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

(21) PI 0718463-8 A2

* B R P I 0 7 1 8 4 6 3 A 2 *

(22) Data de Depósito: 17/11/2007
(43) Data da Publicação: 10/12/2013
(RPI 2240)

(51) Int.Cl.:
G06F 9/06

(54) Título: ORDEM DE COMPROMETIMENTO DE TRANSAÇÃO DE SOFTWARE E GERENCIAMENTO DE CONFLITO

(57) Resumo:

(30) Prioridade Unionista: 17/11/2006 US 11/601.541

(73) Titular(es): Microsoft Corporation

(72) Inventor(es): David Detlefs, Goetz Graefe, John Joseph Duffy, Lingli Zhang, Michael M. Magruder, Vinod K. Grover

(74) Procurador(es): Nellie Anne Daniel-Shores

(86) Pedido Internacional: PCT US2007085035 de 17/11/2007

(87) Publicação Internacional: WO 2008/064139de 29/05/2008

"ORDEM DE COMPROMETIMENTO DE TRANSAÇÃO DE SOFTWARE E GERENCIAMENTO DE CONFLITO"

ANTECEDENTES

Memória transacional de software (STM) é um mecanismo de controle de concomitância, análogo às transações de banco de dados para controlar acesso à memória compartilhada em computação simultânea. Uma transação no contexto de memória transacional é uma peça de código que executa uma série de leituras e gravações para memória compartilhada. STM é usada como uma alternativa para os mecanismos de bloqueio, tradicionais. Os programadores colocam uma anotação declaratória (por exemplo, minúscula) em torno de um bloco de código para indicar propriedades de segurança que eles exigem e o sistema automaticamente garante que esse bloco execute diminutamente com relação às outras regiões de código, protegidas. O modelo de programação de memória transacional de software impede inversão de prioridade baseada em bloqueio, e problemas de impasse.

Embora os sistemas STM típicos tenham muitas vantagens, eles ainda requerem que o programador seja cuidadoso em evitar ordenamentos de acesso à memória, involuntários. Por exemplo, a ordem na qual as transações são comprometidas (isto é, processamento de comprometimento) em um ambiente STM, típico, é irrestrita. As transações competem umas com as outras para comprometimento, significando que se a transação 1 se compromete antes ou após a transação 2, frequentemente é um produto da programação dinâmica do programa (e frequentemente também mediante lógica de programa específico). Além disso, se duas transações conflitam, tal como mediante tentativa de gravar no mesmo fragmento de memória, então a ordem de comprometimento das mesmas pode ser decidida arbitrariamente com base em uma de muitas políticas possíveis de gerenciamento de disputa. Em ambos os cenários, nenhuma ordem de comprometimento específica é garantida; portanto a responsabilidade de garantir que o programa funcione corretamente, em qualquer ordem, é do programador. Isso torna a programação paralela muito difícil.

Uma abordagem para simplificar a programação paralela é a de automaticamente paralelizar os programas sequenciais, de uma maneira que garanta que a semântica do programa seja inalterada. Em outras palavras, se o programa sequencial funciona corretamente, da mesma forma funciona a versão paralelizada. Duas (separadas) variações desse conceito para paralelizar programas sequenciais foram denominadas, respectivamente, futuros seguros e paralelização de loop especulativo. Em futuros seguros, a versão sequencial de um programa poderia realizar "A; B" (isto é, faça A então faça B). O programador pode adicionar uma anotação (um "futuro") indicando que ele acha que poderia ser possível realizar A e B em paralelo sem mudar a semântica do programa – que A não lê quaisquer locais de memória que B lê, nem vice-versa. Porém o sistema trata isso estritamente como uma "sugestão" cuja validade deve ser verificada. Ele executa A e B como transações, e se elas

conflitarem, ele impede que B seja comprometida se ele fosse serializado antes de A. Esse é um aspecto "indesejável" de ordem de comprometimento indeterminada referida acima.

Paralelização de loop especulativo é uma idéia similar, onde as ações realizadas no programa sequencial são as iterações sucessivas de um loop. O programador (ou alguma análise estática) indica que pode ser vantajoso executar o loop em paralelo, e o sistema executa cada iteração do loop como uma transação em paralelo, exigindo que essas transações sejam comprometidas na ordem em que as iterações teriam sido comprometidas no programa original.

SUMÁRIO

Várias tecnologias e técnicas são reveladas para aplicação de ordenamento para as transações em um sistema de memória transacional de software. O sistema de memória transacional de software é provido com um recurso para permitir que uma ordem de comprometimento predeterminada seja especificada para uma pluralidade de transações. A ordem de comprometimento predeterminada é usada no tempo de execução para auxiliar na determinação de uma ordem de comprometimento das transações no sistema de memória transacional de software. Em uma implementação, a ordem de comprometimento predeterminada pode ser ou ordenamento total ou ordenamento parcial. No caso de ordenamento total, as transações são forçadas a comprometimento em uma ordem linear. No caso de ordenamento parcial, as transações podem ser comprometidas em um de múltiplos cenários aceitáveis. Em uma implementação, um arbitrador de comprometimento monitora o valor de próxima-para-comprometimento representando a transação que deve ter permissão para comprometimento a seguir, e quando uma transação específica está pronta para comprometimento, ela pode fazer isso se o seu número de ordem de comprometimento coincidir com o valor de próxima-para-comprometimento do arbitrador de comprometimento.

Um processo de gerenciamento de disputa é invocado quando ocorre um conflito entre uma primeira transação e uma segunda transação. A ordem de comprometimento predeterminada é usada no processo de gerenciamento de disputa para auxiliar na determinação de se a primeira transação ou a segunda transação deve vencer o conflito e ter permissão para prosseguir.

Esse Sumário foi provido para introduzir uma seleção de conceitos em uma forma simplificada que é descrita adicionalmente abaixo na Descrição Detalhada. Esse Sumário não pretende identificar aspectos fundamentais ou aspectos essenciais da matéria em estudo reivindicada nem pretende ser usado como um meio auxiliar na determinação do escopo da matéria em estudo reivindicada.

DESCRIÇÃO RESUMIDA DOS DESENHOS

A Figura 1 é uma vista diagramática de um sistema de computador de uma implementação.

A Figura 2 é uma vista diagramática de uma aplicação de memória transacional de software de uma implementação operando no sistema de computador da Figura 1.

A Figura 3 é um diagrama de fluxo de processo de alto nível para uma implementação do sistema da Figura 1.

A Figura 4 é um diagrama de fluxo de processo para uma implementação do sistema da Figura 1 ilustrando os estágios envolvidos no uso de um arbitrador de comprometimento para fazer cumprir uma ordem de comprometimento, predeterminada.

A Figura 5 é um diagrama de fluxo de processo para uma implementação do sistema da Figura 1 ilustrando os estágios envolvidos no uso de um arbitrador de comprometimento para fazer cumprir um ordenamento total de uma pluralidade de transações.

A Figura 6 é um diagrama de fluxo de processo para uma implementação do sistema da Figura 1 ilustrando os estágios envolvidos no uso de um arbitrador de comprometimento para fazer cumprir um ordenamento parcial de uma pluralidade de transações.

A Figura 7 é um fluxo de processo para uma implementação do sistema da Figura 1 que ilustra os estágios envolvidos na provisão de um processo de gerenciamento de contenção que gerencia conflitos utilizando a informação de ordem de comprometimento predeterminada.

A Figura 8 é um fluxo de processo para uma implementação do sistema da Figura 1 que ilustra os estágios envolvidos na provisão de um processo de gerenciamento de contenção que gerencia os conflitos com transações embutidas utilizando a informação de ordem de comprometimento, predeterminada.

A Figura 9 é um diagrama lógico ilustrando uma árvore de predecessores, exemplar, com predecessores de nível superior que têm um predecessor comum.

A Figura 10 é um diagrama lógico ilustrando uma árvore de predecessores, exemplar, com predecessores de nível superior que não têm um predecessor comum.

A Figura 11 é um diagrama de fluxo de processo para uma implementação do sistema da Figura 1 que ilustra os estágios envolvidos na redução de uma quantidade de trabalho desperdiçado mediante uso de um arbitrador de comprometimento em um sistema de memória transacional de software.

A Figura 12 é um diagrama de fluxo de processo para uma implementação do sistema da Figura 1 que ilustra os estágios envolvidos na análise de uma cadeia inteira de predecessores em um processo de gerenciamento de disputa para determinar a resolução adequada do conflito.

DESCRIÇÃO DETALHADA

Com a finalidade de promover um entendimento dos princípios da invenção, será feita agora referência às modalidades ilustradas nos desenhos e linguagem específica será utilizada para descrever as mesmas. Não obstante será entendido que não se pretende

qualquer limitação do escopo. Quaisquer alterações e modificações adicionais nas modalidades descritas, e quaisquer aplicações adicionais dos princípios conforme aqui descritos são consideradas como normalmente ocorreria àqueles versados na técnica.

O sistema pode ser descrito no contexto geral como um sistema de memória transacional de software, porém o sistema também serve a outros propósitos além desses. Em uma implementação, uma ou mais das técnicas aqui descritas podem ser implementadas como características dentro de um programa de estrutura tal como MICROSOFT®.NET Framework, ou a partir de qualquer outro tipo de programa ou serviço que proporcione plataformas para desenvolvedores para desenvolver aplicações de software. Em outra implementação, uma ou mais das técnicas aqui descritas são implementadas como características com outras aplicações que lidam com o desenvolvimento de aplicações que executam em ambientes simultâneos.

Um recurso é provido no sistema de memória transacional de software para permitir que uma ordem predeterminada de comprometimento seja especificada para uma pluralidade de transações. A ordem predeterminada de comprometimento é usada para auxiliar na determinação de uma ordem para comprometimento das transações. Em uma implementação, um processo de gerenciamento de disputa é invocado quando ocorre um conflito entre uma primeira transação e uma segunda transação. A ordem predeterminada de comprometimento é então usada no processo de gerenciamento de contenção para auxiliar na determinação no sentido de se a primeira transação ou a segunda transação deve vencer o conflito e ter permissão para prosseguir.

Conforme mostrado na Figura 1, um sistema de computador exemplar a ser usado para implementar uma ou mais partes do sistema inclui um dispositivo de computação, tal como o dispositivo de computação 100. Em sua configuração mais básica, o dispositivo de computação 100 inclui tipicamente ao menos uma unidade de processamento 102 e memória 104. Dependendo da configuração exata e do tipo de dispositivo de computação, a memória 104 pode ser volátil (tal como RAM), não-volátil (tal como ROM, memória flash, etc.) ou alguma combinação das duas. Essa configuração mais básica é ilustrada na Figura 1 por intermédio da linha tracejada 106.

Adicionalmente, o dispositivo 100 também pode ter características/funcionalidade, adicionais. Por exemplo, o dispositivo 100 também pode incluir meio de armazenamento adicional (removível e/ou não-removível) incluindo, mas não limitado aos discos, ou fita, magnéticos ou óticos. Tal meio de armazenamento adicional é ilustrado na Figura 1 por intermédio do meio de armazenamento removível 108 e meio de armazenamento não-removível 110. Meios de armazenamento de computador incluem meios voláteis e não-voláteis, removíveis e não-removíveis, implementados em qualquer método ou tecnologia para armazenamento de informação tal como instruções legíveis por computador, estruturas

de dados, módulos de programa ou outros dados. A memória 104, o meio de armazenamento removível 108 e o meio de armazenamento não-removível 110 são, todos, exemplos de meios de armazenamento de computador. Os meios de armazenamento de computador incluem, mas não são limitados a: RAM, ROM, EEPROM, memória flash ou outra tecnologia de memória, CD-ROM, discos digitais versáteis (DVD) ou outro meio de armazenamento ótico, cassetes magnéticos, fita magnética, meio de armazenamento de disco magnético ou outros dispositivos de armazenamento magnético, ou qualquer outro meio que possa ser usado para armazenar a informação desejada e que possa ser acessado pelo dispositivo 100. Quaisquer tais meios de armazenamento de computador podem ser parte do dispositivo 100.

O dispositivo de computação 100 inclui uma ou mais conexões de comunicação 114 que permitem que o dispositivo de computação 100 se comunique com outros computadores/aplicações 115. O dispositivo 100 também pode ter dispositivo(s) de entrada 112 tal como teclado, mouse, caneta gráfica, dispositivo de entrada de voz, dispositivo de entrada de toque, etc. O dispositivo(s) de saída 111 tal como um vídeo, alto-falantes, impressora, etc. também pode ser incluído. Esses dispositivos são bem conhecidos na técnica e não precisam ser discutidos aqui detalhadamente. Em uma implementação, o dispositivo de computação 100 inclui aplicação de memória transacional de software 200. A aplicação de memória transacional de software 200 será descrita em detalhe adicional na Figura 2.

Voltando-se agora para a Figura 2 com referência continuada à Figura 1, é ilustrada uma aplicação de memória transacional de software 200 operando no dispositivo de computação 100. A aplicação de memória transacional de software 200 é um dos programas de aplicação que residem no dispositivo de computação 100. Contudo, será entendido que a aplicação de memória transacional de software 200 pode ser incorporada alternativamente ou adicionalmente como instruções executáveis por computador em um ou mais computadores e/ou em diferentes variações do que apresentado na Figura 1. Alternativamente ou adicionalmente, uma ou mais partes da aplicação de memória transacional de software 200 pode ser parte da memória de sistema 104, ou outros computadores e/ou aplicações 115, ou outras tais variações como ocorreria àqueles na técnica de software de computador.

A aplicação de memória transacional de software 200 inclui lógica de programa 204, a qual é responsável pela realização de algumas ou de todas as técnicas aqui descritas. A lógica de programa 204 inclui lógica para prover um sistema de memória transacional de software (STM) 206; lógica para prover um arbitrador de comprometimento que permite que uma ordem predeterminada de comprometimentos seja especificada, estaticamente ou dinamicamente, para uma pluralidade de transações no sistema STM 208; lógica para permitir que o arbitrador de comprometimento utilize a ordem predeterminada de comprometimento em tempo de execução para auxiliar na determinação de uma ordem na qual com-

prometer a pluralidade de transações no sistema de memória transacional de software 210; lógica para prover um processo de gerenciamento de disputa que é invocado quando ocorrer um conflito entre uma primeira transação e uma segunda transação 212; lógica para usar a ordem de comprometimento predeterminada no processo de gerenciamento de disputa para auxiliar a determinar se a primeira transação ou a segunda transação deve vencer o conflito e ter permissão para prosseguir (por exemplo, dependendo de qual delas tem o número de ordem de comprometimento mais baixo de duas transações no mesmo grupo de transações) 214; lógica para permitir que o arbitrador de comprometimento seja operável para usar o ordenamento predeterminado de comprometimento para monitorar um ou mais valores de ordenamento (por exemplo, no ordenamento total – um campo próxima-para-comprometimento que representa uma próxima transação da pluralidade de transações que devem ter permissão para comprometimento) e para comparar o um ou mais valores de ordenamento com um número de ordenamento de comprometimento específico de uma determinada transação para verificar se o comprometimento da transação determinada tem o ordenamento apropriado que deve ser imposto 216; e outra lógica para operar a aplicação 220. Em uma implementação, a lógica de programa 204 é operável para ser ativada programaticamente a partir de outro programa, tal como utilizando uma única chamada para um procedimento na lógica de programa 204.

Voltando-se agora para as Figuras 3-10 com referência continuada às Figuras 1-2, os estágios para implementar uma ou mais implementações da aplicação de memória transacional de software 200 são descritos em detalhe adicional. A Figura 3 é um diagrama de fluxo de processo de alto nível para aplicação de memória transacional de software 200. Em uma forma, o processo da Figura 3 é ao menos parcialmente implementado na lógica de operação do dispositivo de computação 100. O procedimento começa no ponto de partida 240 com a provisão de um sistema de memória transacional de software (estágio 242). Um recurso é provido para permitir que uma ordem predeterminada de comprometimento (por exemplo, um ordenamento total ou ordenamento parcial) seja especificada para uma pluralidade de transações (por exemplo, atribuídas dinamicamente ou estaticamente) (estágio 244). O termo "ordem predeterminada de comprometimento" conforme aqui usado pretende incluir uma ordem específica na qual um grupo específico de transações relacionadas deve ser comprometido, conforme determinado em qualquer momento antes das transações começarem a executar. O termo "grupo" de transações conforme aqui usado inclui um conjunto específico (por exemplo, uma pluralidade) de transações gerenciadas pelo mesmo arbitrador de comprometimento, assim como filhos embutidos dessas transações.

A ordem predeterminada de comprometimento é usada no tempo de execução para auxiliar na determinação de uma ordem de comprometimento da pluralidade de transações no sistema de memória transacional de software (estágio 246). A ordem predeterminada de

comprometimento é usada para auxiliar a resolver conflitos ocorrendo entre duas ou mais das várias transações (estágio 248). O processo termina no ponto final 250.

A Figura 4 ilustra uma implementação dos estágios envolvidos no uso de um arbitrador de comprometimento para impor uma ordem predeterminada de comprometimento. Em uma forma, o processo da Figura 4 é implementado ao menos parcialmente na lógica de operação do dispositivo de computação 100. O procedimento começa no ponto inicial 270 com a provisão de um ou mais arbitradores de comprometimento para um sistema de memória de transação de software, o arbitrador de comprometimento sendo operável para permitir que uma ordem predeterminada de comprometimento seja especificada para uma pluralidade de transações (estágio 272). O termo "arbitrador de comprometimento" conforme aqui usado pretende incluir qualquer tipo de programa, recurso, ou processo que seja responsável pelo gerenciamento de um ou mais grupos de transações que devem ser ordenadas com relação umas às outras. Em uma implementação, pode haver um ou mais arbitradores de comprometimento ativos dentro de um programa em qualquer momento determinado. Por exemplo, quando tantos arbitradores de comprometimento quanto forem necessários puderem ser criados para gerenciar os diferentes grupos de transações. O arbitrador de comprometimento monitora e atualiza um ou mais valores de ordenamento que são usados para determinar o ordenamento adequado das transações com relação umas às outras (estágio 274). No caso de ordenamento total, um campo próxima-para-comprometimento pode ser usado para representar uma próxima transação de uma pluralidade de transações que devem ser comprometidas a seguir (estágio 274). No caso de ordenamento parcial, um gráfico dirigido de diferentes ordens possíveis é monitorado utilizando-se os valores de ordenamento. Conforme apropriado, o arbitrador de comprometimento utiliza a ordem predeterminada de comprometimento para prover um número de ordem de comprometimento para cada uma das várias transações (estágio 276).

Quando uma transação específica da pluralidade de transações se prepara para comprometimento, se o número de ordem de comprometimento para a transação específica, quando comparado com um ou mais valores de ordenamento, revelar que o comprometimento é apropriado, então o arbitrador de comprometimento permite que a transação seja comprometida (estágio 278). No caso de ordenamento total, esse cenário ocorre quando o campo C e o número de ordem de comprometimento para a transação específica tiverem o mesmo valor. Em tal cenário, o arbitrador de comprometimento permite que a transação seja comprometida e então incrementa o campo próxima-para-comprometimento para um próximo número em uma seqüência (por exemplo, próximo número mais alto) se o comprometimento for bem-sucedido (estágio 278). Quando a transação específica da pluralidade de transações se prepara para comprometimento, se o número de ordem de comprometimento para a transação específica quando comparado com os valores de ordenamento revelar que

o comprometimento não é adequado, então a transação específica é colocada no modo de retenção até que ele seja ativado em um momento posterior após uma transação predecessora ser comprometida (estágio 280). No caso de ordenamento total, entra-se nesse modo de retenção quando o campo próxima-para-comprometimento e o número de ordem para a transação específica não tiverem o mesmo valor.

Em uma implementação, o sistema pode ativar uma transação após sua predecessora imediata ter sido comprometida, em cujo caso ele pode tentar comprometer imediatamente. Alternativamente, o sistema pode optar por ativar uma transação após certa predecessora não imediata ter sido comprometida, embora sua predecessora imediata possa não ter sido ainda comprometida. Após ser ativado, o sistema verifica se é apropriado que a transação realmente seja comprometida. Se esse for o caso, a transação é comprometida. O processo termina no ponto final 282.

A Figura 5 ilustra uma implementação dos estágios envolvidos no uso de um arbitrador de comprometimento para impor um ordenamento total de uma pluralidade de transações. Em uma forma, o processo da Figura 5 é ao menos parcialmente implementado na lógica de operação do dispositivo de computação 100. O procedimento começa no ponto inicial 290 com a provisão de um ou mais arbitradores de comprometimento operáveis para permitir que um ordenamento predeterminado total seja especificado para uma pluralidade de transações (por exemplo, aquela especificando uma ordem exata na qual as várias transações devem ser comprometidas) (estágio 292). Quando uma transação específica das várias transações atinge seu ponto de comprometimento, para impor a ordem de comprometimento, a ordem de comprometimento da transação específica é comparada com um campo próxima-para-comprometimento do arbitrador de comprometimento (estágio 296). Em uma implementação, se o sistema determinar que imposição do ordenamento total não é necessária (por exemplo, tal como porque definitivamente não existe conflito), então a exigência de ordenamento total pode ser invalidada conforme apropriado (estágio 294), então o processo termina no ponto final 302.

Se o ordenamento de comprometimento deve ser imposto, e se a ordem de comprometimento da transação específica tiver um mesmo valor que o campo próxima-para-comprometimento do arbitrador de comprometimento (ponto de decisão 296), então a transação específica é comprometida, e se o comprometimento for bem-sucedido, o campo próxima-para-comprometimento é incrementado e o próximo sucessor é ativado, se existir (estágio 298). Se a ordem de comprometimento da transação específica não tiver o mesmo valor como o campo próxima-para-comprometimento do arbitrador de comprometimento (ponto de decisão 296), então a transação específica é colocada em um modo de retenção/espera até que ela seja ativada em um momento posterior após o comprometimento de uma transação predecessora (estágio 300). Em uma implementação, naquele momento pos-

terior, se ocorrer um conflito com um predecessor, pode ser solicitado que aquela transação específica seja abortada e retomada de tal modo que um predecessor pode realizar progresso avante. Caso contrário, se nenhum tal conflito tiver ocorrido, aquela transação específica deve ser capaz de comprometimento quando as exigências de ordem de comprometimento descritas aqui forem satisfeitas. O processo então termina no ponto final 302.

A Figura 6 ilustra uma implementação dos estágios envolvidos no uso de um arbitrador de comprometimento para impor um ordenamento parcial de uma pluralidade de transações. Em uma forma, o processo da Figura 6 é ao menos parcialmente implementado na lógica de operação do dispositivo de computação 100. O procedimento começa no ponto inicial 310 com a provisão de um ou mais arbitradores de comprometimento operáveis para permitir que um ordenamento predeterminado parcial seja especificado para uma pluralidade de transações (por exemplo, aquele especificando uma pluralidade de ordens aceitáveis nas quais as várias transações devem ser comprometidas – por exemplo, na forma de um gráfico dirigido) (estágio 312). Quando uma transação específica da pluralidade de transações atingir o seu ponto de comprometimento, para impor a ordem de comprometimento, o estado das transações predecessoras (por exemplo, um ou mais valores de ordenamento) é consultado para a transação de comprometimento específica (conforme monitorada pelo arbitrador de comprometimento) (estágio 314). Se todos os predecessores para a transação específica tiverem sido comprometidos (ponto de decisão 316), então a transação específica é comprometida (estágio 318). Se o comprometimento for bem-sucedido, um ou mais valores monitorados pelo arbitrador de comprometimento são atualizados conforme apropriado, e todos os possíveis próximos sucessores são ativados, se existir algum (estágio 318).

Se todos os predecessores para a transação específica não tiverem sido comprometidos (ponto de decisão 316), então a transação específica é colocada no modo de retenção/espera até que seja ativada em um momento posterior após o comprometimento uma transação predecessora (estágio 320). O processo termina no ponto final 322.

A Figura 7 ilustra uma implementação dos estágios envolvidos na provisão de um processo de gerenciamento de disputa que gerencia os conflitos utilizando a informação de ordem predeterminada de comprometimento. Em uma forma, o processo da Figura 7 é implementado ao menos parcialmente na lógica de operação do dispositivo de computação 100. O procedimento começa no ponto inicial 340 com a provisão de um sistema de memória transacional de software que suporta uma ordem predeterminada de comprometimento para um ou mais grupos de transações (estágio 342). Um processo de gerenciamento de contenção é provido o qual é invocado quando ocorre um conflito entre uma primeira transação e uma segunda transação (estágio 344). A ordem predeterminada de comprometimento é usada no processo de gerenciamento de contenção para auxiliar na determinação de se a primeira transação ou a segunda transação deve vencer o conflito e ter permissão para

prosseguir (estágio 346). Se a primeira transação e a segunda transação são parte do mesmo grupo de transação (ponto de decisão 348), então uma ordem predeterminada de comprometimento não é imposta entre essas duas transações (porque nenhuma delas existia) (estágio 350). Em tal cenário, como as duas transações não estão no mesmo grupo de transações, o fator de ordenamento não é usado para ajudar a resolver o conflito (estágio 350).

Se a primeira transação e a segunda transação são parte do mesmo grupo de transações (ponto de decisão 348), então o sistema compara o primeiro número de ordem da primeira transação e o segundo número de ordem da segunda transação (estágio 352). A transação com o número de ordem mais baixo tem permissão para prosseguir (ou com outro ordenamento adequado de prioridade) (estágio 354). O processo termina no ponto final 356.

A Figura 8 ilustra uma implementação dos estágios envolvidos na provisão de um processo de gerenciamento de contenção que gerencia conflitos com as transações embutidas utilizando a informação de ordem predeterminada de comprometimento. Em uma forma, o processo da Figura 8 é ao menos parcialmente implementado na lógica de operação do dispositivo de computação 100. Em uma implementação, a cadeia inteira de predecessores é considerada para cada transação antes de comprometer a transação específica, de modo que qualquer ordenamento presente naquela cadeia é imposto. O procedimento começa no ponto inicial 370 com a provisão de um processo de gerenciamento de disputa que é invocado quando ocorrer um conflito entre uma primeira transação e uma segunda transação (estágio 372). Uma ordem predeterminada de comprometimento é usada no processo de gerenciamento de disputa para auxiliar na determinação de se a primeira transação ou a segunda transação deve vencer o conflito e ter permissão para prosseguir (estágio 372). Se a primeira e a segunda transação não forem parte do mesmo grupo de transações (ponto de decisão 376), então uma ordem predeterminada de comprometimento não é imposta entre essas duas transações (porque nenhuma delas existia (estágio 378) e o processo termina no ponto final 388. Se a primeira e a segunda transação forem parte do mesmo grupo de transações (ponto de decisão 376), então o sistema verifica se as transações embutidas estão envolvidas (ponto de decisão 380).

Se transações embutidas não estiverem envolvidas (ponto de decisão 380), então o número de ordem (ou outro indicador de ordenamento) da primeira transação é comparado com o número de ordem (ou outro indicador de ordenamento) da segunda transação (estágio 384). A transação com o número de ordem inferior tem permissão para prosseguir (ou aquela determinada como sendo a próxima na ordem mediante uso de outros critérios adequados de ordenamento) (estágio 386).

Se transações embutidas estiverem envolvidas (ponto de decisão 380), então o número de ordem (ou outro indicador de ordenamento) do predecessor de nível superior da

primeira transação é comparado com o número de ordem (ou outro indicador de ordenamento) do predecessor de nível superior da segunda transação (estágio 382). O termo "predecessor de nível superior" conforme aqui usado pretende incluir os filhos imediatos de predecessores comuns onde os predecessores comuns estão envolvidos, e o predecessor de nível superior de cada transação onde não há predecessor comum envolvido. Esses cenários envolvendo predecessores comuns e não comuns são ilustrados em detalhe adicional nas Figuras 9 e 10. A transação com o número de ordem inferior tem permissão para prosseguir (por exemplo, a transação relacionada ao predecessor que tinha o número de ordem inferior ou outros critérios adequados) (estágio 386). O processo termina no ponto final 388.

A Figura 9 é um diagrama lógico ilustrando uma árvore de predecessores exemplar com predecessores de nível superior que têm um predecessor comum. No exemplo mostrado, a transação A é um predecessor comum de D e E. Em conflitos ocorrendo entre D e E, o número de ordem de transações B e C (os filhos imediatos de predecessores comuns A) é analisado para determinar qual transação D ou E deve ter permissão para prosseguir (estágio 382 na Figura 8).

A Figura 10 é um diagrama lógico ilustrando uma árvore de predecessor exemplar com predecessores de nível superior que não têm predecessores comuns. No exemplo mostrado, a transação A é uma predecessora da transação C. A transação D é uma predecessora da transação F. Em conflitos ocorrendo entre as transações C e F, então o número de ordem das transações A e D (o predecessor de nível superior de cada uma delas) são comparados para determinar qual transação C ou F deve ter permissão para prosseguir (estágio 382 na Figura 8).

A Figura 11 ilustra uma implementação dos estágios envolvidos na redução da quantidade de trabalho desperdiçado mediante uso de um arbitrador de comprometimento em um sistema de memória transacional de software. Em uma forma, o processo da Figura 11 é ao menos parcialmente implementado na lógica de operação do dispositivo de computação 100. O procedimento começa no ponto inicial 400 com a provisão de um ou mais arbitradores de comprometimento para um sistema de memória transacional de software, o arbitrador de comprometimento sendo operável para permitir que uma ordem predeterminada de comprometimento seja especificada para uma pluralidade de transações (estágio 402). O arbitrador de comprometimento é operável para colocar uma transação no modo de retenção/espera para bloquear aquela transação em termos de re-execução quando uma transação predecessora ainda está em execução (por exemplo, mediante análise da ordem predeterminada de comprometimento para determinar a ordem adequada (estágio 404)). O arbitrador de comprometimento também é operável para ativar as transações que foram colocadas em espera quando a transação(ões) predecessora concluiu (por exemplo, mediante análise outra vez de uma ordem predeterminada de comprometimento para determinar a

ordem adequada) (estágio 406). Mediante provisão desses mecanismos de bloqueio e de ativação, o arbitrador de comprometimento ajuda a reduzir a quantidade de trabalho que é desperdiçada mediante ação de impedir que sejam realizadas operações que teriam que ser posteriormente desfeitas (estágio 408). O processo termina no ponto final 410.

A Figura 12 ilustra uma implementação dos estágios envolvidos na análise de uma cadeia inteira de predecessores em um processo de gerenciamento de disputa para determinar a resolução de conflito adequada. Em uma forma, o processo da Figura 12 é implementado ao menos parcialmente na lógica de operação do dispositivo de computação 100. O procedimento começa no ponto inicial 430 com a provisão de um processo de gerenciamento de disputa que é invocado quando ocorrer um conflito entre uma primeira transação e uma segunda transação (estágio 432). Uma ordem predeterminada de comprometimento é usada no processo de gerenciamento de disputa para auxiliar a determinar se a primeira transação ou a segunda transação deve vencer o conflito e ter permissão para prosseguir (estágio 434). Uma cadeia inteira de predecessores de uma ordem predeterminada de comprometimentos é analisada para ajudar a determinar o gerenciamento de conflito adequado (estágio 436). Por exemplo, se houver quatro transações, dois pais e dois filhos, onde B está embutido dentro de A e D está embutido dentro de C. Suponha que exista uma relação de ordenamento entre A e C onde A deve ser comprometido antes de C. Se B e D conflitarem, o processo de gerenciamento de disputa deve favorecer B porque favorecer D é inútil visto que A deve ser comprometido antes de C (estágio 436). O processo termina no ponto final 438.

Embora os exemplos aqui discutidos sejam relacionados à imposição de ordenamento de comprometimento utilizando várias tecnologias e técnicas, deve ser observado que uma transação pode não ter absolutamente um arbitrador de comprometimento. Em tal caso em que uma transação não tem absolutamente um arbitrador de comprometimento, ocorrerá um comprometimento não-ordenado normal.

Embora a matéria em estudo tenha sido descrita em linguagem específica para características estruturais e/ou ações metodológicas, deve ser entendido que a matéria em questão definida nas reivindicações anexas não é necessariamente limitada aos aspectos ou ações específicas descritas acima. Mais propriamente, os aspectos e ações específicas descritas acima são revelados como formas exemplares de implementar as reivindicações. Todos os equivalentes, alterações, e modificações compreendidas no espírito das implementações conforme aqui descrito e/ou por intermédio das reivindicações a seguir devem ser protegidos.

Por exemplo, aqueles versados na técnica de software de computador reconhecerão que os arranjos de cliente e/ou servidor, conteúdo de tela de interface com o usuário, e/ou leiautes de dados conforme descritos nos exemplos aqui discutidos poderiam ser orga-

nizados de forma diferente em um ou mais computadores para incluir um número menor de opções ou características ou opções ou características adicionais do que retratado nos exemplos.

REIVINDICAÇÕES

1. Método para aplicar ordenamento nas transações em um sistema de memória transacional de software **CARACTERIZADO** por compreender as etapas de:

prover um sistema de memória transacional de software (242);

prover um recurso para permitir que uma ordem predeterminada de comprometimento seja especificada para uma pluralidade de transações (244); e

utilizar a ordem predeterminada de comprometimento em tempo de execução para auxiliar a determinar uma ordem na qual comprometer a pluralidade de transações no sistema de memória transacional de software (246).

2. Método, de acordo com a reivindicação 1, **CARACTERIZADO** por compreender ainda:

utilizar a ordem predeterminada de comprometimento para auxiliar a resolver os conflitos ocorrendo entre duas ou mais das várias transações (248).

3. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que a ordem predeterminada de comprometimento compreende um ordenamento predeterminado total da pluralidade de transações (244).

4. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que a ordem predeterminada de comprometimento compreende um ordenamento predeterminado parcial da pluralidade de transações (244).

5. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que um arbitrador de comprometimento é provido que monitora um ou mais valores de ordenamento para a ordem predeterminada de comprometimento (272).

6. Método, de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que quando uma transação específica da pluralidade de transações se prepara para comprometimento, comparar o um ou mais valores de ordenamento com um valor de ordenamento para a transação específica para determinar se a transação específica pode ser comprometida (278).

7. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que a ordem predeterminada de comprometimento é atribuída dinamicamente (244).

8. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que a ordem predeterminada de comprometimento é atribuída estaticamente (244).

9. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que um arbitrador de comprometimento é provido o qual monitora um campo próxima-para-comprometimento que representa uma próxima transação da pluralidade de transações que devem ter permissão para comprometimento (274).

10. Método, de acordo com a reivindicação 9, **CARACTERIZADO** pelo fato de que o número de ordem de comprometimento é provido para cada uma das várias transações

(276).

11. Método, de acordo com a reivindicação 10, **CARACTERIZADO** pelo fato de que quando uma transação específica das várias transações se prepara para comprometimento, determinar se o número de ordem de comprometimento para a transação específica tem o mesmo valor que o campo próxima-para-comprometimento monitorado pelo arbitrador de comprometimento (278).

12. Método, de acordo com a reivindicação 11, **CARACTERIZADO** pelo fato de que se o número de ordem de comprometimento para a transação específica e o campo próxima-para-comprometimento tiver o mesmo valor, permitir que o comprometimento prossiga (278).

13. Método, de acordo com a reivindicação 12, **CARACTERIZADO** pelo fato de que após o comprometimento prosseguir e ser bem-sucedido, o arbitrador de comprometimento incrementa o campo próxima-para-comprometimento para um próximo número em uma seqüência (278).

14. Método, de acordo com a reivindicação 11, **CARACTERIZADO** pelo fato de que se o número de ordem de comprometimento para transação específica e o campo próxima-para-comprometimento não tiverem o mesmo valor, colocar uma transação específica no modo de retenção até que ela seja ativada em um momento posterior após uma transação predecessora ser comprometida (280).

15. Meio legível por computador **CARACTERIZADO** por ter instruções executáveis por computador para fazer com que um computador realize as etapas de acordo com a reivindicação 1 (200).

16. Método para prover gerenciamento de disputa com ordenamento **CARACTERIZADO** por compreender as etapas de:

prover um sistema de memória transacional de software que suporta uma ordem predeterminada de comprometimento para um ou mais grupos de transações (342);

prover um processo de gerenciamento de disputa que é invocado quando ocorrer um conflito entre uma primeira transação e uma segunda transação (344); e

usar a ordem predeterminada de comprometimento no processo de gerenciamento de disputa para auxiliar a determinar se a primeira transação ou se a segunda transação deve vencer o conflito e ter permissão para prosseguir (346).

17. Método, de acordo com a reivindicação 16, **CARACTERIZADO** pelo fato de que se o processo de gerenciamento de disputa determinar que a primeira transação e a segunda transação constituem parte de um mesmo grupo de transações (348), então um campo de primeiro número de ordem representando a primeira transação e um campo de segundo número de ordem representando a segunda transação são comparados (352), e uma transação específica da primeira transação e da segunda transação que tem um número de or-

dem respectivo mais baixo tem permissão para prosseguir (354).

18. Meio legível por computador **CARACTERIZADO** por ter instruções executáveis por computador para fazer com que um computador realize as etapas de acordo com a reivindicação 16 (200).

19. Meio legível por computador **CARACTERIZADO** por ter instruções executáveis por computador para fazer com que um computador realize as etapas compreendendo:

prover um sistema de memória transacional de software (206);

prover um arbitrador de comprometimento que permite que uma ordem predeterminada de comprometimento seja especificada para uma pluralidade de transações (208), o arbitrador de comprometimento sendo operável para usar a ordem predeterminada de comprometimento em tempo de execução para auxiliar na determinação de uma ordem na qual comprometer a pluralidade de transições no sistema de memória transacional de software (210);

prover um processo de gerenciamento de contenção que é invocado quando ocorrer um conflito entre uma primeira transação e uma segunda transação (212); e

usar a ordem predeterminada de comprometimento no processo de gerenciamento de disputa para auxiliar na determinação de se a primeira transação ou a segunda transação deve vencer o conflito e ter permissão para prosseguir (214).

20. Meio legível por computador, de acordo com a reivindicação 19, **CARACTERIZADO** pelo fato de que o arbitrador de comprometimento é operável para impor ordenamento de comprometimento dentro das transações embutidas (380).

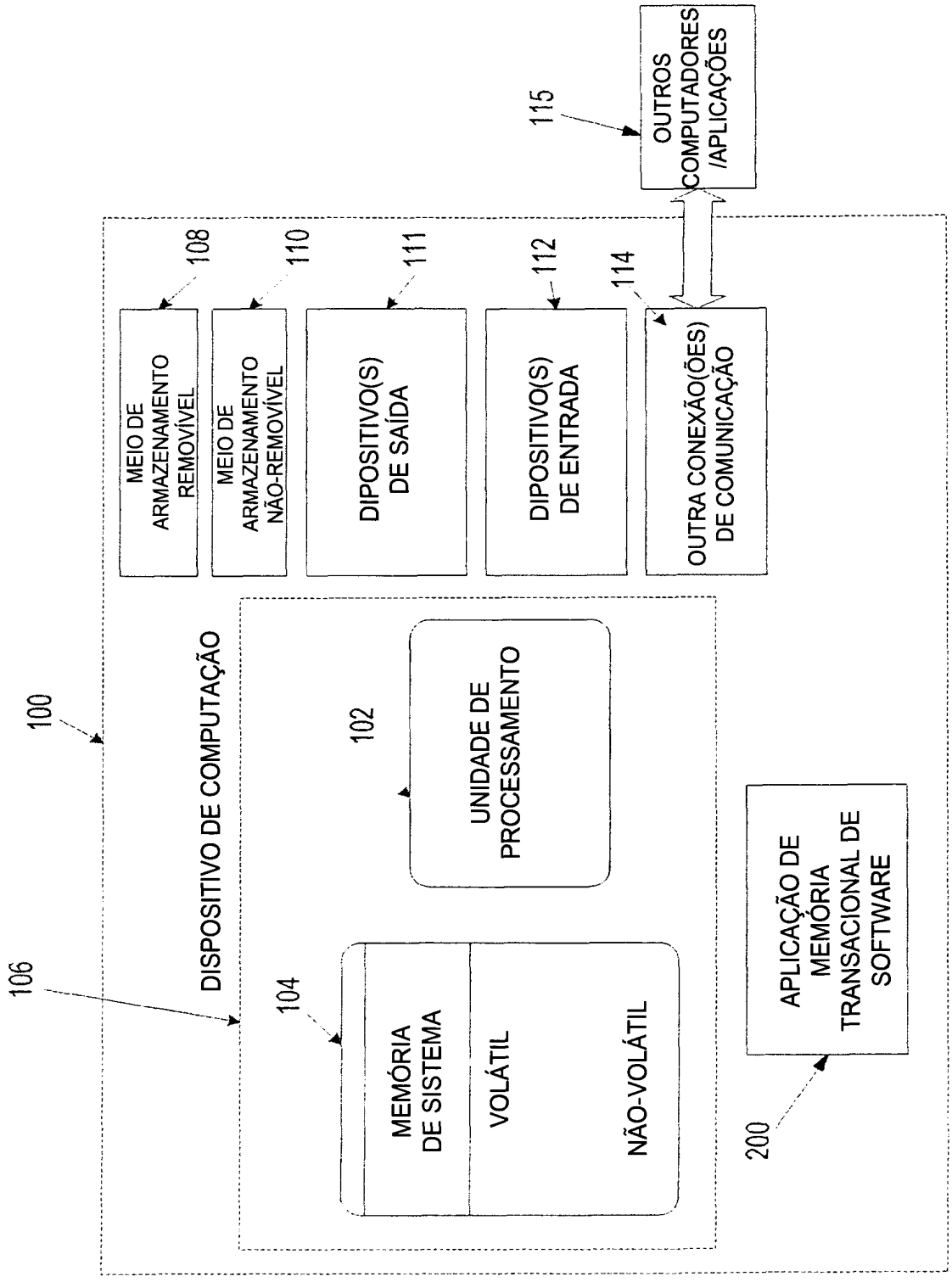
100
DISPOSITIVO DE COMPUTAÇÃO
106
102
UNIDADE DE PROCESSAMENTO
104
MEMÓRIA DE SISTEMA
VOLÁTIL
NÃO-VOLÁTIL
200
APLICAÇÃO DE MEMÓRIA TRANSACIONAL DE SOFTWARE
108
MEIO DE ARMAZENAMENTO REMOVÍVEL
110
MEIO DE ARMAZENAMENTO NÃO-REMOVÍVEL
111
DIPOSITIVO(S) DE SAÍDA
112
DIPOSITIVO(S) DE ENTRADA
114
OUTRA CONEXÃO(ÕES) DE COMUNICAÇÃO
115
OUTROS COMPUTADORES /APLICAÇÕES

FIG. 1

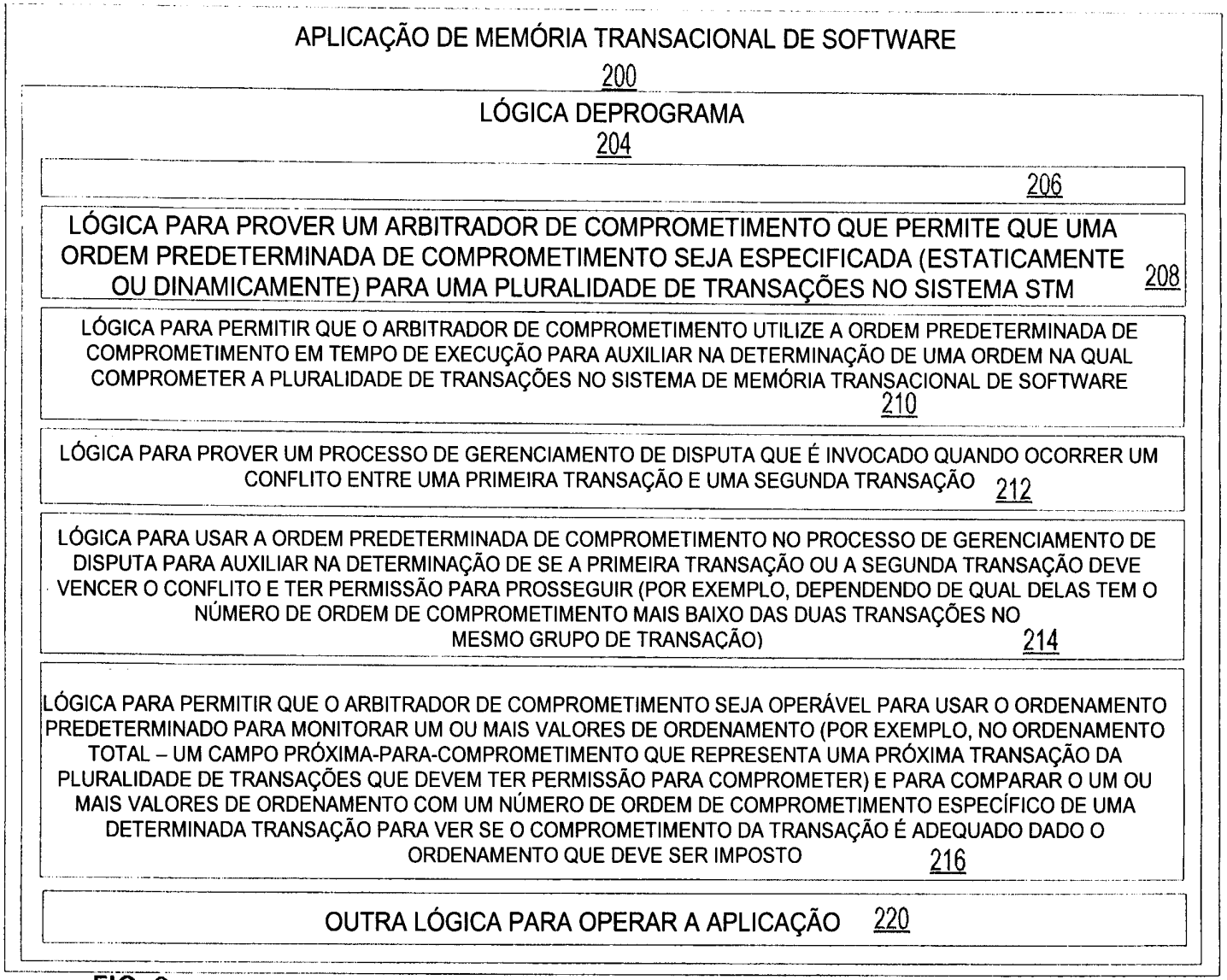
APLICAÇÃO DE MEMÓRIA TRANSACIONAL DE SOFTWARE
200
LÓGICA DEPROGRAMA
204
206
LÓGICA PARA PROVER UM ARBITRADOR DE COMPROMETIMENTO QUE PERMITE QUE UMA ORDEM PREDETERMINADA DE COMPROMETIMENTO SEJA ESPECIFICADA (ESTATICAMENTE OU DINAMICAMENTE) PARA UMA PLURALIDADE DE TRANSAÇÕES NO SISTEMA STM 208
LÓGICA PARA PERMITIR QUE O ARBITRADOR DE COMPROMETIMENTO UTILIZE A ORDEM PREDETERMINADA DE COMPROMETIMENTO EM TEMPO DE EXECUÇÃO PARA AUXILIAR NA DETERMINAÇÃO DE UMA ORDEM NA QUAL COMPROMETER A PLURALIDADE DE TRANSAÇÕES NO SISTEMA DE MEMÓRIA TRANSACIONAL DE SOFTWARE 210
LÓGICA PARA PROVER UM PROCESSO DE GERENCIAMENTO DE DISPUTA QUE É INVOCADO QUANDO OCORRER UM CONFLITO ENTRE UMA PRIMEIRA TRANSAÇÃO E UMA SEGUNDA TRANSAÇÃO 212
LÓGICA PARA USAR A ORDEM PREDETERMINADA DE COMPROMETIMENTO NO PROCESSO DE GERENCIAMENTO DE DISPUTA PARA AUXILIAR NA DETERMINAÇÃO DE SE A PRIMEIRA TRANSAÇÃO OU A SEGUNDA TRANSAÇÃO DEVE VENCER O CONFLITO E TER PERMISSÃO PARA PROSSEGUIR (POR EXEMPLO, DEPENDENDO DE QUAL DELAS TEM O NÚMERO DE ORDEM DE COMPROMETIMENTO MAIS BAIXO DAS DUAS TRANSAÇÕES NO MESMO GRUPO DE TRANSAÇÃO) 214
LÓGICA PARA PERMITIR QUE O ARBITRADOR DE COMPROMETIMENTO SEJA OPERÁVEL PARA USAR O ORDENAMENTO PREDETERMINADO PARA MONITORAR UM OU MAIS VALORES DE ORDENAMENTO (POR EXEMPLO, NO ORDENAMENTO TOTAL – UM CAMPO PRÓXIMA-PARA-COMPROMETIMENTO QUE REPRESENTA UMA PRÓXIMA TRANSAÇÃO DA PLURALIDADE DE TRANSAÇÕES QUE DEVEM TER PERMISSÃO PARA COMPROMETER) E PARA COMPARAR O UM OU MAIS VALORES DE ORDENAMENTO COM UM NÚMERO DE ORDEM DE COMPROMETIMENTO ESPECÍFICO DE UMA DETERMINADA TRANSAÇÃO PARA VER SE O COMPROMETIMENTO DA TRANSAÇÃO É ADEQUADO DADO O ORDENAMENTO QUE DEVE SER IMPOSTO 216
OUTRA LÓGICA PARA OPERAR A APLICAÇÃO 220

FIG. 2

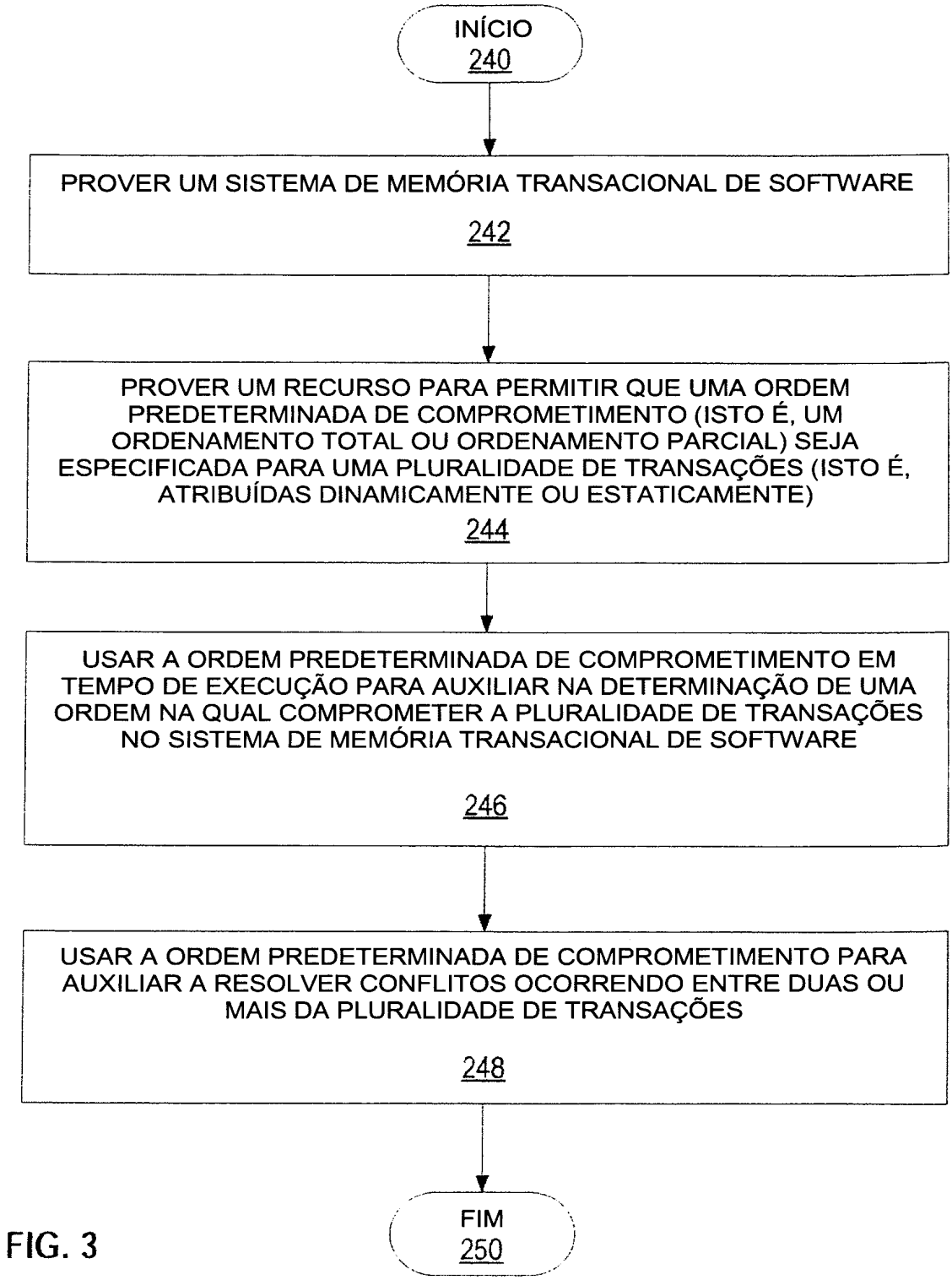
INÍCIO
240
PROVER UM SISTEMA DE MEMÓRIA TRANSACIONAL DE SOFTWARE
242
PROVER UM RECURSO PARA PERMITIR QUE UMA ORDEM PREDETERMINADA DE COMPROMETIMENTO (ISTO É, UM ORDENAMENTO TOTAL OU ORDENAMENTO PARCIAL) SEJA ESPECIFICADA PARA UMA PLURALIDADE DE TRANSAÇÕES (ISTO É, ATRIBUÍDAS DINAMICAMENTE OU ESTATICAMENTE)
244
USAR A ORDEM PREDETERMINADA DE COMPROMETIMENTO EM TEMPO DE EXECUÇÃO PARA AUXILIAR NA DETERMINAÇÃO DE UMA ORDEM NA QUAL COMPROMETER A PLURALIDADE DE TRANSAÇÕES NO SISTEMA DE MEMÓRIA TRANSACIONAL DE SOFTWARE
246
USAR A ORDEM PREDETERMINADA DE COMPROMETIMENTO PARA AUXILIAR A RESOLVER CONFLITOS OCORRENDO ENTRE DUAS OU MAIS DA PLURALIDADE DE TRANSAÇÕES
248
FIM
250

FIG. 3

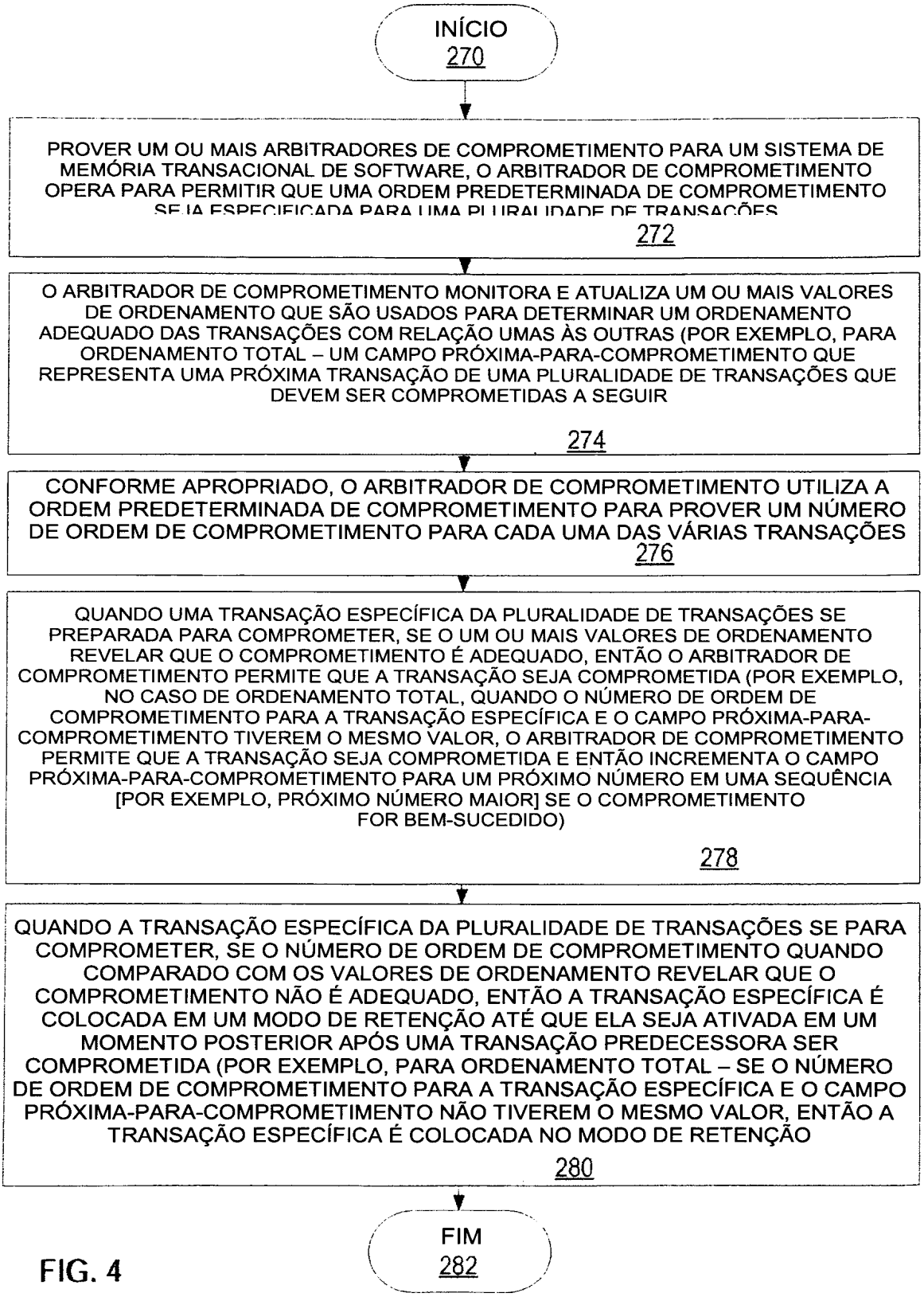
INÍCIO
270
PROVER UM OU MAIS ARBITRADORES DE COMPROMETIMENTO PARA UM SISTEMA DE MEMÓRIA TRANSACIONAL DE SOFTWARE, O ARBITRADOR DE COMPROMETIMENTO OPERA PARA PERMITIR QUE UMA ORDEM PREDETERMINADA DE COMPROMETIMENTO SEJA ESPECIFICADA PARA UMA PLURALIDADE DE TRANSAÇÕES
272
O ARBITRADOR DE COMPROMETIMENTO MONITORA E ATUALIZA UM OU MAIS VALORES DE ORDENAMENTO QUE SÃO USADOS PARA DETERMINAR UM ORDENAMENTO ADEQUADO DAS TRANSAÇÕES COM RELAÇÃO UMAS ÀS OUTRAS (POR EXEMPLO, PARA ORDENAMENTO TOTAL – UM CAMPO PRÓXIMA-PARA-COMPROMETIMENTO QUE REPRESENTA UMA PRÓXIMA TRANSAÇÃO DE UMA PLURALIDADE DE TRANSAÇÕES QUE DEVEM SER COMPROMETIDAS A SEGUIR
274
CONFORME APROPRIADO, O ARBITRADOR DE COMPROMETIMENTO UTILIZA A ORDEM PREDETERMINADA DE COMPROMETIMENTO PARA PROVER UM NÚMERO DE ORDEM DE COMPROMETIMENTO PARA CADA UMA DAS VÁRIAS TRANSAÇÕES
276
QUANDO UMA TRANSAÇÃO ESPECÍFICA DA PLURALIDADE DE TRANSAÇÕES SE PREPARADA PARA COMPROMETER, SE O UM OU MAIS VALORES DE ORDENAMENTO REVELAR QUE O COMPROMETIMENTO É ADEQUADO, ENTÃO O ARBITRADOR DE COMPROMETIMENTO PERMITE QUE A TRANSAÇÃO SEJA COMPROMETIDA (POR EXEMPLO, NO CASO DE ORDENAMENTO TOTAL, QUANDO O NÚMERO DE ORDEM DE COMPROMETIMENTO PARA A TRANSAÇÃO ESPECÍFICA E O CAMPO PRÓXIMA-PARA-COMPROMETIMENTO TIVEREM O MESMO VALOR, O ARBITRADOR DE COMPROMETIMENTO PERMITE QUE A TRANSAÇÃO SEJA COMPROMETIDA E ENTÃO INCREMENTA O CAMPO PRÓXIMA-PARA-COMPROMETIMENTO PARA UM PRÓXIMO NÚMERO EM UMA SEQUÊNCIA [POR EXEMPLO, PRÓXIMO NÚMERO MAIOR] SE O COMPROMETIMENTO FOR BEM-SUCEDIDO)
278
QUANDO A TRANSAÇÃO ESPECÍFICA DA PLURALIDADE DE TRANSAÇÕES SE PARA COMPROMETER, SE O NÚMERO DE ORDEM DE COMPROMETIMENTO QUANDO COMPARADO COM OS VALORES DE ORDENAMENTO REVELAR QUE O COMPROMETIMENTO NÃO É ADEQUADO, ENTÃO A TRANSAÇÃO ESPECÍFICA É COLOCADA EM UM MODO DE RETENÇÃO ATÉ QUE ELA SEJA ATIVADA EM UM MOMENTO POSTERIOR APÓS UMA TRANSAÇÃO PREDECESSORA SER COMPROMETIDA (POR EXEMPLO, PARA ORDENAMENTO TOTAL – SE O NÚMERO DE ORDEM DE COMPROMETIMENTO PARA A TRANSAÇÃO ESPECÍFICA E O CAMPO PRÓXIMA-PARA-COMPROMETIMENTO NÃO TIVEREM O MESMO VALOR, ENTÃO A TRANSAÇÃO ESPECÍFICA É COLOCADA NO MODO DE RETENÇÃO
280
FIM
282

FIG. 4

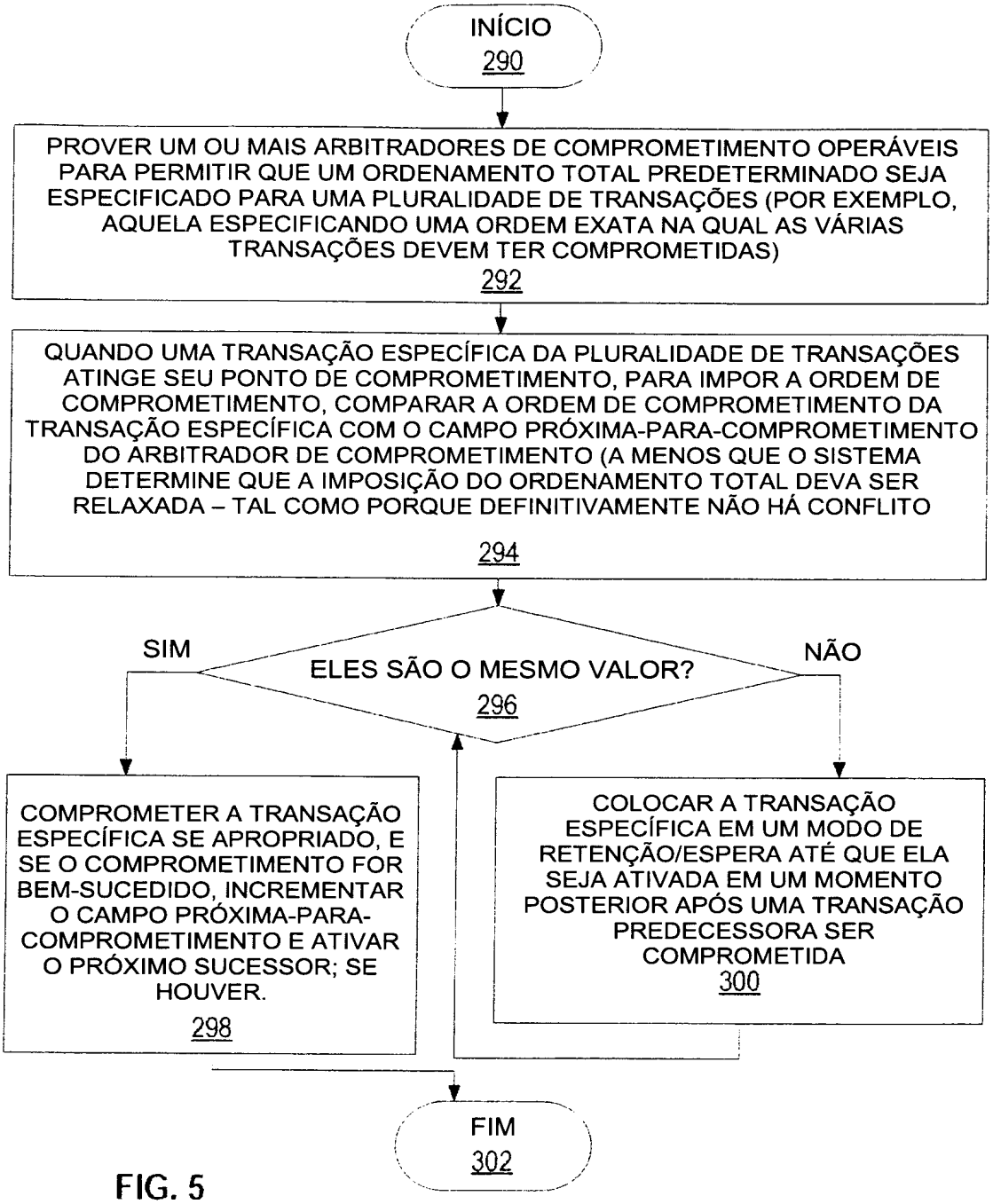
INÍCIO
290
PROVER UM OU MAIS ARBITRADORES DE COMPROMETIMENTO OPERÁVEIS PARA PERMITIR QUE UM ORDENAMENTO TOTAL PREDETERMINADO SEJA ESPECIFICADO PARA UMA PLURALIDADE DE TRANSAÇÕES (POR EXEMPLO, AQUELA ESPECIFICANDO UMA ORDEM EXATA NA QUAL AS VÁRIAS TRANSAÇÕES DEVEM TER COMPROMETIDAS)
292
QUANDO UMA TRANSAÇÃO ESPECÍFICA DA PLURALIDADE DE TRANSAÇÕES ATINGE SEU PONTO DE COMPROMETIMENTO, PARA IMPOR A ORDEM DE COMPROMETIMENTO, COMPARAR A ORDEM DE COMPROMETIMENTO DA TRANSAÇÃO ESPECÍFICA COM O CAMPO PRÓXIMA-PARA-COMPROMETIMENTO DO ARBITRADOR DE COMPROMETIMENTO (A MENOS QUE O SISTEMA DETERMINE QUE A IMPOSIÇÃO DO ORDENAMENTO TOTAL DEVA SER RELAXADA – TAL COMO PORQUE DEFINITIVAMENTE NÃO HÁ CONFLITO
294
ELES SÃO O MESMO VALOR?
296
SIM
NÃO
COMPROMETER A TRANSAÇÃO ESPECÍFICA SE APROPRIADO, E SE O COMPROMETIMENTO FOR BEM-SUCEDIDO, INCREMENTAR O CAMPO PRÓXIMA-PARA-COMPROMETIMENTO E ATIVAR O PRÓXIMO SUCESSOR; SE HOUVER.
298
COLOCAR A TRANSAÇÃO ESPECÍFICA EM UM MODO DE RETENÇÃO/ESPERA ATÉ QUE ELA SEJA ATIVADA EM UM MOMENTO POSTERIOR APÓS UMA TRANSAÇÃO PREDECESSORA SER COMPROMETIDA
300
FIM
302

FIG. 5

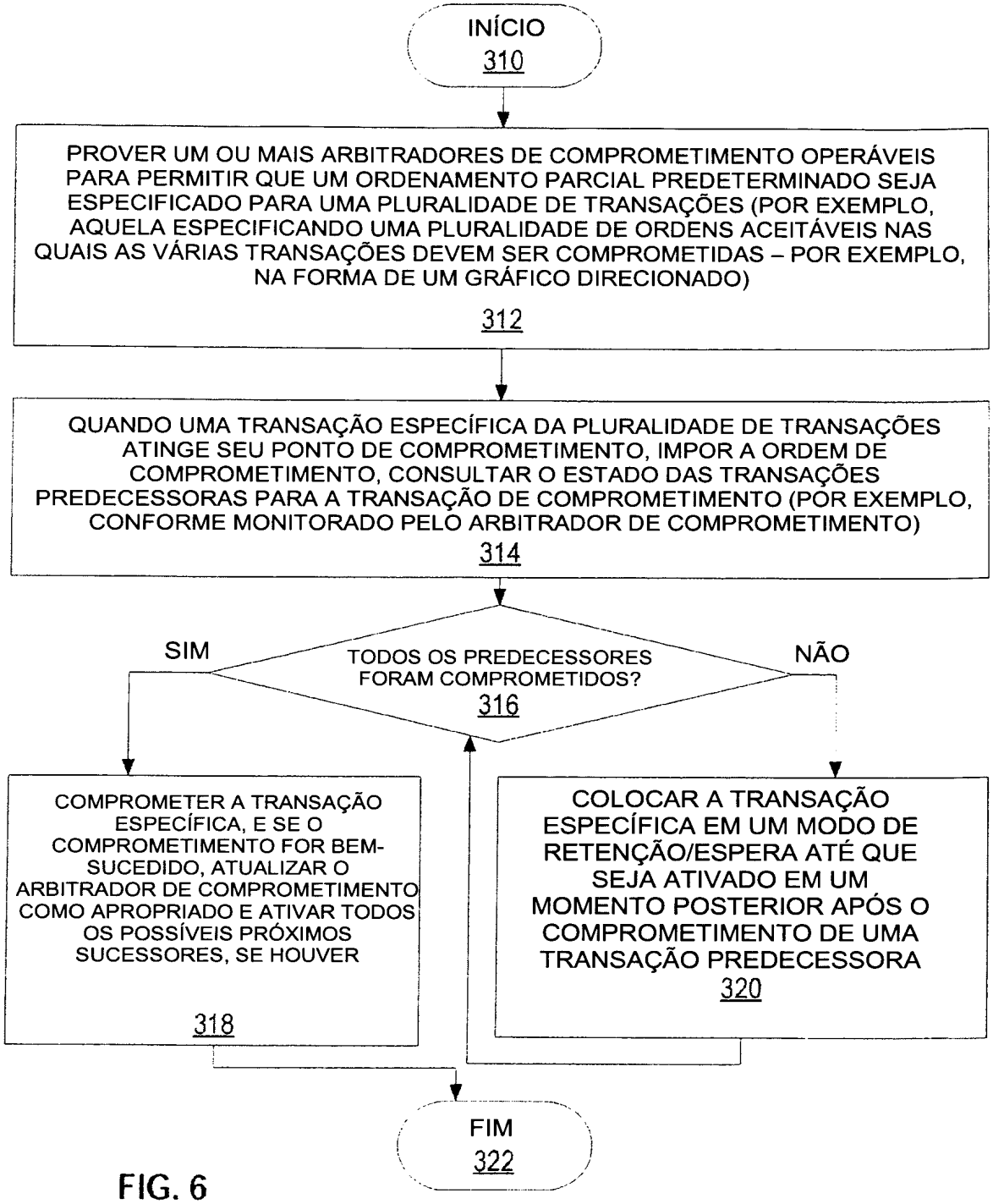
INÍCIO
310
PROVER UM OU MAIS ARBITRADORES DE COMPROMETIMENTO OPERÁVEIS PARA PERMITIR QUE UM ORDENAMENTO PARCIAL PREDETERMINADO SEJA ESPECIFICADO PARA UMA PLURALIDADE DE TRANSAÇÕES (POR EXEMPLO, AQUELA ESPECIFICANDO UMA PLURALIDADE DE ORDENS ACEITÁVEIS NAS QUAIS AS VÁRIAS TRANSAÇÕES DEVEM SER COMPROMETIDAS – POR EXEMPLO, NA FORMA DE UM GRÁFICO DIRECIONADO)
312
QUANDO UMA TRANSAÇÃO ESPECÍFICA DA PLURALIDADE DE TRANSAÇÕES ATINGE SEU PONTO DE COMPROMETIMENTO, IMPOR A ORDEM DE COMPROMETIMENTO, CONSULTAR O ESTADO DAS TRANSAÇÕES PREDECESSORAS PARA A TRANSAÇÃO DE COMPROMETIMENTO (POR EXEMPLO, CONFORME MONITORADO PELO ARBITRADOR DE COMPROMETIMENTO)
314
TODOS OS PREDECESSORES FORAM COMPROMETIDOS?
316
SIM
NÃO
COMPROMETER A TRANSAÇÃO ESPECÍFICA, E SE O COMPROMETIMENTO FOR BEM-SUCEDIDO, ATUALIZAR O ARBITRADOR DE COMPROMETIMENTO COMO APROPRIADO E ATIVAR TODOS OS POSSÍVEIS PRÓXIMOS SUCESSORES, SE HOUVER
318
COLOCAR A TRANSAÇÃO ESPECÍFICA EM UM MODO DE RETENÇÃO/ESPERA ATÉ QUE SEJA ATIVADO EM UM MOMENTO POSTERIOR APÓS O COMPROMETIMENTO DE UMA TRANSAÇÃO PREDECESSORA
320
FIM
322

FIG. 6

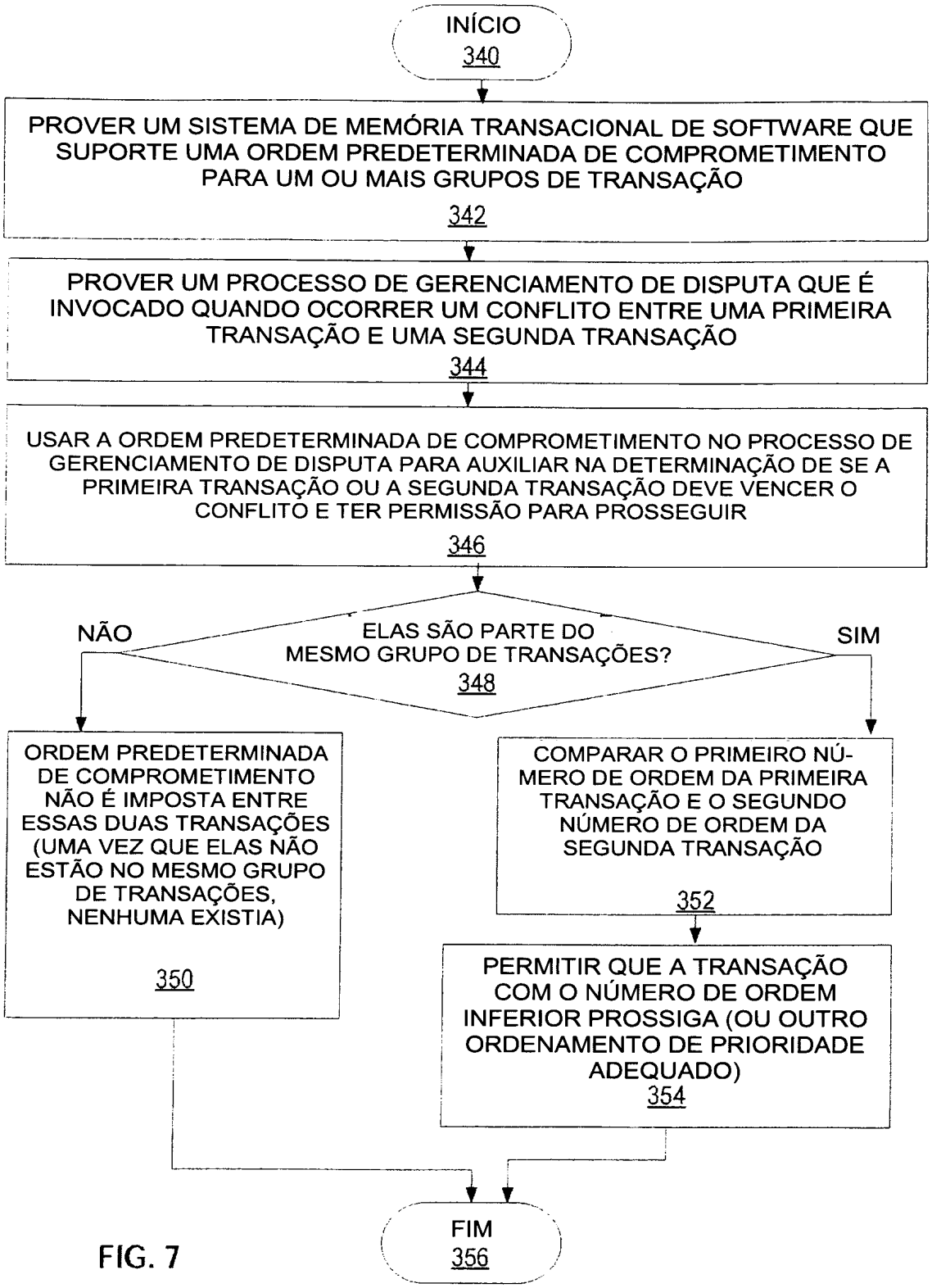
INÍCIO
340
PROVER UM SISTEMA DE MEMÓRIA TRANSACIONAL DE SOFTWARE QUE SUPORTE UMA ORDEM PREDETERMINADA DE COMPROMETIMENTO PARA UM OU MAIS GRUPOS DE TRANSAÇÃO
342
PROVER UM PROCESSO DE GERENCIAMENTO DE DISPUTA QUE É INVOCADO QUANDO OCORRER UM CONFLITO ENTRE UMA PRIMEIRA TRANSAÇÃO E UMA SEGUNDA TRANSAÇÃO
344
USAR A ORDEM PREDETERMINADA DE COMPROMETIMENTO NO PROCESSO DE GERENCIAMENTO DE DISPUTA PARA AUXILIAR NA DETERMINAÇÃO DE SE A PRIMEIRA TRANSAÇÃO OU A SEGUNDA TRANSAÇÃO DEVE VENCER O CONFLITO E TER PERMISSÃO PARA PROSSEGUIR
346
NÃO
ELAS SÃO PARTE DO MESMO GRUPO DE TRANSAÇÕES?
348
SIM
ORDEM PREDETERMINADA DE COMPROMETIMENTO NÃO É IMPOSTA ENTRE ESSAS DUAS TRANSAÇÕES (UMA VEZ QUE ELAS NÃO ESTÃO NO MESMO GRUPO DE TRANSAÇÕES, NENHUMA EXISTIA)
350
COMPARAR O PRIMEIRO NÚMERO DE ORDEM DA PRIMEIRA TRANSAÇÃO E O SEGUNDO NÚMERO DE ORDEM DA SEGUNDA TRANSAÇÃO
352
PERMITIR QUE A TRANSAÇÃO COM O NÚMERO DE ORDEM INFERIOR PROSSIGA (OU OUTRO ORDENAMENTO DE PRIORIDADE ADEQUADO)
354
FIM
356

FIG. 7

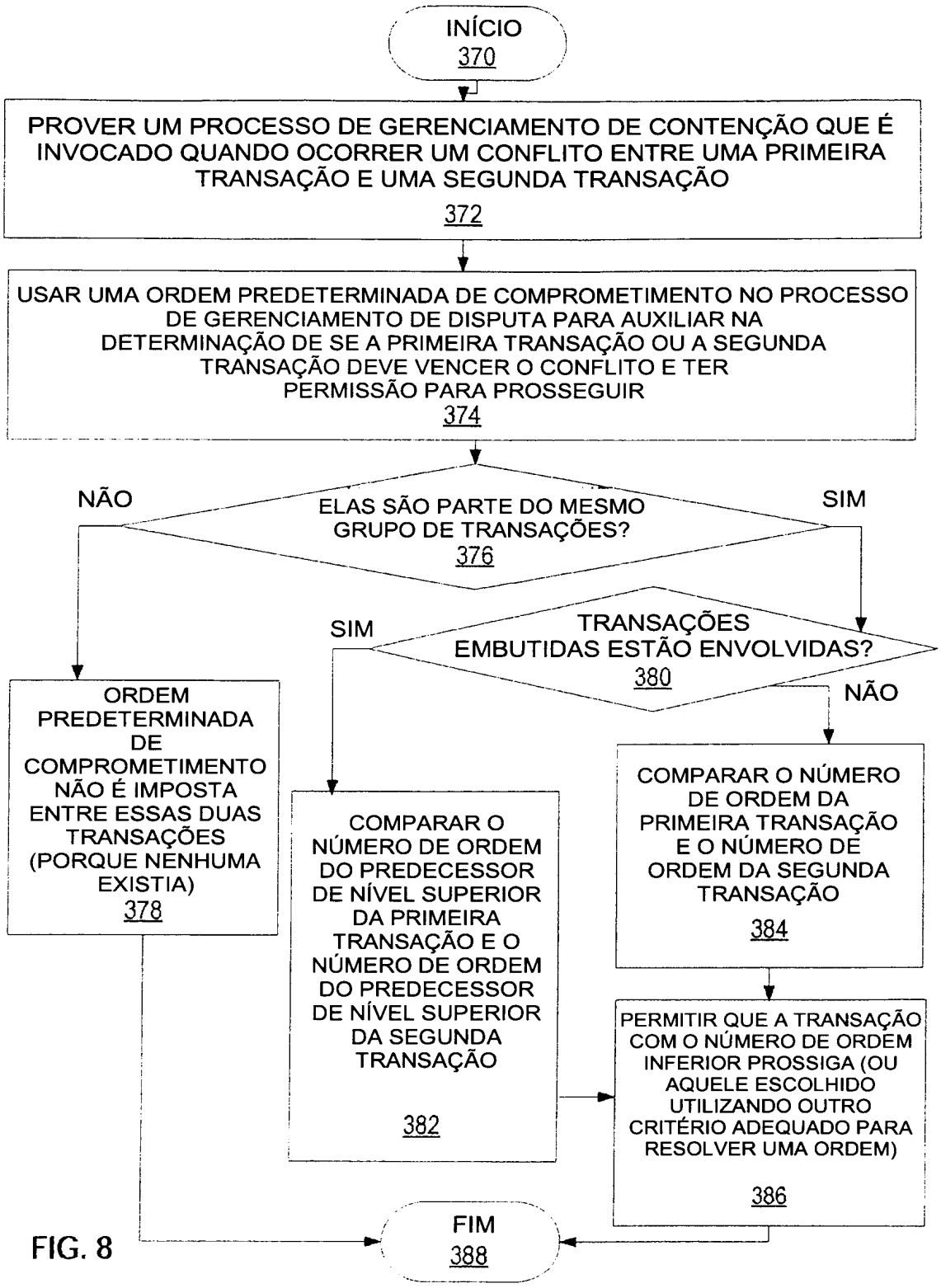
INÍCIO
370
PROVER UM PROCESSO DE GERENCIAMENTO DE CONTENÇÃO QUE É INVOCADO QUANDO OCORRER UM CONFLITO ENTRE UMA PRIMEIRA TRANSAÇÃO E UMA SEGUNDA TRANSAÇÃO
372
USAR UMA ORDEM PREDETERMINADA DE COMPROMETIMENTO NO PROCESSO DE GERENCIAMENTO DE DISPUTA PARA AUXILIAR NA DETERMINAÇÃO DE SE A PRIMEIRA TRANSAÇÃO OU A SEGUNDA TRANSAÇÃO DEVE VENCER O CONFLITO E TER PERMISSÃO PARA PROSSEGUIR
374
ELAS SÃO PARTE DO MESMO GRUPO DE TRANSAÇÕES?
376
NÃO
SIM
TRANSAÇÕES EMBUTIDAS ESTÃO ENVOLVIDAS?
380
SIM
NÃO
ORDEM PREDETERMINADA DE COMPROMETIMENTO NÃO É IMPOSTA ENTRE ESSAS DUAS TRANSAÇÕES (PORQUE NENHUMA EXISTIA)
378
COMPARAR O NÚMERO DE ORDEM DO PREDECESSOR DE NÍVEL SUPERIOR DA PRIMEIRA TRANSAÇÃO E O NÚMERO DE ORDEM DO PREDECESSOR DE NÍVEL SUPERIOR DA SEGUNDA TRANSAÇÃO
382
COMPARAR O NÚMERO DE ORDEM DA PRIMEIRA TRANSAÇÃO E O NÚMERO DE ORDEM DA SEGUNDA TRANSAÇÃO
384
PERMITIR QUE A TRANSAÇÃO COM O NÚMERO DE ORDEM INFERIOR PROSSIGA (OU AQUELE ESCOLHIDO UTILIZANDO OUTRO CRITÉRIO ADEQUADO PARA RESOLVER UMA ORDEM)
386
FIM
388

FIG. 8

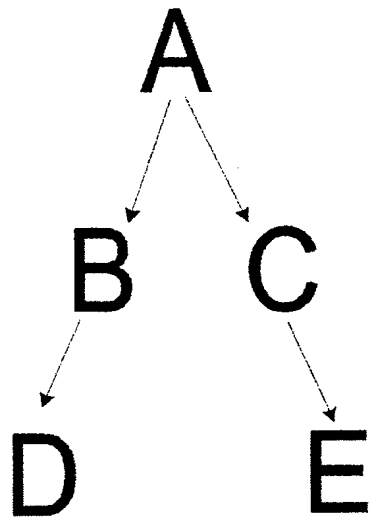

FIG. 9

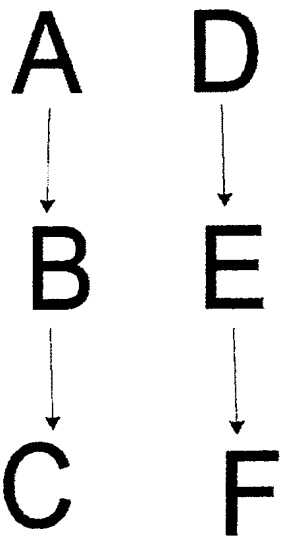

FIG. 10

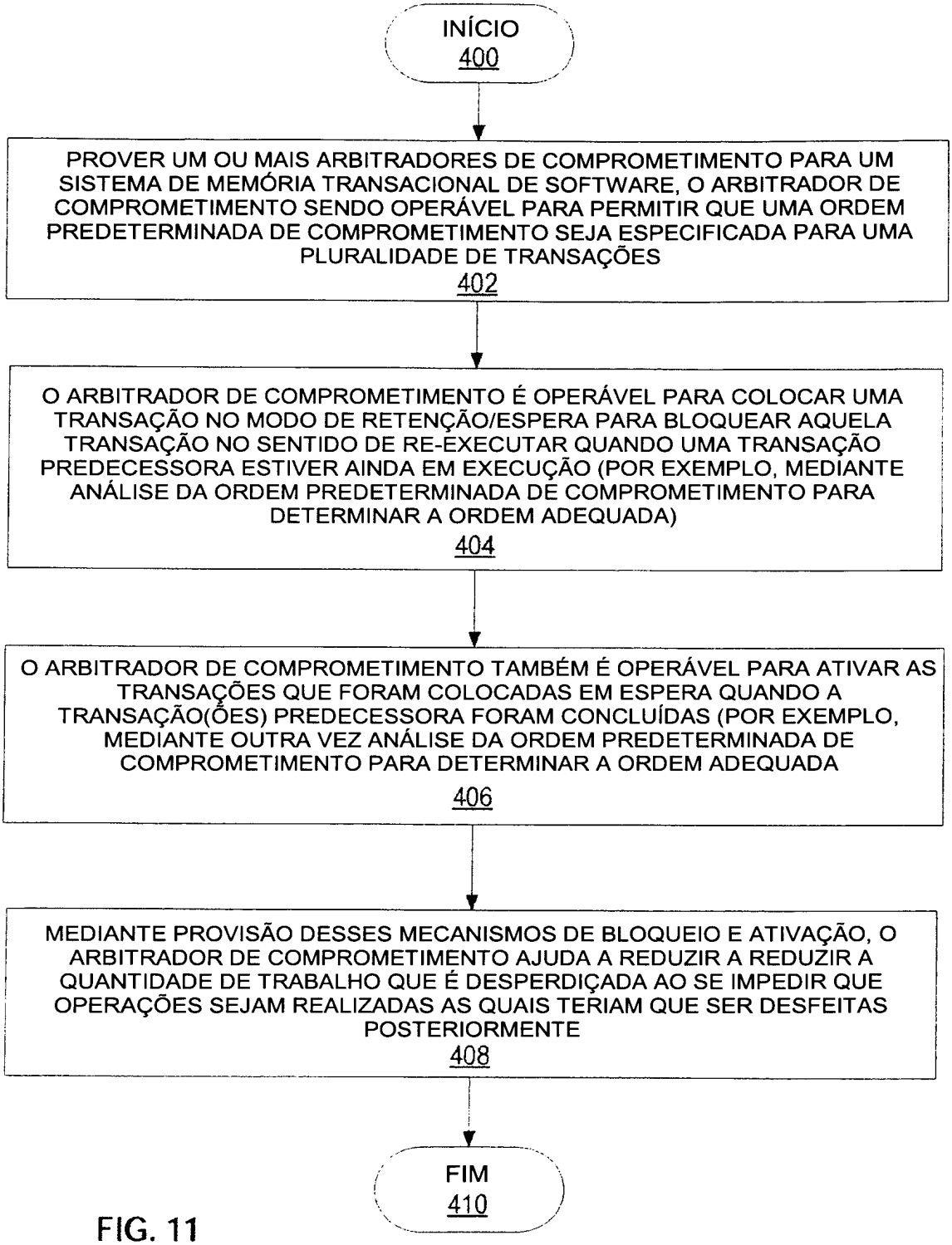
INÍCIO
400
PROVER UM OU MAIS ARBITRADORES DE COMPROMETIMENTO PARA UM SISTEMA DE MEMÓRIA TRANSACIONAL DE SOFTWARE, O ARBITRADOR DE COMPROMETIMENTO SENDO OPERÁVEL PARA PERMITIR QUE UMA ORDEM PREDETERMINADA DE COMPROMETIMENTO SEJA ESPECIFICADA PARA UMA PLURALIDADE DE TRANSAÇÕES
402
O ARBITRADOR DE COMPROMETIMENTO É OPERÁVEL PARA COLOCAR UMA TRANSAÇÃO NO MODO DE RETENÇÃO/ESPERA PARA BLOQUEAR AQUELA TRANSAÇÃO NO SENTIDO DE RE-EXECUTAR QUANDO UMA TRANSAÇÃO PREDECESSORA ESTIVER AINDA EM EXECUÇÃO (POR EXEMPLO, MEDIANTE ANÁLISE DA ORDEM PREDETERMINADA DE COMPROMETIMENTO PARA DETERMINAR A ORDEM ADEQUADA)
404
O ARBITRADOR DE COMPROMETIMENTO TAMBÉM É OPERÁVEL PARA ATIVAR AS TRANSAÇÕES QUE FORAM COLOCADAS EM ESPERA QUANDO A TRANSAÇÃO(ÕES) PREDECESSORA FORAM CONCLUÍDAS (POR EXEMPLO, MEDIANTE OUTRA VEZ ANÁLISE DA ORDEM PREDETERMINADA DE COMPROMETIMENTO PARA DETERMINAR A ORDEM ADEQUADA
406
MEDIANTE PROVISÃO DESSES MECANISMOS DE BLOQUEIO E ATIVAÇÃO, O ARBITRADOR DE COMPROMETIMENTO AJUDA A REDUZIR A REDUZIR A QUANTIDADE DE TRABALHO QUE É DESPERDIÇADA AO SE IMPEDIR QUE OPERAÇÕES SEJAM REALIZADAS AS QUAIS TERIAM QUE SER DESFEITAS POSTERIORMENTE
408
FIM
410

FIG. 11

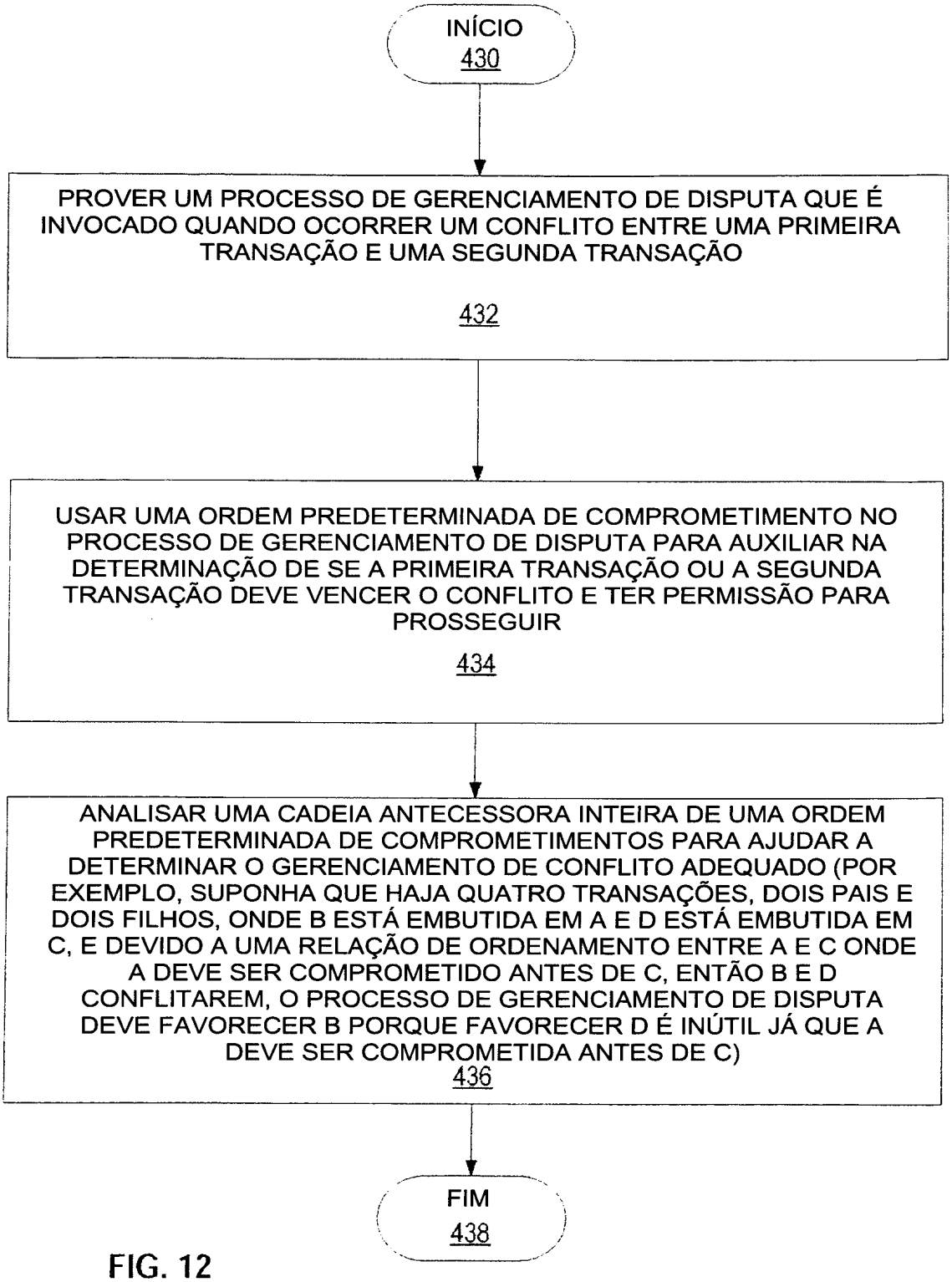
INÍCIO
430
PROVER UM PROCESSO DE GERENCIAMENTO DE DISPUTA QUE É INVOCADO QUANDO OCORRER UM CONFLITO ENTRE UMA PRIMEIRA TRANSAÇÃO E UMA SEGUNDA TRANSAÇÃO
432
USAR UMA ORDEM PREDETERMINADA DE COMPROMETIMENTO NO PROCESSO DE GERENCIAMENTO DE DISPUTA PARA AUXILIAR NA DETERMINAÇÃO DE SE A PRIMEIRA TRANSAÇÃO OU A SEGUNDA TRANSAÇÃO DEVE VENCER O CONFLITO E TER PERMISSÃO PARA PROSSEGUIR
434
ANALISAR UMA CADEIA ANTECESSORA INTEIRA DE UMA ORDEM PREDETERMINADA DE COMPROMETIMENTOS PARA AJUDAR A DETERMINAR O GERENCIAMENTO DE CONFLITO ADEQUADO (POR EXEMPLO, SUPONHA QUE HAJA QUATRO TRANSAÇÕES, DOIS PAIS E DOIS FILHOS, ONDE B ESTÁ EMBUTIDA EM A E D ESTÁ EMBUTIDA EM C, E DEVIDO A UMA RELAÇÃO DE ORDENAMENTO ENTRE A E C ONDE A DEVE SER COMPROMETIDO ANTES DE C, ENTÃO B E D CONFLITAREM, O PROCESSO DE GERENCIAMENTO DE DISPUTA DEVE FAVORECER B PORQUE FAVORECER D É INÚTIL JÁ QUE A DEVE SER COMPROMETIDA ANTES DE C)
436
FIM
438

FIG. 12

RESUMO

"ORDEM DE COMPROMETIMENTO DE TRANSAÇÃO DE SOFTWARE E GERENCIAMENTO DE CONFLITO"

São reveladas várias tecnologias e técnicas para aplicar ordenamento às transações em um sistema de memória transacional de software. É provido um sistema de memória transacional de software com um recurso para permitir que uma ordem predeterminada de comprometimento seja especificada para uma pluralidade de transações. A ordem predeterminada de comprometimento é usada em tempo de execução para auxiliar na determinação de uma ordem de comprometimento das transações no sistema de memória transacional de software. Um processo de gerenciamento de disputa é invocado quando ocorrer um conflito entre uma primeira transação e uma segunda transação. A ordem predeterminada de comprometimento é usada no processo de gerenciamento de disputa para auxiliar na determinação de se a primeira transação ou a segunda transação deve vencer o conflito e ter permissão para prosseguir.