



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

(21) **PI0707345-3 A2**

(22) Data de Depósito: 23/02/2007
(43) Data da Publicação: 03/05/2011
(RPI 2104)



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

(54) Título: **SERVIR RESULTADOS DE CONSULTA CACHED COM BASE EM UMA PORÇÃO DE CONSULTA**

(30) Prioridade Unionista: 28/02/2006 US 11/363.895

(73) Titular(es): Microsoft Corporation

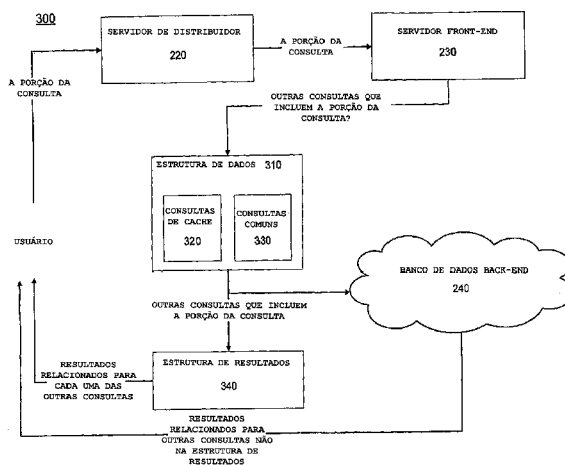
(72) Inventor(es): Gaura V Sareen, Girish Kumar, Michael Acheson Isard, Yinzhe Yu

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

(86) Pedido Internacional: PCT US2007004799 de 23/02/2007

(87) Publicação Internacional: WO 2007/100700 de 07/09/2007

(57) **Resumo:** SERVIR RESULTADOS DE CONSULTA CACHED COM BASE EM UMA PORÇÃO DE CONSULTA. As modalidades consideram um sistema e método para obter resultados relacionados para uma porção de uma consulta e para gerar um conjunto atualizado de consultas para um cache de um servidor. Outras consultas iniciando com a porção da consulta podem ser identificadas e obtidas a partir de uma estrutura de dados que inclui um cache de servidor e um conjunto de consultas comuns. Após as outras consultas serem identificadas, os resultados para as outras consultas são obtidos a partir do cache de servidor ou a partir de um banco de dados back-end. Um conjunto de consultas comuns, que pode incluir consultas deletadas e adicionais, pode ser utilizado para gerar o conjunto atualizado de consultas para o servidor. Tanto as consultas faltantes como as consultas deletadas que podem pertencer ao servidor com base em uma função de atribuição são inseridas em uma fila, que é posteriormente fornecida ao cache do servidor. A transferência pode ocorrer durante baixa atividade ou estado inativo.





PI0707345-3

"SERVIR RESULTADOS DE CONSULTA CACHED COM BASE EM UMA PORÇÃO DE CONSULTA"

ANTECEDENTES

Um motor de busca de rede orienta uma busca de usuário ou consulta, a um servidor de distribuidor, que, por sua vez, remete a consulta para um servidor front-end. O servidor front-end pode conter consultas e resultados correspondentes ou pode consultar com um banco de dados back-end para o resultado de consultas. O motor de busca de rede freqüentemente recebe consultas repetidas a partir de usuários que entram os mesmos termos de busca. Consultas recentes e seus resultados podem ser mantidos nos caches front-end do servidor front-end por um período de tempo predeterminado, uma vez que é mais eficiente e menos demorado consultar o cache front-end para os resultados desejados do que é consultar o banco de dados back-end, que contém um índice de rede completo e é tipicamente maior do que o cache front-end. Um índice de rede completo pode incluir todos os resultados disponíveis na rede, como páginas da rede; portanto, resultando em uma busca demorada uma vez que o índice é muito maior do que os resultados armazenados no cache front-end. Entretanto, se os resultados desejados não forem encontrados no cache front-end, o banco de dados back-end pode ser consultado.

Além de armazenar consultas e resultados recentes, os caches front-end podem servir também como um registro de buscas úteis, que podem incluir aqueles resultados que usuários consultaram mais freqüentemente ou aqueles resultados

que um administrador de sistema acredita serão consultados.

Um usuário deve entrar, tipicamente, um termo de busca completo ou consulta para ser apresentado com resultados relevantes. Se uma porção de uma consulta, como os primeiros poucos caracteres, for entrada, o usuário é frequentemente apresentado com mensagens de erro ou resultados incorretos. Adicionalmente, se um usuário não tem certeza da consulta a entrar para receber resultados, o usuário deve entrar individualmente várias consultas e separar através dos resultados para encontrar páginas de rede desejáveis, por exemplo. Seria desejável que um usuário entrasse uma porção de uma consulta e recebesse resultados relevantes para um grupo de consultas iniciando com a porção entrada. Adicionalmente, à medida que o usuário continua a entrar a consulta, desse modo estendendo a porção, seria desejável que os resultados ajustassem automaticamente à medida que a consulta muda.

Adicionalmente, um tal mecanismo de consultar porção seria adicionalmente desejável se recursos fossem preservados e as operações concluídas mais rápido por utilização adicional do cache front-end.

SUMÁRIO

Resultados relacionados para uma porção de uma consulta são obtidos identificando outras consultas que iniciam com a porção da consulta e obtendo os resultados das outras consultas. A porção da consulta pode ser atribuída a um servidor front-end que recebe outras consultas iniciando com a mesma porção da consulta. As outras consultas podem

ser armazenadas em uma estrutura de dados que inclui um cache de servidor e um conjunto comum de consultas. Os resultados das outras consultas podem ser armazenados no cache de servidor ou em um banco de dados back-end.

5 Um conjunto de consultas comuns pode ser formado por compilar consultas deletadas, deletadas a partir do cache de servidor. O conjunto de consultas comuns também pode incluir consultas adicionais a partir de uma lista predeterminada.

10 Um conjunto atualizado de consultas para o cache de servidor pode ser computado utilizando o conjunto de consultas comuns. Uma consulta faltante pode ser uma consulta que pertence ao cache de servidor porém talvez não esteja armazenado no cache de servidor. A consulta faltante pode
15 fazer parte do conjunto de consultas comuns. A consulta faltante pode ser inserida em uma fila e posteriormente colocada no cache de servidor durante um período de inatividade ou baixa atividade. Similarmente, uma consulta expirada, deletada a partir do cache de servidor, também pode ser inserida
20 na fila e posteriormente colocada no cache de servidor.

Esse Sumário é fornecido para introduzir uma seleção de conceitos em uma forma simplificada que são adicionalmente descritos abaixo na Descrição Detalhada. Esse Sumário não pretende identificar aspectos chave ou aspectos essenciais da matéria reivindicada, nem pretende ser utilizado
25 para limitar o escopo da matéria reivindicada.

BREVE DESCRIÇÃO DOS DESENHOS

O sumário acima e a descrição detalhada a seguir

são entendidos de forma melhor quando lidos em combinação com os desenhos apensos. Modalidades exemplares são mostradas nos desenhos, entretanto, entende-se que as modalidades não são limitadas aos métodos específicos e meios representados nas mesmas. Nos desenhos:

A figura 1 é um diagrama de blocos representando um dispositivo de computação exemplar;

A figura 2 é um diagrama de blocos representando uma organização de servidor exemplar;

10 A figura 3 é um diagrama de blocos representando um sistema de consulta;

A figura 4 é um diagrama de blocos representando um sistema de construção de consultas;

15 As figuras 5a e 5b são um diagrama de fluxo ilustrando uma modalidade de um método de fornecer resultados de consulta;

A figura 6 é um diagrama de fluxo ilustrando uma modalidade de um método de construir um conjunto de consultas comuns; e

20 A figura 7 é um diagrama de fluxo ilustrando uma modalidade de um método de atualizar um conjunto de consultas cache.

DESCRIÇÃO DETALHADA

25 Com referência à figura 1, um sistema exemplar para implementar a invenção inclui um dispositivo de computação de propósito geral na forma de um computador 110. Componentes de computador 110 podem incluir, porém não são limitados a, uma unidade de processamento 120, uma memória de

sistema 130, e um barramento de sistema 121 que acopla vários componentes de sistema incluindo a memória de sistema à unidade de processamento 120. O barramento de sistema 121 pode ser qualquer um de vários tipos de estruturas de barramento incluindo um barramento de memória ou controlador de memória, um barramento periférico, e um barramento local utilizando qualquer de uma variedade de arquiteturas de barramento. Como exemplo, e não limitação, tais arquiteturas incluem barramento de Arquitetura Padrão na indústria (ISA), barramento de Arquitetura de Micro canal (MCA), barramento de ISA aperfeiçoada (EISA), barramento local da Video Electronics Standards Association (VESA), e barramento de Interconectar Componente Periférico (PCI) (também conhecido como barramento Mezzanine).

O computador 110 inclui, tipicamente, uma variedade de meios legíveis por computador. Meios legíveis por computador podem ser quaisquer meios disponíveis que podem ser acessados pelo computador 110 e incluem meios tanto voláteis como não voláteis, meios removíveis e não removíveis. Como exemplo, e sem limitação, meios legíveis por computador podem compreender meios de armazenagem em computador e meios de comunicação. Meios de armazenagem em computador incluem meios tanto voláteis como não voláteis, removíveis quanto não removíveis implementados em qualquer método ou tecnologia para armazenagem de informações como instruções legíveis por computador, estruturas de dados, módulos de programa ou outros dados. Meios de armazenagem de computador incluem, porém não são limitados a, RAM, ROM, EEPROM, memória flash

ou outra tecnologia de memória, CD-ROM, digital versatile disks (DVD) ou outra armazenagem de disco óptico, cassetes magnéticos, fita magnética, armazenagem de disco magnético ou outros dispositivos de armazenagem magnética, ou qualquer
5 outro meio que possa ser utilizado para armazenar as informações desejadas e que possa ser acessado pelo computador 110. Meios de comunicação incorporam, tipicamente, instruções legíveis por computador, estruturas de dados, módulos de programa ou outros dados em um sinal de dados modulado
10 como uma onda portadora ou outro mecanismo de transporte e inclui quaisquer meios de fornecimento de informações. O termo "sinal de dados modulado" significa um sinal que tem uma ou mais de suas características definida ou alterada de tal modo a codificar informações no sinal. Como exemplo, e
15 não limitação, meios de comunicação incluem meios com fio como uma rede com fio ou conexão de fio direta, e meios sem fio como meios sem fio acústicos, RF, infravermelho e outros. Combinações de quaisquer dos acima devem ser incluídas também no escopo de meios legíveis por computador.

20 A memória do sistema 130 inclui meios de armazenagem em computador na forma de memória volátil e/ou não volátil como ROM 131 e RAM 132. Um sistema de entrada/saída básico 133 (BIOS), contendo as rotinas básicas que ajudam a transferir informações entre elementos no computador 110,
25 como durante partida, é tipicamente armazenado na ROM 131. A RAM 132 contém, tipicamente, módulos de dados e/ou programas que são imediatamente acessíveis a e/ou atualmente sendo operados pela unidade de processamento 120. Como exemplo, e

não limitação, a figura 1 ilustra o sistema operacional 134, programas de aplicação 135, outros módulos de programa 136 e dados de programa 137.

O computador 110 também pode incluir outros meios
5 de armazenagem em computador removíveis/não removíveis, voláteis/não voláteis. Somente como exemplo, a figura 1 ilustra uma unidade de disco rígido 141 que lê a partir de ou grava para meios magnéticos não voláteis, não removíveis, uma unidade de disco magnético 151 que lê a partir de ou
10 grava para um disco magnético não volátil, removível 152, e uma unidade de disco óptico 155 que lê a partir de ou grava para um disco óptico removível, não volátil 156, como um CD-ROM ou outros meios ópticos. Outros meios de armazenagem em computador removíveis/não removíveis, voláteis/não voláteis
15 que podem ser utilizados no ambiente operacional exemplar incluem, porém não são limitados a, cassetes de fita magnética, cartões de memória flash, digital versatile disks, 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
20 interface de memória não removível como uma interface 140, e a unidade de disco magnético 151 e a unidade de disco óptico 155 são tipicamente conectadas ao barramento de sistema 121 por uma interface de memória removível, como uma interface
25 150.

As unidades e seus meios de armazenagem em computador associados, discutidos acima e ilustrados na figura 1, fornecem armazenagem de instruções legíveis por computador,

estruturas de dados, componentes, módulos de programa e outros dados para o computador 110. Na figura 1, por exemplo, a unidade de disco rígido 141 é ilustrada como armazenando sistema operacional 144, programas de aplicação 145, outros
5 módulos de programa 146, e dados de programa 147. Observe que esses componentes podem ser iguais ou diferentes do sistema operacional 134, programas de aplicação 135, outros módulos de programa 136 e dados de programa 137. O sistema operacional 144, programas de aplicação 145, outros módulos
10 de programa 146, e dados de programa 147 recebem números diferentes aqui para ilustrar que, no mínimo, há cópias diferentes. Um usuário pode entrar comandos e informações no computador 110 através de dispositivos de entrada como um teclado 162 e um dispositivo de indicação 161, comumente
15 mencionado como um mouse, trackball ou mesa sensível ao toque. Outros dispositivos de entrada (não mostrados) podem incluir um microfone, manche, bloco de jogo, prato de satélite, scanner ou similar. Esses e outros dispositivos de entrada são freqüentemente conectados à unidade de processamento 120 através de uma interface de entrada de usuário 160
20 que é acoplada ao barramento de sistema, porém pode ser conectada por outra interface e estruturas de barramento, como uma porta paralela, porta de jogo ou um barramento serial universal (USB). Um monitor 191 ou outro tipo de dispositivo
25 de exibição também é conectado ao barramento de sistema 121 através de uma interface, como uma interface de vídeo 190. Além do monitor, computadores também podem incluir outros dispositivos de saída periféricos como alto-falantes 197 e

uma impressora 196, que podem ser conectados através de uma interface periférica de saída 195.

O computador 110 pode operar em um ambiente ligado em rede utilizando conexões lógicas a um ou mais computadores remotos, como computador remoto 180. O computador remoto 180 pode ser um computador pessoal, um servidor, um roteador, um PC de rede, um dispositivo par ou outro nó de rede comum, e tipicamente inclui muitos ou todos os elementos descritos acima em relação ao computador 110, embora somente um dispositivo de armazenagem de memória 181 tenha sido ilustrado na figura 1. As conexões lógicas representadas incluem uma rede de área local (LAN) 171 e uma rede remota (WAN) 173, porém podem incluir também outras redes. Tais ambientes de ligação em rede são comuns em escritórios, redes de computador de empresa, intranets e Internet.

Quando utilizado em um ambiente de ligação em rede LAN, o computador 110 é conectado a LAN 171 através de um adaptador ou interface de rede 170. Quando utilizado em um ambiente de ligação em rede WAN, o computador 110 inclui, tipicamente, um modem 172 ou outro meio para estabelecer comunicação através da WAN 173, como Internet. O modem 172, que pode ser interno ou externo, pode ser conectado ao barramento de sistema 121 via interface de entrada de usuário 160, ou outro mecanismo apropriado. Em um ambiente ligado em rede, módulos de programa representados em relação ao computador 110, ou porções do mesmo, podem ser armazenadas no dispositivo de armazenagem de memória remoto. Como exemplo, e não limitação, a figura 1 ilustra programas de aplicação

remotos 185 como residindo no dispositivo de memória 181. Será reconhecido que as conexões de rede mostradas são exemplares e outros meios de estabelecer um link de comunicação entre os computadores pode ser utilizado.

5 Toda ou porções dos métodos da presente invenção descrita acima podem ser incorporadas em hardware, software ou uma combinação dos dois. Quando incorporado em software, os métodos da presente invenção, ou certos aspectos ou porções dos mesmos, podem ser incorporados na forma de código
10 de programa que quando executado por um sistema de computação fazem com que o sistema de computação execute os métodos da presente invenção. Esse código de programa pode ser armazenado em qualquer meio legível por computador, como aquele termo é definido acima.

15 A figura 2 representa uma organização de servidor 200 que pode ser utilizada para fornecer resultados para consultas, como consultas ou porções de consultas inseridas em um motor de busca de internet através, por exemplo, do computador 110. Uma consulta se origina na Internet 210, on-
20 de um usuário pode inserir uma consulta para a qual deseja obter resultados relevantes, como páginas de rede dirigidas à consulta. A consulta pode ser inserida em um motor de busca, que pode permitir que o usuário insira a consulta de interesse e que pode exibir os resultados relevantes para o
25 usuário acessar. A Internet 210 pode ser conectada a um servidor de distribuidor 220, que recebe a consulta a partir da Internet 210. Múltiplos servidores de distribuidor 220 podem ser conectados à Internet 210. A figura 2 ilustra três ser-

vidores de distribuidor 220 (220a, 220b, e 220c), embora o número dos servidores de distribuidor 220 não seja desse modo limitado. A consulta pode ser atribuída a um servidor específico dos servidores de distribuidor 220, por exemplo, por um endereço de protocolo de Internet (IP) de origem da consulta. Outros esquemas podem ser utilizados para atribuir a consulta a um dos servidores de distribuidor 220. Alternativamente, a consulta pode ser aleatoriamente enviada a qualquer um dos servidores de distribuidor 220.

O servidor de distribuidor 220 que recebe a consulta pode transmitir a consulta para um servidor front-end 230, como um dos servidores front-end 230a, 230b, 230c e 230n mostrados na organização de servidor 200 da figura 2. A organização de servidor 200 não é limitada a um certo número dos servidores front-end 230. O servidor front-end 230 que recebe a consulta a partir do servidor de distribuidor 220 pode fazer isso com base em um mapeamento que liga a consulta ao servidor front-end 230. Por exemplo, a consulta pode ser hashed para produzir um valor, que, por sua vez, é utilizado em uma função de mapeamento que coloca certas consultas em servidores front-end específicos 230 e pode assegurar que a mesma consulta seja sempre, ou quase isso, dirigida ao mesmo servidor front-end 230. A função de mapeamento pode determinar qual dos servidores front-end 230 receberá a consulta. O servidor front-end 230 não necessita ser atribuído com base em uma função de mapeamento e pode em vez disso, por exemplo e não por limitação, receber aleatoriamente a consulta.

Um cache, ou armazenagem local, pode conter consultas e seus resultados correspondentes. O cache pode fazer parte do servidor front-end 230, e as consultas e resultados do cache podem ser um conjunto de consultas e resultados recentes, como aqueles a partir de um período de tempo prede-

5 terminado, por exemplo três horas. Quando o servidor front-end 230 recebe a consulta a partir do servidor de distribuidor 220, o servidor front-end 230 pode buscar seu cache para determinar se a consulta, e seus resultados correspondentes,

10 já estão armazenados no cache. Se a consulta estiver no cache, então o servidor front-end 230 pode retornar os resultados a partir do cache. Os resultados podem ser fornecidos ao usuário que entrou a consulta.

Os servidores front-end 230 podem estar em comunicação com um banco de dados back-end 240, que pode conter

15 resultados existentes, por exemplo páginas de rede. Resultados de consulta podem ser armazenados no banco de dados back-end 240. Se a consulta entrada não estiver no cache do servidor front-end 230, então o banco de dados back-end 240

20 pode ser consultado para retornar resultados de consulta relevantes ao usuário. Como o banco de dados back-end 240 é tipicamente muito maior do que o cache dos servidores front-end 230, a consulta com o banco de dados back-end pode demorar um período de tempo significativamente mais longo do que

25 a consulta com o cache dos servidores front-end 230. Portanto, a consulta com o banco de dados back-end 240 pode ocorrer somente, por exemplo, quando o servidor front-end 230 não contiver os resultados solicitados ou quando o servidor

front-end 230 encontra um problema, como uma falha do servidor.

A figura 3 é um diagrama de blocos que representa um sistema de consulta 300, que pode ser utilizado para obter resultados para uma porção de uma consulta. O sistema de consulta 300 pode incluir o servidor de distribuidor 220 para receber uma porção de uma consulta. A porção da consulta pode ser, por exemplo, os três primeiros caracteres de uma consulta. Um usuário que deseja resultados para a consulta, pode digitar ou entrar a porção da consulta em um motor de busca na Internet. A porção da consulta pode ser dirigida ao servidor de distribuidor 220 com base em um endereço IP do computador 110 a partir de onde a porção da consulta se origina. A porção da consulta pode ser aleatoriamente dirigida ao servidor de distribuidor 220.

O servidor de distribuidor 220 pode ser responsável por atribuir a porção da consulta ao servidor front-end 230. Uma função hash pode ser utilizada pelo servidor de distribuidor 220 para essa tarefa de atribuição. O servidor de distribuidor 220 pode, por exemplo, atribuir a porção da consulta ao servidor front-end 230 de modo que outras consultas que têm a mesma porção serão atribuídas ao mesmo servidor front-end 230. Após a porção da consulta ser atribuída ao servidor front-end 230, o servidor de distribuidor 220 pode transmitir a porção da consulta ao servidor front-end atribuído, 230.

Após o servidor front-end 230 receber a porção da consulta, o servidor front-end 230 pode ser responsável por

executar uma busca para obter outras consultas que têm a mesma porção que a porção da consulta. Para obter essas consultas, o servidor front-end 230 pode consultar uma estrutura de dados 310. A estrutura de dados 310, embora mostrado
5 como um componente separado na figura 3, pode fazer parte do servidor front-end 230. A estrutura de dados 310 pode incluir um cache de servidor, como consultas cache 320. As consultas cache 320 podem incluir consultas anteriores que foram mantidas na estrutura de dados 310 por um período de tempo
10 predeterminado. Desse modo, as consultas cache 320 podem, por exemplo, incluir as consultas mais recentes. As consultas cache 320 podem ser mantidas na estrutura de dados 310 para acesso fácil e rápido a consultas recentes. A estrutura de dados 310 pode incluir também um conjunto de consultas
15 comuns 330. A formação do conjunto de consultas comuns é descrita em mais detalhe abaixo.

A estrutura de dados pode ser uma estrutura trie aumentada, que é uma estrutura que facilita recuperação rápida de informações relacionadas a uma consulta ou uma por-
20 ção de uma consulta. Por exemplo, possíveis porções de uma consulta juntamente com informações adicionais relacionadas a cada porção da consulta podem ser armazenadas na estrutura trie. As informações relacionadas a cada porção da consulta podem ser, por exemplo, uma lista de um número predetermina-
25 do de consultas mais freqüentemente solicitadas iniciando com a porção da consulta. A lista pode ser separada em uma ordem de freqüência descendente, de modo que a consulta solicitada mais comumente iniciando com a porção da consulta é

primeiramente listada.

Quando uma lista de outras consultas, que pode incluir a porção da consulta, é obtida a partir da estrutura de dados 310, uma estrutura de resultados 340, que pode ser
5 uma estrutura cache, é consultada. A estrutura de resultados pode ser uma parte do servidor front-end 230. A estrutura de resultados 340 contém resultados para consultas. Após consulta com a estrutura de resultados 340, os resultados para consultas que iniciam com a porção da consulta são obtidos
10 e fornecidos ao usuário. Esses resultados podem ser, por exemplo, uma seleção múltipla de páginas da rede.

O banco de dados back-end 240 também pode ser incluído como parte do sistema de consulta 300. O banco de dados back-end 240 pode ser consultado se resultados para uma
15 ou mais das consultas que incluem a porção da consulta não forem localizados na estrutura de resultados 340. Os resultados obtidos a partir do banco de dados back-end 240 são também fornecidos ao usuário, como mostrado na figura 3.

Uma representação de diagrama de blocos de um sistema de construção de consultas comuns 400 é ilustrada na
20 figura 4. O sistema 400 pode ser atribuído a tarefa de criar as consultas comuns 330, que podem ser utilizadas para obter consultas que têm a mesma porção que a porção da consulta. O sistema de construção de consultas comuns 400 pode incluir o
25 servidor front-end 230, que recebe a porção da consulta a partir do servidor de distribuidor 220. Um cache de servidor 410 pode fazer parte do servidor front-end 230. O cache de servidor 410 pode ser uma estrutura de cache do servidor

front-end 230 que armazena consultas anteriores, como consultas que foram mantidas por um período de tempo predeterminado. O cache de servidor 410 do servidor front-end 230 pode ser o mesmo componentes que as consultas de cache 320 da estrutura de dados 310, que podem fazer parte do servidor front-end 230.

As consultas podem ser freqüentemente deletadas a partir do cache de servidor 410 do servidor front-end 230. As deleções podem ocorrer devido ao tamanho limitado do cache de servidor 410; uma consulta pode necessitar ser deletada para criar espaço de armazenagem para uma consulta mais nova (ou porção da consulta). Ou, as deleções podem ocorrer periodicamente. Quando as consultas são deletadas a partir do cache de servidor 410, uma entrada log 420 pode receber e rastrear as consultas deletadas. Após decorrer um período de tempo predeterminado, a entrada log 420 pode fornecer as consultas deletadas para um compilador de entrada log 430, que compila as entradas deletadas. O compilador de entrada log 430 pode compilar as entradas deletadas com base em critérios anteriormente determinados, como, por exemplo, aquelas entradas que aparecem mais de uma vez durante um período de uma semana. O compilador de entrada log 430, após ter concluído a compilação de consultas deletadas, pode fornecer a compilação para as consultas comuns 330 da estrutura de dados 310. A estrutura de dados 310 armazena então as consultas comuns para uso em fornecer consultas que têm a mesma porção que a porção da consulta buscada para a estrutura de resultados 340.

As consultas comuns 330 podem servir como um suplemento para as consultas cache 320. As consultas cache 320 podem ser consultas mais recentes, enquanto as consultas comuns 330, que podem não necessariamente ser consultas recentes, podem ser consultas freqüentemente solicitadas a partir de um período de tempo relativamente longo, predeterminado, como uma semana.

As figuras 5a e 5b ilustram um método de fornecer resultados de consulta. Em 505, uma porção de uma consulta é obtida. A porção da consulta pode ser iniciada por um usuário, que, desejando resultados relacionados à consulta, entra a porção da consulta em um motor de busca na Internet, por exemplo. A porção da consulta pode ser obtida no servidor de distribuidor 220. Em 510, a porção da consulta é examinada para determinar se a porção da consulta casa com uma entrada predeterminada. Se a porção da consulta casar uma entrada predeterminada, então, em 515 caracteres seguintes são atribuídos como a porção da consulta. Como um número de entradas que ocorrem freqüentemente podem existir que o usuário não pretende que seja parte da consulta, essa comparação pode eliminar a provisão de resultados de consulta para consultas indesejáveis. Por exemplo, se um usuário entra tais consultas iniciando como "http" e "www", o usuário tipicamente não deseja buscar essas consultas porém pode, em vez disso, estar interessado nos caracteres que seguem essas entradas que ocorrem freqüentemente. Desse modo, por atribuir caracteres seguinte, como três caracteres após "www", como a porção da consulta, a porção adequada da consulta pode

ser buscada. A porção da consulta pode ser comparada com outras entradas predeterminadas.

Após caracteres seguintes serem atribuídos como a porção da consulta, e se a porção da consulta não casar com uma entrada predeterminada, então o método de fornecer resultados de consulta pode prosseguir para 520 a partir de 515 e 510, respectivamente. Em 520, uma representação da porção da consulta pode ser computada. Essa representação pode ser uma função hash da porção da consulta, por exemplo. Outras representações também são possíveis.

Em 525, após 520, a porção da consulta pode ser atribuída a um servidor front-end 230. Como uma representação da porção da consulta foi computada em 520, então a representação pode ser utilizada para atribuir a porção da consulta ao servidor front-end 230. Outros critérios de atribuição podem ser utilizados para atribuir a porção da consulta ao servidor front-end. OU a atribuição pode ser uma seleção aleatória. Em 530, a porção da consulta pode ser enviada para o servidor front-end atribuído 230.

535 determina se uma estrutura de dados, como a estrutura de dados 310 descrita acima com relação à figura 3 e figura 4, contém consultas que iniciam com a porção da consulta. Se a estrutura de dados 310, que pode incluir consultas cache 320 que podem incluir consultas anteriores que foram mantidas na estrutura de dados 310 por um período de tempo predeterminado e consultas comuns 330, não contiver consultas que iniciam com a porção da consulta, então o usuário, em 540, recebe uma lista de resultado vazia. Se, en-

tretanto, a estrutura de dados 310 contiver consultas que iniciam com a porção da consulta, então o método prossegue para 545 (figura 5b). Em 545, as consultas iniciando com a porção da consulta podem ser obtidas a partir da estrutura de dados 310.

Em 550, uma determinação é feita para determinar se, para cada consulta obtida iniciando com a porção da consulta, os resultados são armazenados em um cache de servidor, como a estrutura de resultados 340, que pode fazer parte do servidor front-end 230. Em 560, se os resultados para uma consulta obtida iniciando com a porção da consulta não estiverem na estrutura de resultados 340, então os resultados são obtidos a partir do banco de dados back-end 240. Em 555, se em vez disso os resultados para uma consulta obtida iniciando com a porção da consulta estiverem na estrutura de resultados 340, então os resultados são obtidos a partir da estrutura de resultados 340. A determinação em 550 é executada para cada consulta obtida que inclui a porção da consulta. Isso pode assegurar que os resultados são fornecidos para cada consulta que inclui a porção da consulta, quer os resultados sejam do banco de dados back-end 240 ou da estrutura de resultados 340. O método de fornecer consulta termina em 565 onde os resultados são fornecidos.

A figura 6 é um diagrama de fluxo ilustrando um método de construir um conjunto de consultas comuns. Em 610, consultas são geradas com base em uma lista predeterminada. A lista predeterminada pode incluir consultas que um administrador de sistema acredita devam fazer parte do conjunto

de consultas comuns 330. Por exemplo, uma consulta predeterminada pode ser um novo filme que não foi utilizado em buscas de consulta por usuários porém que se espera que se torne popular e buscado amplamente no futuro próximo. Pela adição de tais consultas às consultas comuns 330, que é parte da estrutura de dados 310, uma busca para a consulta, ou porção da consulta pode não exigir consulta com o banco de dados back-end 240.

Em 615, consultas deletadas a partir de um cache de um servidor, como o cache de servidor 410 do servidor front-end 230 ou as consultas cache 320, são compiladas. As consultas deletadas podem ser compiladas durante um período de tempo predeterminado. As consultas deletadas podem ser rastreadas pela entrada log 420 e compiladas pelo compilador de entrada log 430, como discutido acima com relação ao sistema de construção de consulta comum 400 da figura 4.

Em 620, o conjunto de consultas comuns 330 é formado. As consultas comuns podem incluir uma combinação ou compilação das consultas predeterminadas, geradas em 610, e as consultas deletadas, geradas em 615. Após formar as consultas comuns 330, em 625, as consultas comuns 330 são distribuídas para os servidores front-end 230. Se as consultas deletadas forem aquelas deletadas a partir de um servidor front-end específico, como o servidor front-end 230b da figura 2, então as consultas comuns podem ser distribuídas para os outros servidores front-end 230a, 230c, e 230n do sistema 200.

A figura 7 é um diagrama de fluxo ilustrando um

método de atualizar as consultas cache 320 de uma estrutura de dados 310 de um servidor front-end, que pode fazer parte de um servidor front-end 230. As consultas cache 320 podem incluir consultas recentes. Em 710, um conjunto atualizado
5 de consultas é computado. O conjunto atualizado de consultas pode incluir consultas a partir das consultas comuns 330 que seriam atribuídas ao servidor front-end 230 com base, por exemplo, em uma operação de hashing e mapeamento que opera para orientar as mesmas consultas e as mesmas porções das
10 consultas para o mesmo servidor front-end 230. O conjunto atualizado de consultas pode ser computado pelo servidor front-end 230. Em 720, o conjunto atualizado de consultas pode ser armazenado na estrutura de dados 310 com as consultas cache 320. Isso serve para atualizar as consultas cache
15 320.

Em 730, uma consulta no conjunto atualizado de consultas porém não nas consultas cache 320 (uma consulta faltante) é determinada. Se nenhuma consulta casar com esse critério, o método prossegue para buscar uma consulta que
20 efetivamente case com os critérios. O método continua de modo que a consulta faltante seja adicionada às consultas cache 320. Em 740, a consulta faltante é adicionada a uma fila.

O método pode prosseguir para 750, onde o sistema
25 é analisado para determinar se o sistema está levemente carregado. Cada vez que um servidor front end 230 envia uma solicitação para o sistema back-end 240, a resposta a partir do sistema back-end 240 pode incluir realimentação que pode

ser utilizada para determinar o estado atual do back-end 240, que pode incluir carga back-end atual. Por exemplo, a realimentação pode incluir um número médio de solicitações que estão em uma fila esperando para serem processadas pelo
 5 back-end 240 ou uma qualidade de uma resposta que o back-end 240 é capaz de obter durante um limite de tempo de processamento de consulta. O servidor front-end 230 pode então utilizar um algoritmo para determinar se o back-end atual 240 deve ser considerado como intensamente ou levemente carregado com base na realimentação. Se o sistema estiver levemente
 10 carregado, então o método pode prosseguir para 760.

Em 760, a partir de 740 ou 750, a consulta faltante é removida da fila. Em 770, a consulta faltante pode ser fornecida ao banco de dados back-end 240. Em 780, o banco de
 15 dados back-end 240 atualiza as consultas cache 320 com a consulta faltante. A consulta faltante não mais está "faltando" nas consultas cache 320.

Como pode ser reconhecido, as modalidades reveladas podem ser implementadas como um todo ou em parte em um
 20 ou mais sistemas ou dispositivos de computação. A figura 1 ilustra os componentes funcionais de um exemplo de um sistema de computação 100, no qual aspectos podem ser incorporados ou postos em prática. Como utilizado aqui, os termos "sistema de computação", "sistema de computador", e "computador"
 25 tador" se referem a qualquer máquina, sistema ou dispositivo que compreende um processador capaz de executar ou de outro modo processar código de programa e/ou dados. Os exemplos de sistemas de computação incluem, sem nenhuma limitação pre-

tendida, computadores pessoais (PCs), minicomputadores, computadores de grande porte, clientes finos, PCs de rede, servidores, estações de trabalho, computadores laptop, computadores portáteis, eletrônica programável pelo consumidor, consoles de multimeios, consoles de jogos, receptores de satélite, set-top boxes, caixas eletrônicos automatizados, casas de jogos, telefones móveis, assistentes pessoais digitais (PDAs), e qualquer outro sistema ou máquina baseado em processador. Os termos "código de programa" e "código" se referem a qualquer conjunto de instruções que são executadas ou de outro modo processadas por um processador. Código de programa e/ou dados podem ser implementados na forma de rotinas, programas, objetos, módulos, estruturas de dados e similares que executam funções específicas.

Observa-se que os exemplos acima foram fornecidos simplesmente para fins de explicação e não são de modo algum para serem interpretados como limitadores. Embora a invenção tenha sido descrita com referência a várias modalidades, entende-se que as palavras que foram utilizadas aqui são palavras de descrição e ilustração, em vez de palavras de limitação. Além disso, embora as modalidades tenham sido descritas aqui com referência a meios específicos, materiais e exemplos, as modalidades não pretendem ser limitadas aos detalhes revelados aqui; em vez disso, as modalidades estendem a todas as estruturas funcionalmente equivalentes, métodos e usos, como estão compreendidas no escopo das reivindicações apenas.

REIVINDICAÇÕES

1. Método de fornecer resultados de consulta, o método sendo **CARACTERIZADO** por compreender:

obter uma porção de uma consulta;

5 atribuir a porção da consulta a um servidor (230);

obter outras consultas iniciando com a porção da consulta; e

obter resultados de algumas ou todas as outras consultas com base na porção da consulta.

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

computar uma representação da porção da consulta;

em que a atribuição da porção da consulta a um servidor (230) compreende atribuir a porção da consulta a um
15 servidor (230) com base na representação.

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

determinar se a porção da consulta casa com uma entrada predeterminada; e

20 atribuir caracteres seguintes como a porção da consulta.

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

armazenar consultas cache (320) e um conjunto de
25 consultas comuns (330) em uma estrutura de dados (310);

em que a obtenção de outras consultas iniciando com a porção da consulta compreende obter outras consultas iniciando com a porção da consulta a partir da estrutura de

dados (310).

5. Método, de acordo com a reivindicação 4, **CARACTERIZADO** por compreender ainda:

modificar a estrutura de dados (310) para refletir
5 alterações feitas nas consultas cache (320) e alterações
feitas no conjunto de consultas comuns (330).

6. Método, de acordo com a reivindicação 4, **CARACTERIZADO** por compreender ainda:

armazenar as consultas cache (320) e o conjunto de
10 consultas comuns (330) de acordo com uma frequência de uso;
e

exibir os resultados de cada uma das outras consultas iniciando com a porção da consulta em uma ordem decedente da frequência de uso.

15 7. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que a obtenção dos resultados de cada uma das outras consultas iniciando com a porção da consulta compreende obter resultados de cada uma das outras consultas a partir de cache (410) do servidor (230).

20 8. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que obter resultados de cada uma das outras consultas iniciando com a porção da consulta compreende obter resultados de cada uma das outras consultas a partir de um servidor back-end (240).

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

fornecer um resultado vazio se as outras consultas iniciando com a porção da consulta não puderem ser obtidas.

10. Sistema de consulta (300), **CARACTERIZADO** por compreender:

um servidor de distribuidor (220) que recebe uma porção de uma consulta;

5 uma estrutura de dados (310) que armazena consultas cache (320) e um conjunto de consultas comuns (330);

uma estrutura de resultados (340) que provê resultados para as consultas cache (320) e o conjunto de consultas comuns (330); e

10 um servidor front-end (230) que consulta a estrutura de dados (310) e a estrutura de resultados (340) para fornecer resultados relacionados com base na porção da consulta.

11. Sistema, de acordo com a reivindicação 10,
15 **CARACTERIZADO** por compreender ainda um banco de dados backend (240).

12. Sistema, de acordo com a reivindicação 10,
CARACTERIZADO pelo fato de que o servidor de distribuidor (220) atribui a porção da consulta ao servidor front-end
20 (230).

13. Sistema, de acordo com a reivindicação 10,
CARACTERIZADO pelo fato de que a porção da consulta é um prefixo da consulta.

14. Método de construir um conjunto de consultas
25 comuns (330), o método sendo **CARACTERIZADO** por compreender:

compilar entradas deletadas a partir de um cache (410) de um servidor front-end (230);

formar um conjunto de consultas comuns (330) com-

preendidas das entradas deletadas, em que as entradas deletadas são a partir de uma faixa de tempo predeterminada; e distribuir o conjunto de consultas comuns (330) para outro servidor front-end (230).

5 15. Método, de acordo com a reivindicação 14, **CARACTERIZADO** pelo fato de que a distribuição do conjunto de consultas comuns (330) para outro servidor front-end (230) compreende distribuir o conjunto de consultas comuns (330) para múltiplos servidores front-end (230).

10 16. Método, de acordo com a reivindicação 14, **CARACTERIZADO** por compreender ainda:

gerar consultas adicionais baseadas em uma lista predeterminada; e

adicionar as consultas adicionais ao conjunto de
15 consultas comuns (330).

17. Método, de acordo com a reivindicação 14, **CARACTERIZADO** por compreender ainda:

computar um conjunto atualizado de consultas para um servidor front-end (230), em que o conjunto atualizado de
20 consultas é compreendido de consultas atribuídas no conjunto de consultas comuns (330) que são atribuídas ao servidor front-end (230); e

construir uma estrutura de dados (310) para o servidor front-end (230), em que o conteúdo da estrutura de da-
25 dos é compreendido do conjunto atualizado de consultas e consultas cache (320).

18. Método, de acordo com a reivindicação 17, **CARACTERIZADO** por compreender ainda:

computar consultas faltantes, em que as consultas faltantes pertencem em um cache (410) do servidor front-end (230); e

inserir as consultas faltantes em uma fila.

5 19. Método, de acordo com a reivindicação 18, **CARACTERIZADO** por compreender ainda:

remover as consultas faltantes da fila;

enviar as consultas faltantes a partir da fila para um banco de dados back-end (240); e

10 inserir as consultas faltantes no cache (410) do servidor front-end (230).

20. Método, de acordo com a reivindicação 17, **CARACTERIZADO** por compreender ainda:

15 determinar se uma consulta expirada que expirou a partir do cache (410) do servidor (230) pertence ao conjunto atualizado de consultas;

inserir a consulta expirada em uma fila,

remover a consulta expirada a partir da fila,

20 enviar a consulta expirada a partir da fila para um banco de dados back-end (240); e

inserir a consulta expirada no cache (410) do servidor front-end (230).

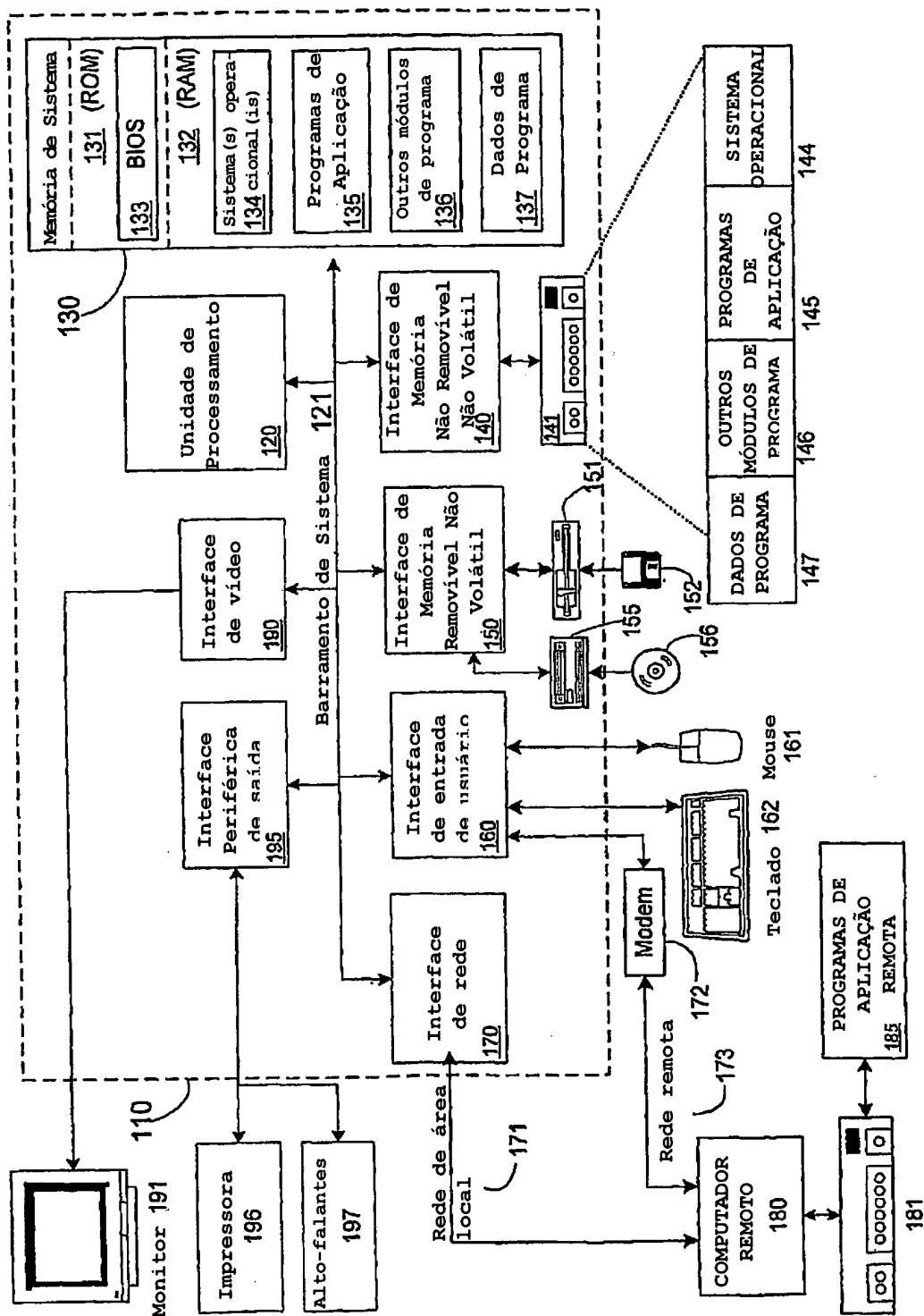


FIG. 1

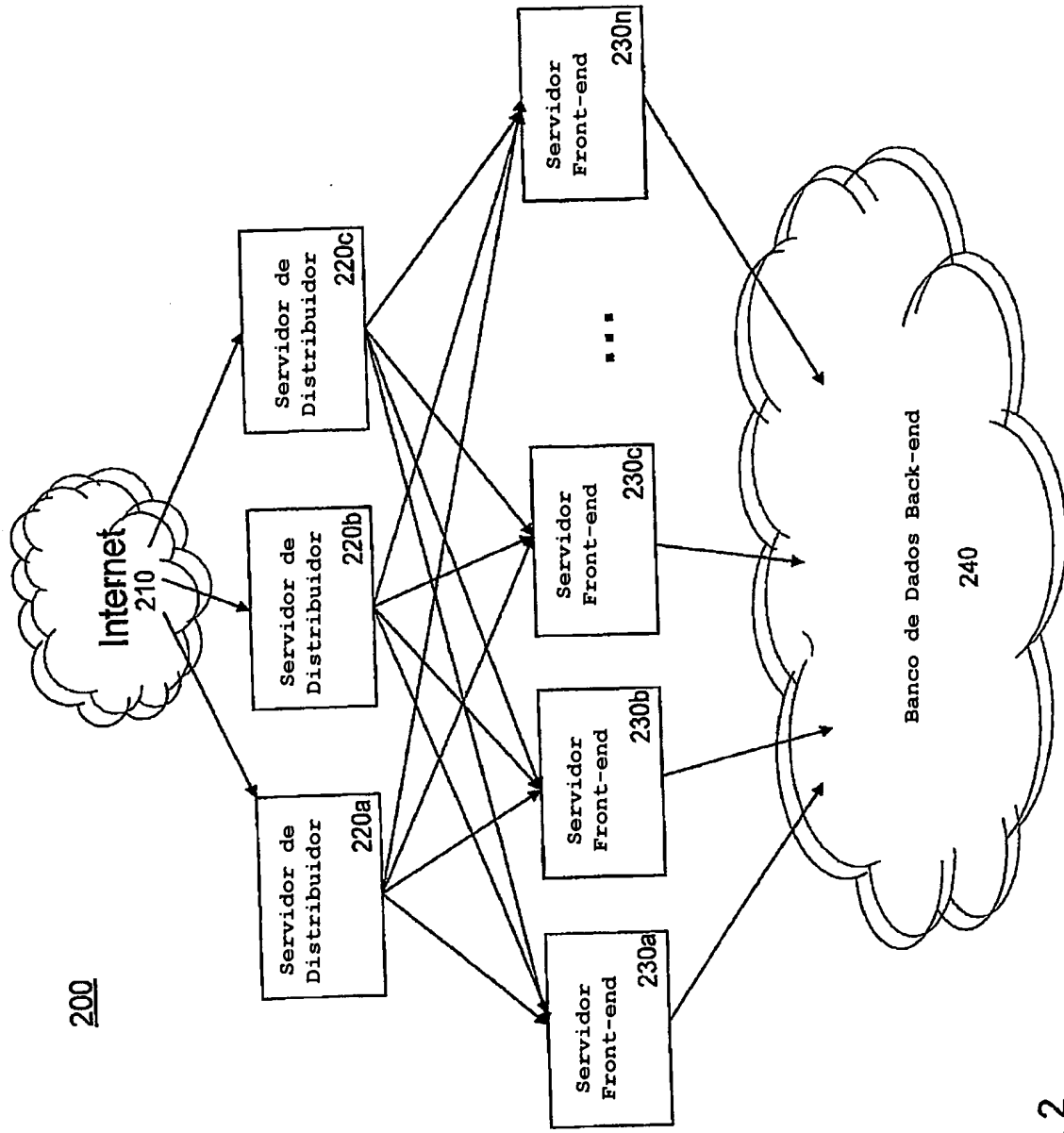


FIG. 2

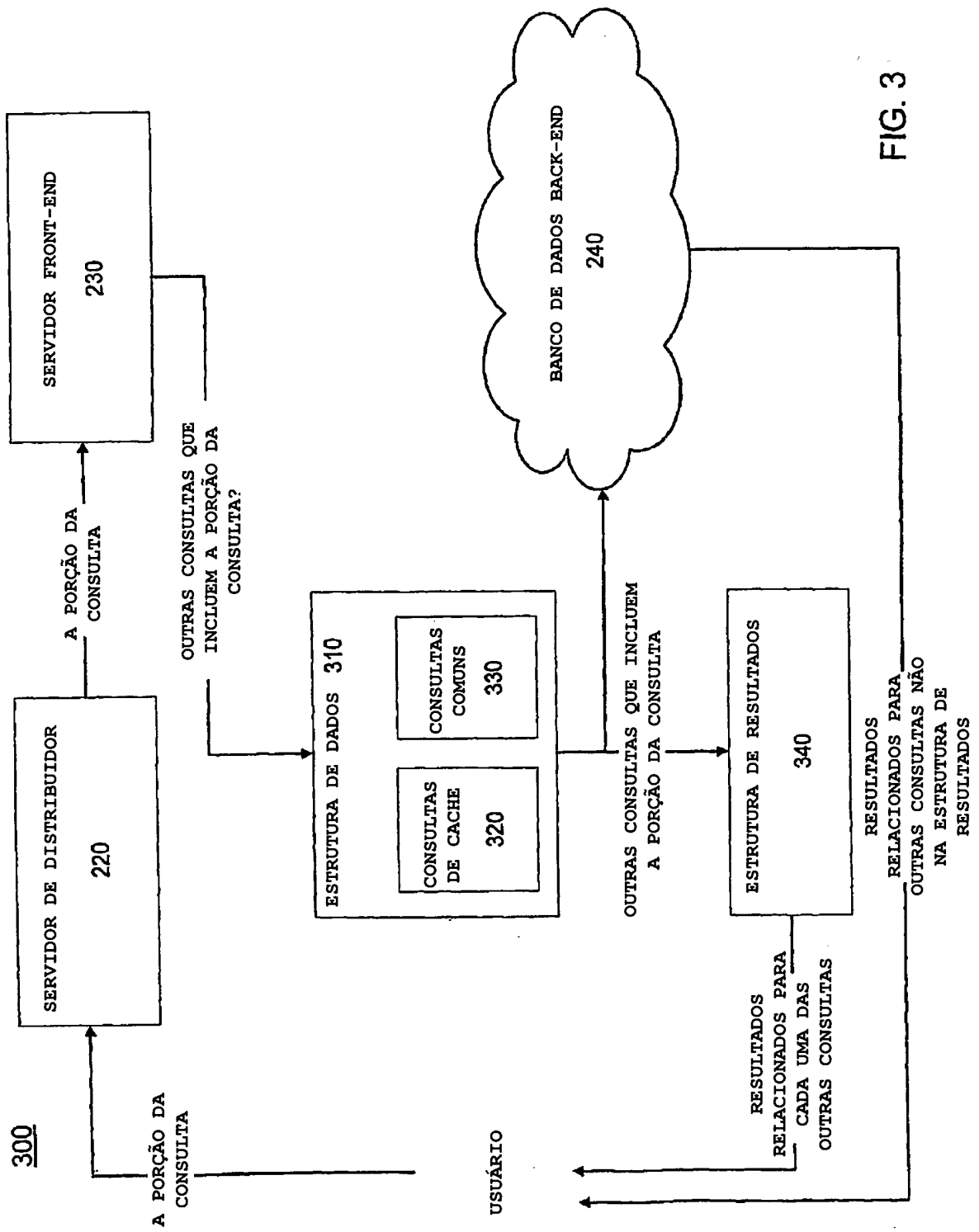


FIG. 3

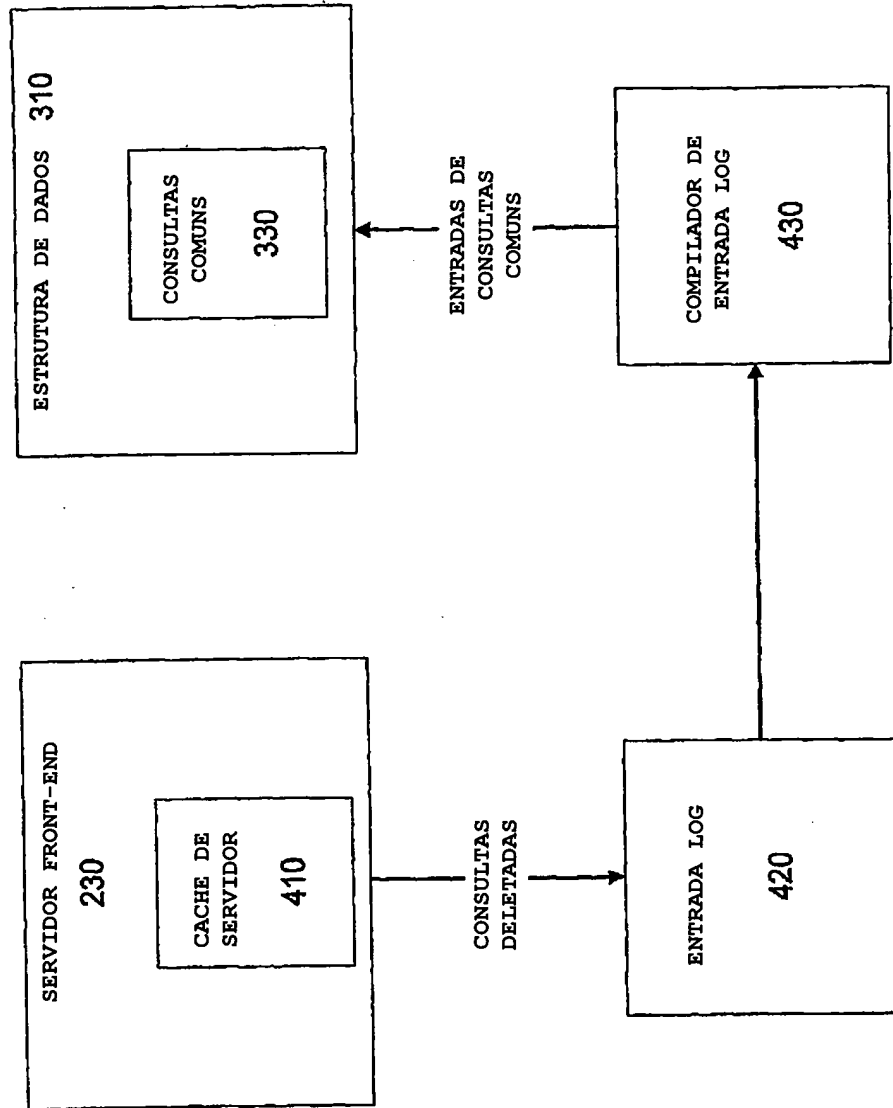
400

FIG. 4

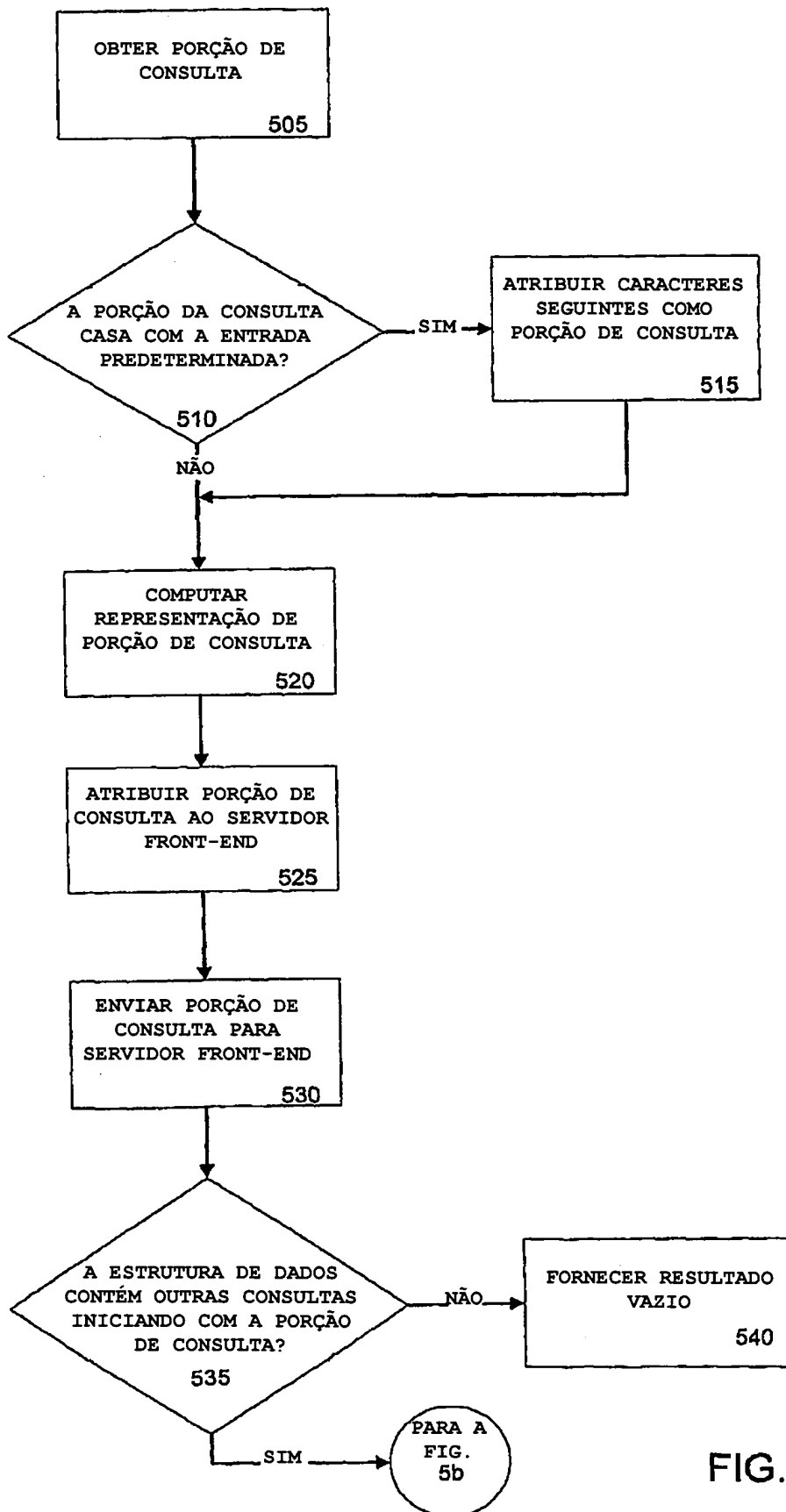


FIG. 5a

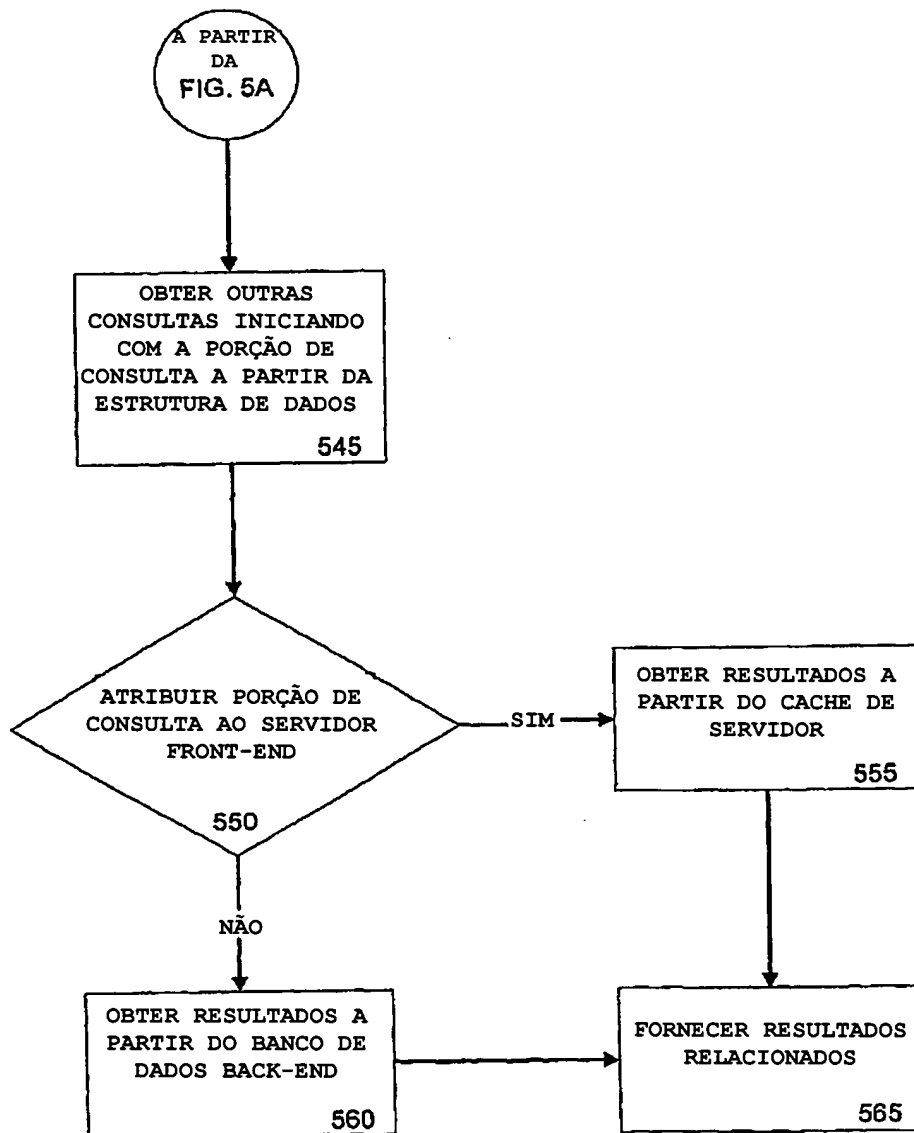


FIG. 5b

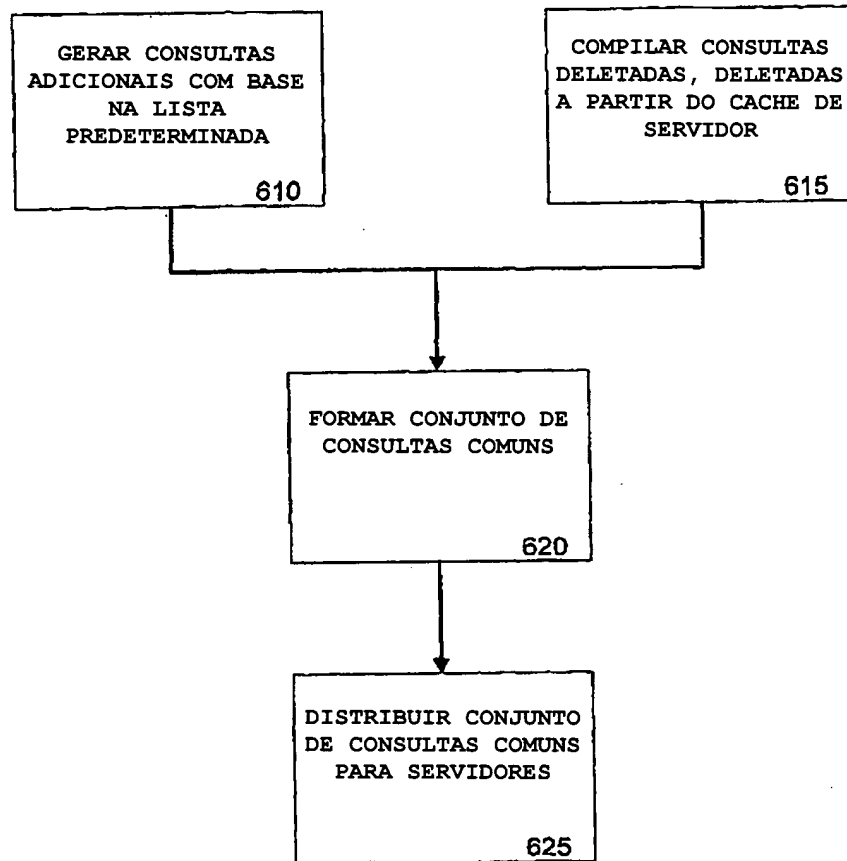


FIG. 6

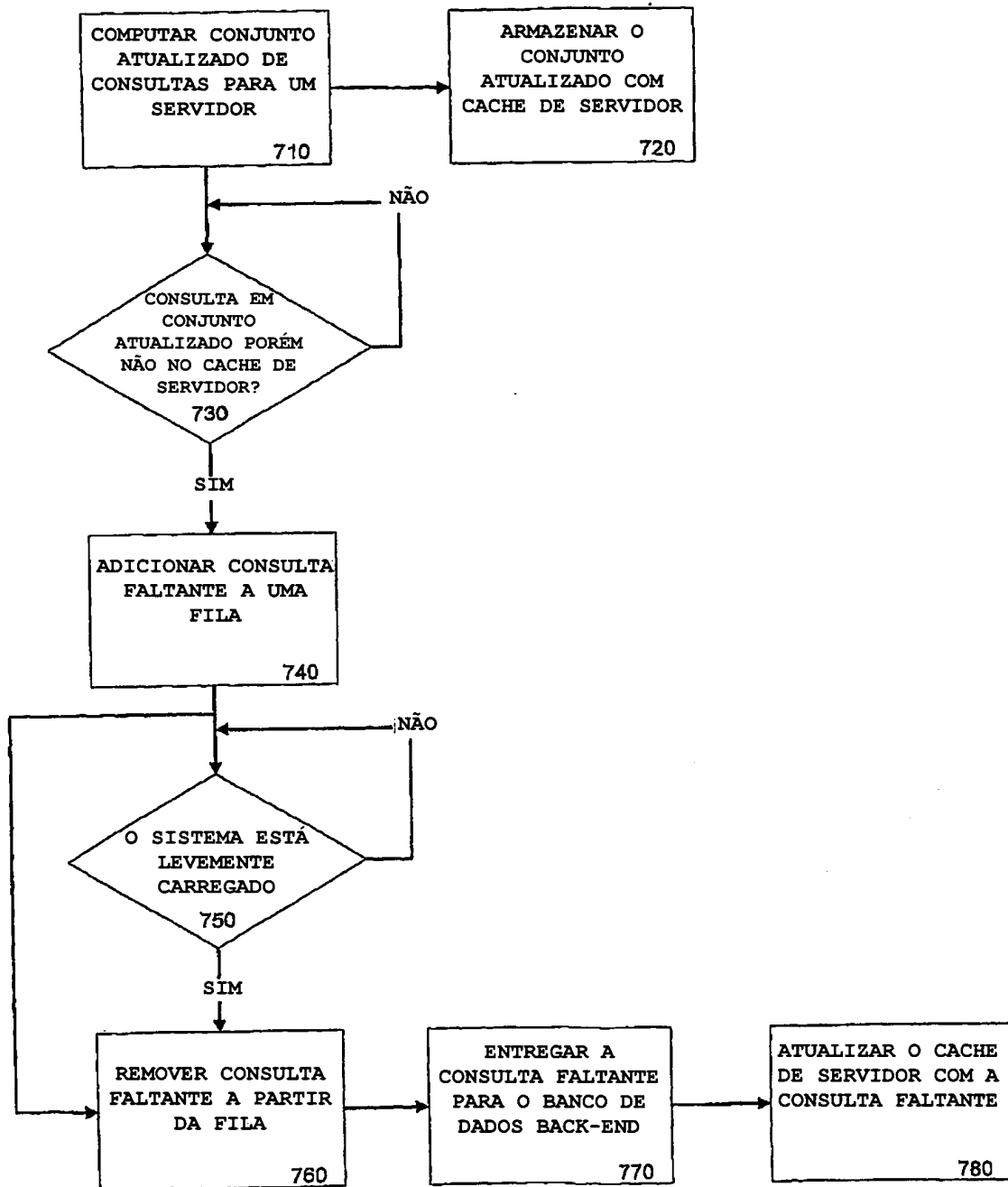


FIG. 7

RESUMO

"SERVIR RESULTADOS DE CONSULTA CACHED COM BASE EM
UMA PORÇÃO DE CONSULTA"

As modalidades consideram um sistema e método para
5 obter resultados relacionados para uma porção de uma consul-
ta e para gerar um conjunto atualizado de consultas para um
cache de um servidor. Outras consultas iniciando com a por-
ção da consulta podem ser identificadas e obtidas a partir
de uma estrutura de dados que inclui um cache de servidor e
10 um conjunto de consultas comuns. Após as outras consultas
serem identificadas, os resultados para as outras consultas
são obtidos a partir do cache de servidor ou a partir de um
banco de dados back-end. Um conjunto de consultas comuns,
que pode incluir consultas deletadas e adicionais, pode ser
15 utilizado para gerar o conjunto atualizado de consultas para
o servidor. Tanto as consultas faltantes como as consultas
deletadas que podem pertencer ao servidor com base em uma
função de atribuição são inseridas em uma fila, que é poste-
riormente fornecida ao cache do servidor. A transferência
20 pode ocorrer durante baixa atividade ou estado inativo.