da mesma são mostradas nos desenhos e descritas acima em detalhes. Deve-se entender, entretanto, que não há intenção de limitar a invenção às formas específicas mostradas, porém ao contrário, a intenção é cobrir todas as modificações, construções alternativas, e equivalentes que estão dentro do espírito e escopo da invenção.

REIVINDICAÇÕES

1. Método implementado por computador compreendendo as etapas de:

capturar informações ricas sobre uso de memória que incluem pelo menos um dentre dados de padrão, frequência, contexto, e agrupamento, e

persistir as informações ricas dentro de um arquivo,

caracterizado pelo fato de que ainda compreende as etapas de:

receber um evento com relação a um contexto; e

priorizar a memória em resposta ao evento, de modo que as páginas mais valiosas prováveis de serem usadas no contexto sejam maiores em valor com relação às páginas menos prováveis de serem usadas.

2. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** capturar as informações ricas compreende amostragem real de uso de página.

3. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** capturar as informações ricas compreende a etapa de rastrear uso real de página.

4. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** priorizar a memória compreende a etapa de alterar um subconjunto para o qual uma página de memória é classificada.

5. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** priorizar a memória compreende a etapa de carregar pelo menos uma página a partir do armazenamento secundário na memória.

6. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** priorizar a memória compreende a etapa de substituir as páginas na memória por páginas do armazenamento secundário.

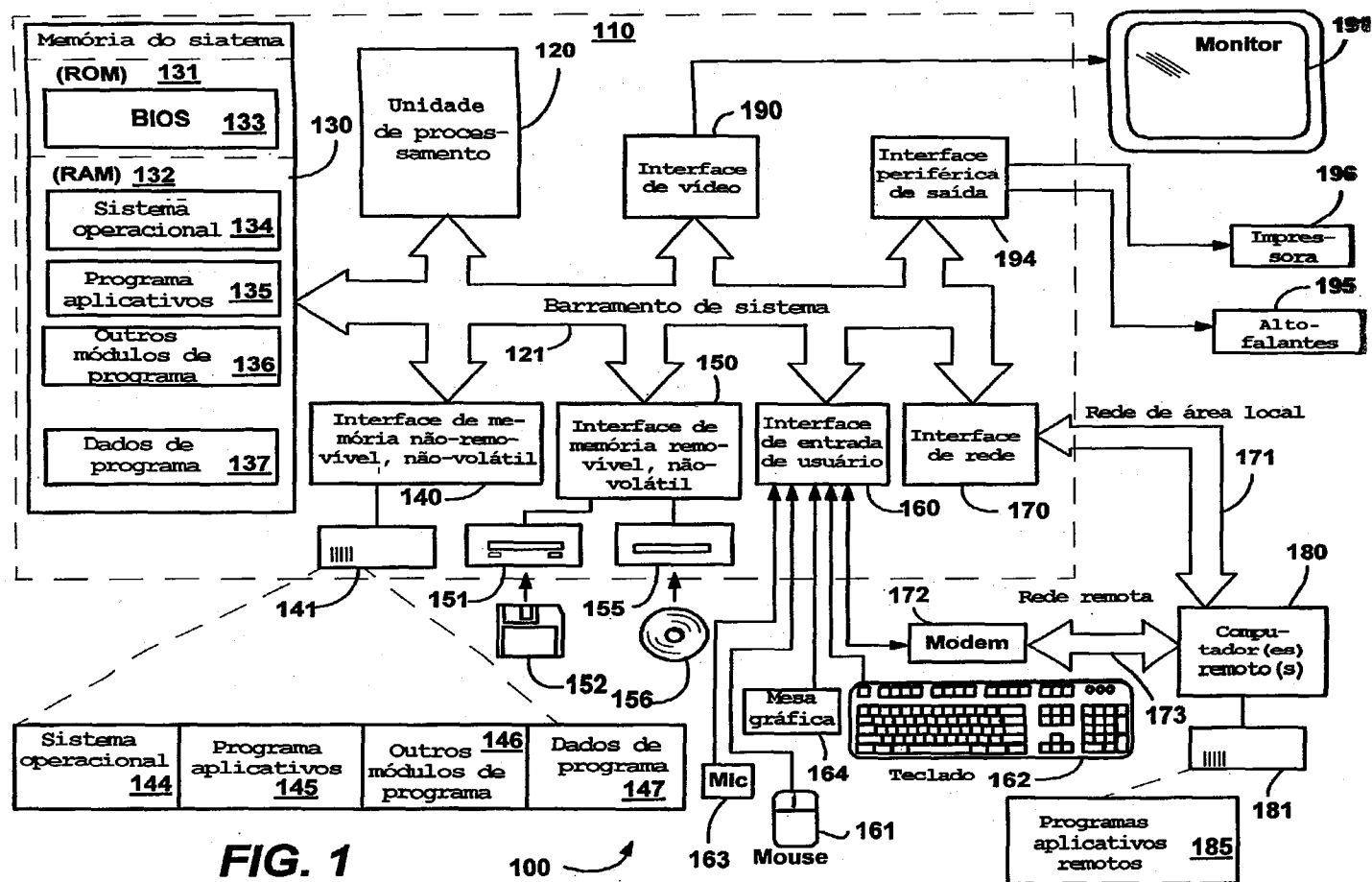


FIG. 1

100

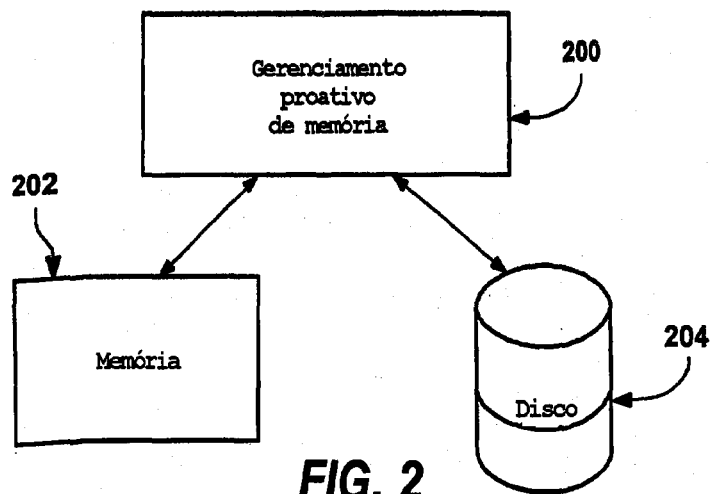


FIG. 2

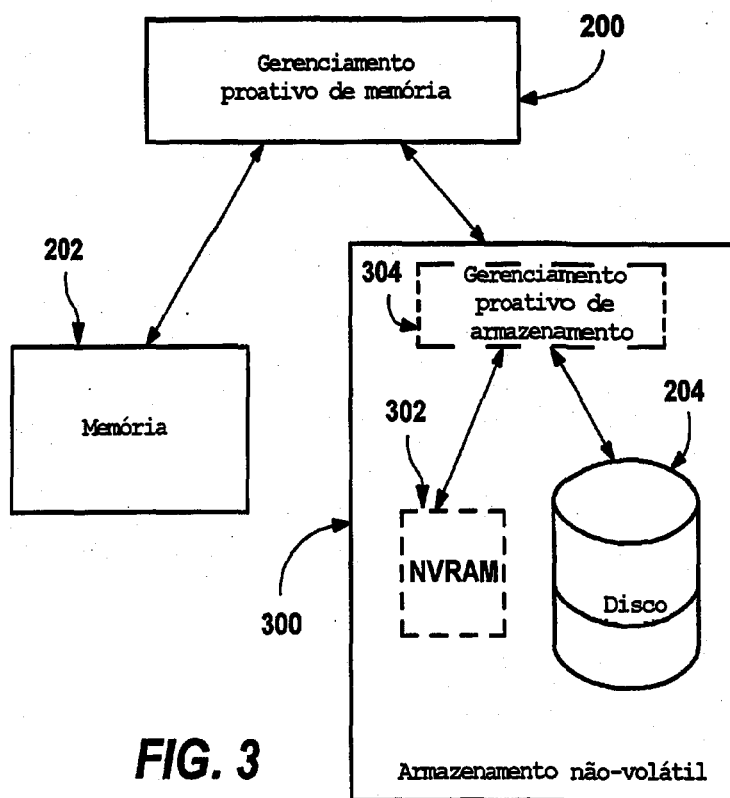


FIG. 3

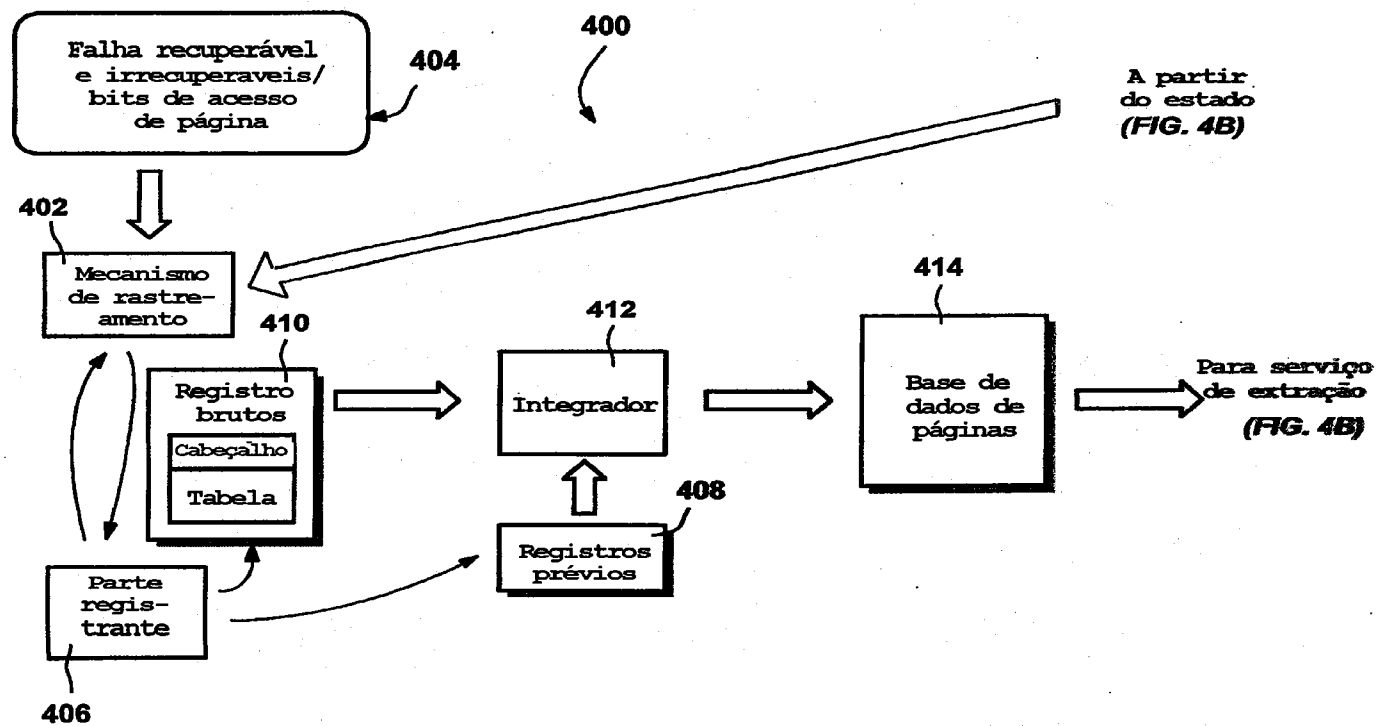


FIG. 4A

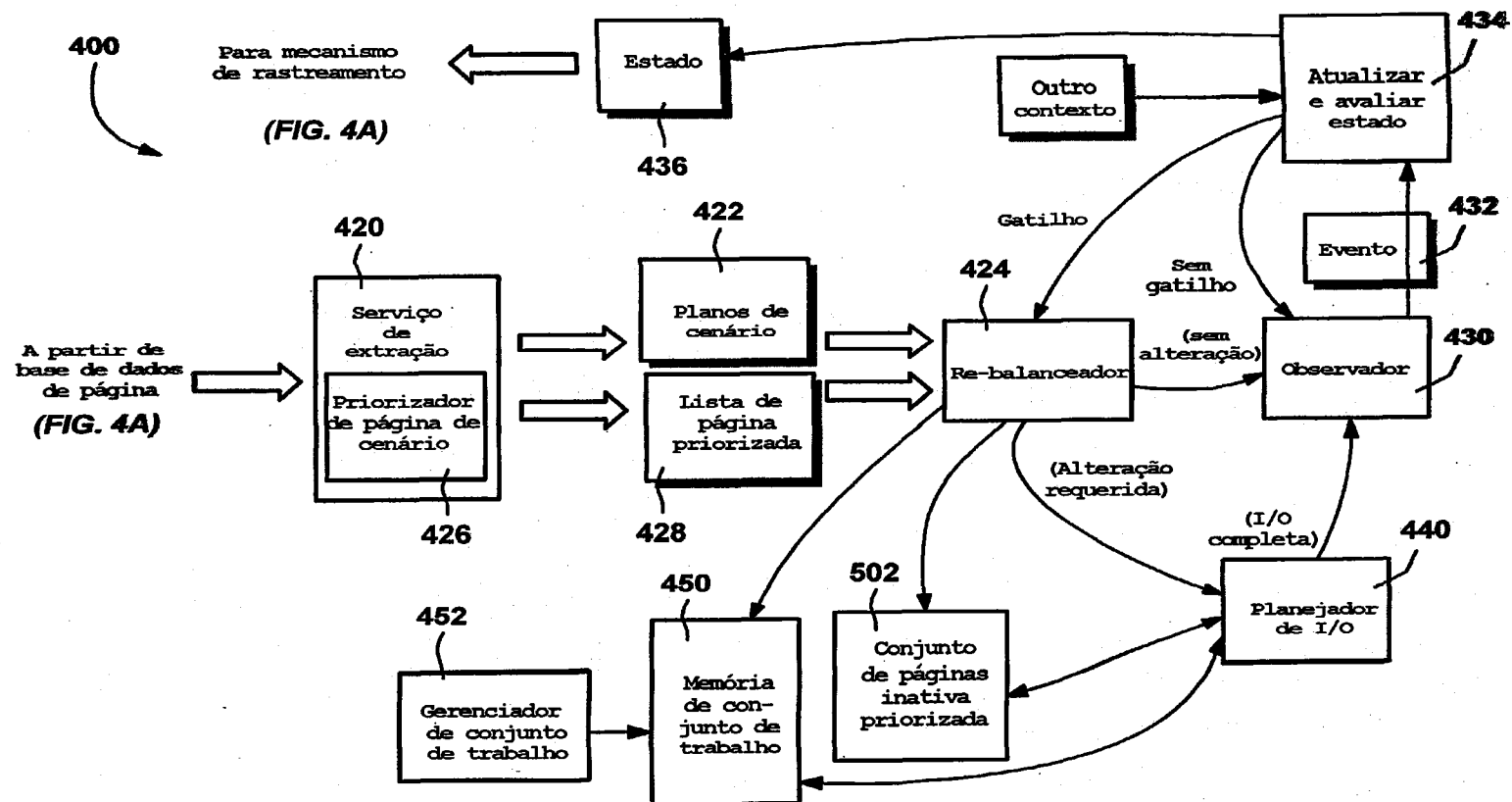


FIG. 4B

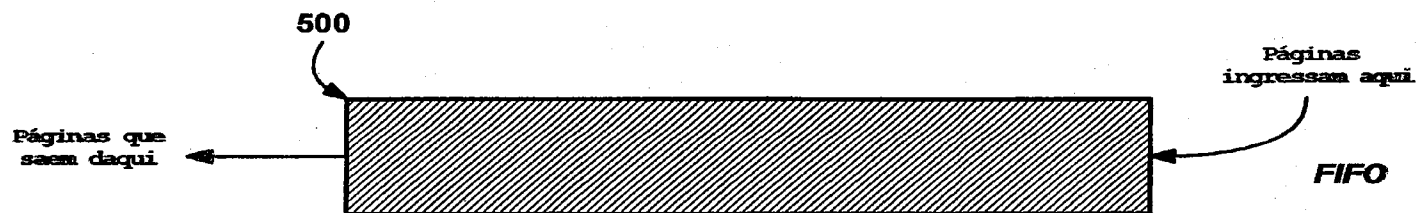


FIG. 5A

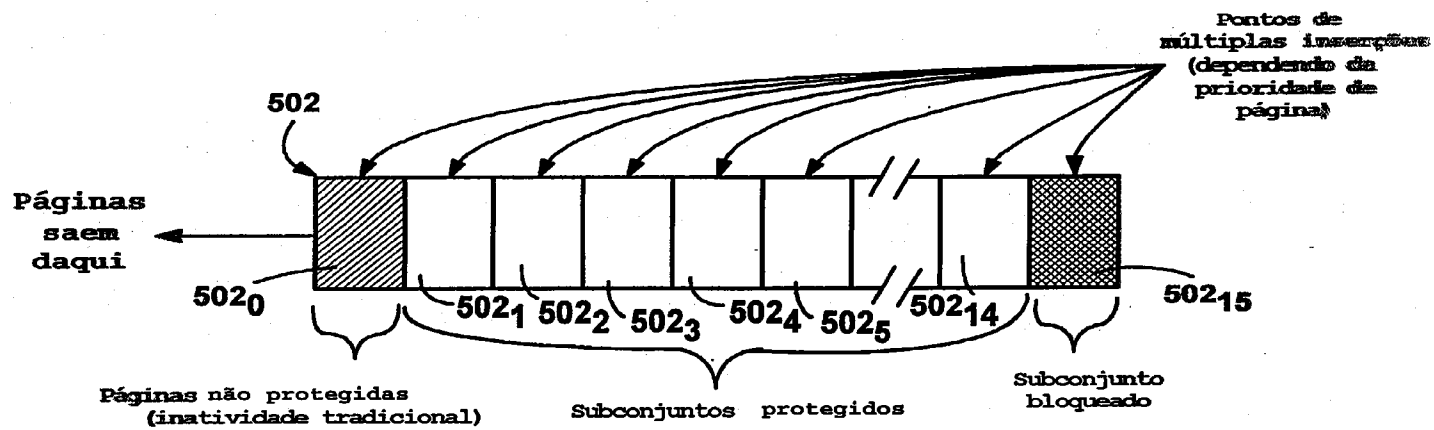


FIG. 5B

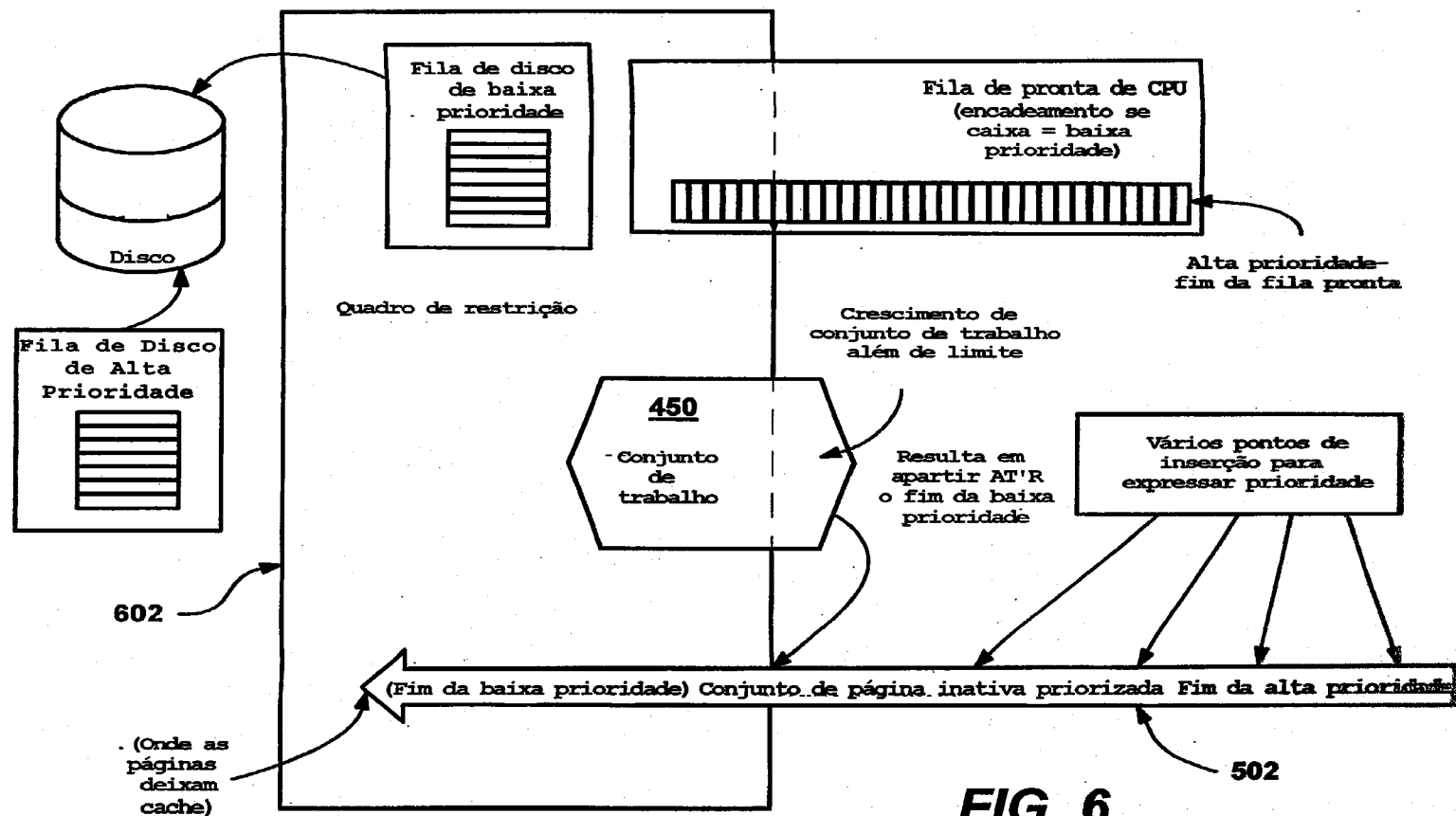


FIG. 6