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

(21) BR 112020019697-5 A2

(22) Data do Depósito: 27/03/2019

(43) Data da Publicação Nacional: 05/01/2021

(54) Título: TÉCNICAS PARA PROJETO DE REPARO ANTIENTROPIA ESCALONADO

(51) Int. Cl.: G06F 9/48; G06F 16/27.

(30) Prioridade Unionista: 27/03/2018 US 62/648,907.

(71) Depositante(es): NETFLIX, INC..

(72) Inventor(es): VINAY CHELLA; JOSEPH LYNCH; AJAY UPADHYAY.

(86) Pedido PCT: PCT US2019024417 de 27/03/2019

(87) Publicação PCT: WO 2019/191320 de 03/10/2019

(85) Data da Fase Nacional: 28/09/2020

(57) Resumo: TÉCNICAS PARA PROJETO DE REPARO ANTIENTROPIA ESCALONADO. Várias modalidades da invenção revelada neste documento fornecem técnicas para realizar procedimentos de reparos antientropia distribuídos através de uma pluralidade de nós em uma rede de base de dados distribuída. Um nó incluído em uma pluralidade de nós dentro da rede de base de dados distribuída determina, antes de todos os outros nós incluídos na pluralidade de nós, que um primeiro procedimento de reparo antientropia terminou. O nó determina que um segundo procedimento de reparo antientropia está pronto para começar. O nó gera um escalonamento para executar uma ou mais operações associadas com o segundo procedimento de reparo antientropia. O nó grava o escalonamento em uma estrutura de dados de escalonamentos de reparos compartilhados para iniciar o segundo procedimento de reparo antientropia através de múltiplos nós incluídos na pluralidade de nós. Cada um dos nós incluídos na pluralidade de nós então realiza um reparo de nó com base no escalonamento.

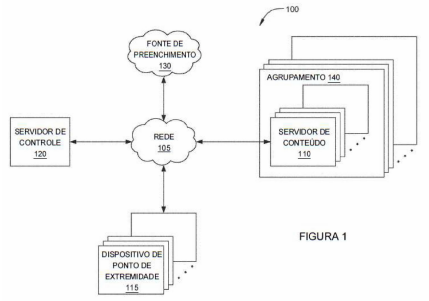


FIGURA 1

“TÉCNICAS PARA PROJETO DE REPARO ANTIENTROPIA ESCALONADO”

REFERÊNCIA CRUZADA PARA PEDIDOS RELACIONADOS

[001] Este pedido reivindica benefício do pedido de patente Estados Unidos 62/648.907 (Número de Protocolo Representante NETF0192USL), depositado em 27 de março de 2018, o qual está incorporado a este documento pela referência.

ANTECEDENTES DA INVENÇÃO

Campo da Invenção

[002] A presente invenção de uma maneira geral diz respeito a sistemas de computadores distribuídos e, mais especificamente, a técnicas para projeto de reparo antientropia escalonado.

Descrição da Técnica Relacionada

[003] Em certas redes de bases de dados distribuídas, coleções de dados são copiadas e armazenadas em múltiplos dispositivos de computação, referidos neste documento como “nós”, por toda a rede de base de dados distribuída. As múltiplas cópias de uma coleção de dados são referidas neste documento como “réplicas”. Armazenar réplicas por toda a rede de base de dados distribuída fornece redundância contra perda de dados, pelo que se uma das réplicas se tornar corrompida então as réplicas remanescentes podem ser acessadas a fim de recuperar os dados correspondentes. Adicionalmente, quando réplicas são armazenadas em nós que são distintos geograficamente, um usuário solicitando acesso a uma coleção particular de dados pode recuperar os dados do nó que está mais perto do dispositivo de computação do usuário. Como um resultado, latência entre solicitar acesso a uma réplica e recuperar a

réplica pode ser reduzida.

[004] Ao longo do tempo, dados em uma réplica podem se tornar inconsistentes com dados em outras réplicas correspondentes. Como um exemplo, um usuário acessando uma réplica particular em um nó pode modificar uma parte da réplica e então armazenar a réplica modificada de volta no mesmo nó. Como um resultado, a réplica modificada fica inconsistente com as outras réplicas distribuídas por toda a rede de base de dados distribuída. Para corrigir tais inconsistências, um dos nós é designado como um coordenador de reparo antientropia. O coordenador de reparo antientropia designado então analisa as diferenças entre um subconjunto de réplicas com relação às inconsistências e atualiza o subconjunto de réplicas de maneira que o subconjunto de réplicas se torne consistente. Este processo de analisar e atualizar réplicas é referido neste documento como um "procedimento de reparo antientropia", ou mais simplesmente como um "reparo". Com reparos convencionais, usuários manualmente escalonam reparos como um processo por lote off-line por meio de ferramentas de reparo externas à base de dados distribuída. Como um resultado, com procedimentos de reparos antientropia convencionais, usuários são responsáveis por projetar e manter procedimentos de reparos antientropia para os nós dentro da rede de base de dados distribuída.

[005] Um problema com a abordagem indicada acima para reparar nós é que soluções e opções de reparo são difíceis para usuários entenderem e executarem de modo apropriado. Procedimentos de reparos antientropia simples podem ser efetivos para redes de bases de dados distribuídas que têm

um número relativamente pequeno de nós, mas tais procedimentos de reparos antientropia simples podem falhar com uma rede de base de dados distribuída que tem dezenas de milhares de nós. De modo similar, procedimentos de reparos antientropia que podem ser efetivos para nós que armazenam réplicas que são relativamente pequenas em tamanho podem falhar para nós que armazenam réplicas que são relativamente grandes, e vice-versa. Como um resultado, projetar e manter procedimentos de reparos antientropia pode ser difícil para uma rede de base de dados distribuída que tem um número oscilante de nós e/ou tamanhos diferentes de réplicas.

[006] Um outro problema com a abordagem indicada acima para reparar nós é que quando o coordenador de reparo antientropia e/ou um ou mais outros nós falham enquanto um reparo está em andamento, o progresso de reparo completo pode ser perdido. Como um resultado, o procedimento de reparo antientropia precisa ser reiniciado desde o início, resultando em uma perda de horas ou dias de trabalho de reparo que foi realizado antes da falha. Adicionalmente, o procedimento de reparo antientropia externo pode não ter visibilidade total do estado da base de dados distribuída e do progresso de reparo no tempo de falha. Consequentemente, o procedimento de reparo antientropia pode ter dificuldade ao determinar quais ações específicas são necessárias para reiniciar o reparo. Também um outro problema com a abordagem indicada acima para reparar nós é que reparos frequentemente exigem uso significativo de armazenamento de disco, recursos de unidade central de processamento (CPU) e largura de banda de rede, o que pode reduzir

significativamente o desempenho da rede de base de dados distribuída.

[007] Tal como o exposto acima ilustra, o que é necessário na arte são técnicas mais efetivas para reparar nós em uma rede de base de dados distribuída.

SUMÁRIO DA INVENÇÃO

[008] Várias modalidades do presente pedido estabelecem um método implementado por computador para realizar procedimentos de reparos antientropia distribuídos através de uma pluralidade de nós em uma rede de base de dados distribuída. O método inclui determinar, por um primeiro nó incluído em uma pluralidade de nós e antes de todos os outros nós incluídos na pluralidade de nós, que um primeiro procedimento de reparo antientropia terminou. O método inclui adicionalmente determinar, pelo primeiro nó, que um segundo procedimento de reparo antientropia está pronto para começar. O método inclui adicionalmente gerar um escalonamento para executar uma ou mais operações associadas com o segundo procedimento de reparo antientropia. O método inclui adicionalmente gravar o escalonamento em uma estrutura de dados de escalonamentos de reparos compartilhados para iniciar o segundo procedimento de reparo antientropia através de múltiplos nós incluídos na pluralidade de nós.

[009] Outras modalidades da presente invenção incluem, sem limitação, um meio legível por computador incluindo instruções para realizar um ou mais aspectos das técnicas divulgadas, assim como um dispositivo de computação para realizar um ou mais aspectos das técnicas divulgadas.

[010] Pelo menos uma vantagem técnica das técnicas

divulgadas em relação à técnica anterior é que procedimentos de reparos antientropia escalam automaticamente para o número de nós e para o tamanho variado das réplicas em uma rede de base de dados distribuída. Como um resultado, o procedimento de reparo antientropia se adapta a cada subconjunto de dados armazenados em um nó particular a fim de reduzir a quantidade de acesso a disco, recursos de CPU e largura de banda de rede consumida, em relação às técnicas convencionais. Além disso, usuários são aliviados de projetar e manter manualmente procedimentos de reparos antientropia à medida que o número de nós e o tamanho de réplicas mudam ao longo do tempo. Uma outra vantagem técnica das técnicas divulgadas em relação à técnica anterior é que o procedimento de reparo antientropia é distribuído através dos nós. Adicionalmente, uma vez que os nós falhos reiniciam execução, o procedimento de reparo antientropia pode reiniciar relativamente na mesma condição de quando a falha ocorreu. Consequentemente, menos ou nenhum progresso de reparo é perdido quando um ou mais outros nós falham durante um reparo, em relação às abordagens de técnica anterior. Estas vantagens técnicas representam um ou mais melhoramentos tecnológicos em relação às abordagens de técnica anterior.

DESCRIÇÃO RESUMIDA DOS DESENHOS

[011] A fim de que a maneira na qual os recursos relatados acima da invenção possa ser entendida detalhadamente, uma descrição mais particular da invenção, resumida acima, pode ser feita por meio de referência para modalidades, algumas das quais estão ilustradas nos

desenhos anexos. É para ser notado, entretanto, que os desenhos anexos ilustram somente modalidades típicas desta invenção e, portanto, não devem ser considerados como limitantes de seu escopo, já que a invenção pode admitir outras modalidades igualmente efetivas.

[012] A figura 1 ilustra uma infraestrutura de rede usada para distribuir conteúdo para servidores de conteúdo e para dispositivos de pontos de extremidades, de acordo com várias modalidades da invenção;

a figura 2 é uma ilustração mais detalhada do servidor de conteúdo que pode ser implementado em combinação com a infraestrutura de rede da figura 1, de acordo com várias modalidades da presente invenção;

a figura 3 é uma ilustração mais detalhada do servidor de controle que pode ser implementado em combinação com a infraestrutura de rede da figura 1, de acordo com várias modalidades da presente invenção;

a figura 4 é uma ilustração mais detalhada do dispositivo de ponto de extremidade que pode ser implementado em combinação com a infraestrutura de rede da figura 1, de acordo com várias modalidades da presente invenção; e

as figuras 5A-5B mostram um fluxograma de etapas de método para realizar procedimentos de reparos antientropia distribuídos através de uma pluralidade de servidores de conteúdo em uma rede de base de dados distribuída, de acordo com várias modalidades da presente invenção.

DESCRIÇÃO DETALHADA

[013] Na descrição seguinte, inúmeros detalhes específicos são expostos para fornecer um entendimento mais

completo da presente invenção. Entretanto, estará aparente para os versados na técnica que modalidades da presente invenção podem ser praticadas sem um ou mais destes detalhes específicos.

Vista Geral de Sistema

[014] A figura 1 ilustra uma infraestrutura de rede 100 usada para distribuir conteúdo para os servidores de conteúdo 110 e para os dispositivos de pontos de extremidades 115, de acordo com várias modalidades da invenção. Tal como mostrado, a infraestrutura de rede 100 inclui os agrupamentos 140, um servidor de controle 120 e os dispositivos de pontos de extremidades 115, cada um dos quais é conectado por meio de uma rede de comunicações 105. A rede 105 pode ser qualquer ambiente adequado para capacitar comunicações entre sistemas de computadores remotos ou locais e dispositivos de computação, incluindo, sem limitação, LANs sem fio e com fio e WANs (Redes de Área Ampla) baseadas em Internet.

[015] Cada dispositivo de ponto de extremidade 115 inclui, sem limitação, um dispositivo de computação que pode ser um computador pessoal, console de videogame, assistente digital pessoal, telefone móvel, dispositivo móvel ou qualquer outro dispositivo adequado para implementar um ou mais aspectos da presente invenção. Cada dispositivo de ponto de extremidade 115 se comunica com um ou mais servidores de conteúdo 110 por meio da rede 105 para transferir conteúdo, tal como dados textuais, dados gráficos, dados de áudio, dados de vídeo e outros tipos de dados. Os servidores de conteúdo 110 também são referidos neste documento como “caches”, “nós de computação” ou mais

simplesmente como “nós”. O conteúdo transferível, também referido neste documento como um “arquivo”, é então apresentado para um usuário de um ou mais dispositivos de pontos de extremidades 115. Em várias modalidades, os dispositivos de pontos de extremidades 115 podem incluir sistemas de computadores, aparelhos conversores de sinais, computador móvel, smartphones, tablets, sistemas de videogames de consoles e portáteis, gravadores de vídeo digital (DVRs), reprodutores de DVD, TVs digitais conectadas, dispositivos dedicados de fluxo contínuo de mídia (por exemplo, o conversor de sinais Roku®) e/ou qualquer outra plataforma de computação tecnicamente exequível que tenha conectividade de rede e que seja capaz de apresentar conteúdo, tal como texto, imagens, vídeo e/ou conteúdo de áudio, para um usuário.

[016] Cada agrupamento 140 inclui um ou mais servidores de conteúdo 110. Tal como descrito adicionalmente neste documento, cada agrupamento 140 é capaz de executar independentemente um procedimento de reparo antientropia em relação ao servidor de conteúdo 110 incluído no agrupamento particular 140. Cada servidor de conteúdo 110 inclui, sem limitação, um dispositivo de armazenamento que pode ser um sistema autônomo de armazenamento conectado em rede (NAS), uma rede de área de armazenamento (SAN), um agrupamento ou “fazenda” de dispositivos de armazenamento, uma arquitetura de armazenamento distribuído, ou qualquer outro dispositivo adequado para implementar um ou mais aspectos da presente invenção. Adicionalmente ou de forma alternativa, cada servidor de conteúdo 110 pode incluir, sem limitação, um dispositivo de computação com um subsistema de

armazenamento que pode ser um servidor autônomo, um agrupamento ou "fazenda" de servidores, um ou mais dispositivos de rede, ou qualquer outro dispositivo adequado para implementar um ou mais aspectos da presente invenção. Adicionalmente, cada servidor de conteúdo 110 pode incluir, sem limitação, um servidor de rede e uma base de dados, e pode ser configurado para se comunicar com o servidor de controle 120 para determinar a localização e disponibilidade de vários arquivos que são rastreados e gerenciados pelo servidor de controle 120.

[017] De uma maneira mais geral, cada servidor de conteúdo 110 armazena grupos de arquivos chamados de tabelas. Cada tabela, por sua vez, consiste de partições, onde uma partição é uma unidade de dados de tamanho arbitrário. O tamanho de uma partição pode ser pequeno, tal como um único par chave-valor, ou grande, tal como um Gigabyte de dados. Cada servidor de conteúdo 110 adicionalmente pode se comunicar com uma fonte de preenchimento 130 e com um ou mais outros servidores de conteúdo 110 a fim de "preencher" cada servidor de conteúdo 110 com cópias de vários arquivos. Além disso, os servidores de conteúdo 110 podem responder às solicitações para arquivos recebidas dos dispositivos de pontos de extremidades 115. Os arquivos podem então ser distribuídos pelo servidor de conteúdo 110 ou por meio de uma rede de distribuição de conteúdo mais ampla. Em algumas modalidades, os servidores de conteúdo 110 capacitam autenticação de usuários (por exemplo, usando um nome de usuário e senha) a fim de acessar arquivos armazenados nos servidores de conteúdo 110.

[018] O servidor de controle 120 pode incluir, sem limitação, um dispositivo de computação que pode ser um servidor autônomo, um agrupamento ou "fazenda" de servidores, um ou mais dispositivos de rede, ou qualquer outro dispositivo adequado para implementar um ou mais aspectos da presente invenção. Embora somente um único servidor de controle 120 esteja mostrado na figura 1, em várias modalidades múltiplos servidores de controle 120 podem ser implementados para rastrear e gerenciar arquivos.

[019] Em várias modalidades, a fonte de preenchimento 130 pode incluir um serviço de armazenamento on-line (por exemplo, Amazon® Simple Storage Service, Google® Cloud Storage, etc.) no qual um catálogo de arquivos, incluindo milhares ou milhões de arquivos, é armazenado e acessado a fim de preencher os servidores de conteúdo 110. Embora somente uma única fonte de preenchimento 130 esteja mostrada na figura 1, em várias modalidades múltiplas fontes de preenchimento 130 podem ser implementadas para solicitações de serviços para arquivos.

Realização de Procedimentos de Reparos Antientropia Distribuídos Através de um Agrupamento

[020] A figura 2 é um diagrama de blocos de um servidor de conteúdo 110 que pode ser implementado em combinação com a infraestrutura de rede 100 da figura 1, de acordo com várias modalidades da presente invenção. Tal como mostrado, o servidor de conteúdo 110 inclui, sem limitação, um processador 204, um disco de sistema 206, uma interface de dispositivos de entrada/saída (I/O) 208, uma interface de rede 210, uma interconexão 212 e uma memória de sistema 214.

[021] O processador 204 é incluído para ser representativo de uma única CPU, múltiplas CPUs, uma única CPU tendo múltiplos núcleos de processamento e outros mais. O processador 204 é configurado para recuperar e executar instruções de programação, tais como uma aplicação de servidor 217 e uma aplicação de reparo 219, armazenadas na memória de sistema 214. De modo similar, o processador 204 é configurado para armazenar dados de aplicação (por exemplo, bibliotecas de software) e recuperar dados de aplicação da memória de sistema 214. A interconexão 212 é configurada para facilitar transmissão de dados, tais como instruções de programação e dados de aplicação, entre o processador 204, o disco de sistema 206, a interface de dispositivos de entrada/saída 208, a interface de rede 210 e a memória de sistema 214. A interface de dispositivos de entrada/saída 208 é configurada para receber dados de entrada dos dispositivos de entrada/saída 216 e transmitir os dados de entrada para o processador 204 por meio da interconexão 212. Por exemplo, os dispositivos de entrada/saída 216 podem incluir uma ou mais teclas, um teclado, um mouse e/ou outros dispositivos de entrada. A interface de dispositivos de entrada/saída 208 é configurada adicionalmente para receber dados de saída do processador 204 por meio da interconexão 212 e transmitir os dados de saída para os dispositivos de entrada/saída 216.

[022] O disco de sistema 206 pode incluir uma ou mais unidades de discos rígidos, dispositivos de armazenamento de estado sólido ou dispositivos de armazenamento similares. O disco de sistema 206 é configurado para

armazenar dados não voláteis tais como os arquivos 218 (por exemplo, arquivos de áudio, arquivos de vídeo, legendas, arquivos de aplicação, bibliotecas de software, etc.). Os arquivos 218, ou, mais especificamente, partições e/ou registros associados com um ou mais arquivos 218, podem então ser recuperados por um ou mais dispositivos de pontos de extremidades 115 por meio da rede 105. Em algumas modalidades, a interface de rede 210 é configurada para operar em conformidade com o padrão Ethernet.

[023] A memória de sistema 214 inclui, sem limitação, uma aplicação de servidor 217, uma aplicação de reparo 219 e um armazenamento de dados 221. O armazenamento de dados 221 inclui um armazenamento de dados de aplicação de servidor, onde dados são armazenados e recuperados pela aplicação de servidor 217. Em algumas modalidades, o armazenamento de dados 221 pode incluir também um armazenamento de dados de reparos, onde dados são armazenados e recuperados pela aplicação de reparo 219. Adicionalmente ou de forma alternativa, o armazenamento de dados de reparos pode residir em um armazenamento de dados que é externo ao servidor de conteúdo 110. Em tais casos, a aplicação de reparo 219 pode acessar o armazenamento externo de dados de reparos por meio da interface de rede 210.

[024] A aplicação de servidor 217 é configurada para atender solicitações para uma ou mais partições em um ou mais arquivos 218 recebidas do dispositivo de ponto de extremidade 115 e de outros servidores de conteúdo 110. Quando a aplicação de servidor 217 recebe uma solicitação para uma ou mais partições dentro de um ou mais arquivos

218, a aplicação de servidor 217 recupera os arquivos correspondentes 218 do disco de sistema 206 e transmite a(s) partição(s) materializada(s) para um dispositivo de ponto de extremidade 115 ou para um servidor de conteúdo 110 por meio da rede 105. Ao realizar estas operações a aplicação de servidor 217 armazena dados no armazenamento de dados 221 e recupera dados do mesmo.

[025] A aplicação de reparo 219, quando executada pelo processador 204, realiza uma ou mais operações associadas com o servidor de conteúdo 110 da figura 1, tal como descrito adicionalmente neste documento. Ao realizar estas operações a aplicação de reparo 219 armazena dados no armazenamento de dados 221 e recupera dados do mesmo.

[026] Em operação, a aplicação de reparo 219 em execução no processador 204 gerencia a execução dos procedimentos de reparos antientropia por meio de atualizações de estados de reparos. Em uma modalidade, a aplicação de reparo 219 gerencia e coordena a execução dos procedimentos de reparos antientropia por meio de uma estrutura de dados que inclui quatro tipos de tabelas de estados. Estas quatro tabelas de estados são referidas neste documento como repair_process, repair_sequence, repair_status e repair_hook_status. Para um procedimento de reparo antientropia particular, existe um registro de estado repair_process e um registro repair_sequence para cada servidor de conteúdo 110 incluído no agrupamento 140. Adicionalmente, para um procedimento de reparo antientropia particular executando em um servidor de conteúdo 110 particular, existe um registro repair_status por tabela de partições e uma tabela de estado repair_hook_status para

cada servidor de conteúdo 110 incluído no agrupamento 140. Em várias modalidades, cada uma das tabelas de estados repair_process, repair_sequence, repair_status e repair_hook_status pode existir em cada agrupamento ou em um armazenamento de dados mestre, em qualquer combinação tecnicamente exequível. Em tais modalidades, a tabela de estado repair_status mestre e uma tabela de estado repair_hook_status mestre podem ser divididas em múltiplas seções, onde cada seção corresponde a um servidor de conteúdo 110 diferente incluído no agrupamento 140.

[027] A primeira tabela de estado é a tabela de estado repair_process, a qual inclui parâmetros para executar um procedimento de reparo antientropia no nível do agrupamento total 140. A segunda tabela de estado é a tabela de estado repair_sequence, a qual define o escalonamento para realizar o procedimento de reparo antientropia através dos servidores de conteúdo 110 incluídos no agrupamento 140. O escalonamento é na forma de uma sequência ou ordem na qual os nós realizam um reparo de nó. Em particular, a tabela de estado repair_sequence rastreia o status de cada servidor de conteúdo 110 no agrupamento 140 e indica quais servidores de conteúdo 110 realizaram um reparo de nó durante o procedimento de reparo antientropia atual. A terceira tabela de estado é a tabela de estado repair_status, a qual rastreia o status de reparo para elementos específicos incluídos em cada servidor de conteúdo 110 incluído no agrupamento 140. Tal como descrito adicionalmente neste documento, estes elementos específicos incluem cada uma das partições e subfaixas de dados sofrendo reparo dentro de um servidor de conteúdo

correspondente 110. A quarta tabela de estado é a tabela de estado repair_hook_status, a qual rastreia o status de operações associadas com procedimentos pós-reparo realizados pelos servidores de conteúdo 110. Tais operações associadas com procedimentos pós-reparo são referidas neste documento como “ganchos pós-reparo”.

[028] De uma maneira geral, ganchos pós-reparo incluem várias tarefas de manutenção realizadas por um servidor de conteúdo 110 uma vez que o servidor de conteúdo 110 e todos os servidores de conteúdo 110 vizinhos relevantes tenham realizado um reparo de nó associado com o procedimento de reparo antientropia. Tal como usado neste documento, os vizinhos de um determinado servidor de conteúdo 110 são o conjunto dos outros servidores de conteúdo 110 incluídos no agrupamento que armazena uma ou mais partições em comum com o determinado servidor de conteúdo 110. De uma maneira geral, um servidor de conteúdo 110 não pode realizar ganchos pós-reparo até que todos os outros servidores de conteúdo 110 que compartilham os mesmos dados tenham realizado reparos de nós. Esta regra impede que dados necessários para reparo de nó dos servidores de conteúdo 110 vizinhos sejam eliminados antes da conclusão desses reparos de nós. Após um servidor de conteúdo 110 particular e todos os servidores de conteúdo 110 vizinhos relevantes terem realizado um reparo de nó, o servidor de conteúdo 110 particular pode armazenar uma ou mais partições que foram necessárias durante reparo, mas que não são mais necessárias uma vez que o reparo esteja completo. Portanto, um gancho de limpeza elimina estas partições que não são mais necessárias.

[029] Adicionalmente, após um servidor de conteúdo 110 particular e todos os servidores de conteúdo 110 vizinhos relevantes terem realizado um reparo de nó, o servidor de conteúdo 110 particular pode armazenar os dados reparados em um modo ineficiente. Mais especificamente, durante reparo de nó, o servidor de conteúdo 110 lê, grava e copia blocos de dados em e de várias localizações de armazenamento do disco de sistema 206. Como um resultado, uma vez que o reparo de nó esteja completo, partes de uma partição particular podem ser armazenadas em localizações aleatórias no disco de sistema 206, e/ou a partição pode incluir seções de espaço de armazenamento livre ou não usado incorporado à partição. Recuperar partições armazenadas dessa maneira pode ser demorado, porque o disco de sistema 206 pode precisar recuperar dados de localizações de armazenamento que estão distantes umas das outras e pode precisar saltar seções de espaço de armazenamento livre ou não usado. Como um resultado de armazenar dados em um modo ineficiente como este, um dispositivo de ponto de extremidade 115 solicitando dados do servidor de conteúdo 110 após o reparo estar completo pode experimentar latência aumentada, em relação às solicitações feitas antes do reparo. Para reduzir latência de acesso, o processador 204 realiza um processo de compactação para armazenar partições no disco de sistema 206 em um modo linear e contíguo sem seções internas de espaço de armazenamento livre ou não usado. Em particular, o processador 204 realiza um gancho de compactação para armazenar os dados em um modo mais eficiente, diminuindo desse modo latência de solicitação.

[030] Além disso, quaisquer ganchos pós-reparo adicionais podem ser definidos por um usuário de tal maneira que um servidor de conteúdo 110 particular realiza estes ganchos pós-reparo adicionais após o servidor de conteúdo 110 particular e todos os servidores de conteúdo 110 vizinhos relevantes terem realizado um reparo de nó. Um gancho de reparo definido por usuário como este pode transmitir uma mensagem para uma aplicação de monitoramento (não mostrada) de que o servidor de conteúdo 110 completou um reparo de nó. Neste modo, a aplicação de monitoramento rastreia o progresso de cada servidor de conteúdo 110 no agrupamento 140 que está realizando um reparo e computa métricas correspondentes, tal como o tempo consumido durante reparo do agrupamento 140.

[031] A aplicação de reparo 219 executa através dos servidores de conteúdo 110 em rede de base de dados distribuída, onde nenhum servidor de conteúdo 110 é designado antecipadamente como um coordenador de reparo antientropia. Em algumas modalidades, cada servidor de conteúdo 110 em um agrupamento 140 realiza a aplicação de reparo 219 periodicamente, tal como uma vez a cada dois minutos. Ao executar a aplicação de reparo 219, o servidor de conteúdo 110 primeiro recupera dados da tabela de estado repair_process e/ou da tabela de estado repair_sequence para determinar se um procedimento de reparo antientropia já está em andamento. Se os dados da tabela de estado repair_process e/ou da tabela de estado repair_sequence indicarem que um procedimento de reparo antientropia está em andamento, então o servidor de conteúdo 110 determina, a partir de dados na tabela de estado repair_sequence, se

esse servidor de conteúdo 110 é o próximo em sequência para realizar o reparo de nó. Se os dados na tabela de estado repair_sequence indicarem que o servidor de conteúdo 110 é o próximo em sequência, então o servidor de conteúdo 110 repara as réplicas e tabelas de estados associadas e sai do procedimento de reparo. Se os dados na tabela de estado repair_sequence indicarem que o servidor de conteúdo 110 não é o próximo em sequência, então o servidor de conteúdo 110 determina se o servidor de conteúdo 110 está autorizado para realizar ganchos pós-reparo relacionados com o servidor de conteúdo 110. O servidor de conteúdo 110 está autorizado para realizar ganchos pós-reparo se os dados na tabela de estado repair_sequence indicarem que o servidor de conteúdo 110 e todos os servidores de conteúdo 110 vizinhos completaram reparo de nó. Se o servidor de conteúdo 110 estiver autorizado, então o servidor de conteúdo 110 realiza os ganchos pós-reparo e sai do procedimento de reparo. De outro modo, o servidor de conteúdo 110 sai do procedimento de reparo sem realizar os ganchos pós-reparo.

[032] Se, por outro lado, o servidor de conteúdo 110 determinar que um procedimento de reparo antientropia não está em andamento, então o servidor de conteúdo 110 determina se um novo procedimento de reparo antientropia deve ser iniciado. De uma maneira geral, um novo procedimento de reparo antientropia deve ser iniciado se o procedimento de reparo antientropia anterior foi completado e não existirem restrições a respeito de iniciar novo procedimento de reparo antientropia. Por exemplo, início de um novo procedimento de reparo antientropia pode ser

restringido para uma hora do dia especificada como fora do horário de pico, onde fora do horário de pico representa uma faixa especificada de tempo quando a carga nos servidores de conteúdo 110 é suposta estar abaixo de um nível limiar específico. Se um tempo corrente cair na faixa especificada de tempo designado para operações de reparos, então um novo procedimento de reparo antientropia pode ser iniciado. Se um novo procedimento de reparo antientropia não puder ser iniciado, o servidor de conteúdo 110 sai do procedimento de reparo. De outro modo, o servidor de conteúdo 110 tenta iniciar um novo procedimento de reparo antientropia. A tentativa é bem sucedida se o servidor de conteúdo 110 puder adquirir um bloqueio de dados na tabela de estado repair_procedure. Se o servidor de conteúdo 110 for malsucedido ao adquirir um bloqueio de dados na tabela de estado repair_procedure, então um outro servidor de conteúdo 110 adquiriu o bloqueio e está iniciando um novo procedimento de reparo antientropia. Se, entretanto, o servidor de conteúdo 110 for bem sucedido ao adquirir o bloqueio, o servidor de conteúdo 110 preenche a tabela de estado repair_process com parâmetros de nível de agrupamento. Além disso, o servidor de conteúdo 110 preenche a tabela de estado repair_sequence com uma nova sequência para os servidores de conteúdo 110 realizarem o novo procedimento de reparo antientropia. O servidor de conteúdo 110 então sai do procedimento de reparo.

[033] De uma maneira geral, o único momento em que os servidores de conteúdo 110 são coordenados uns com os outros é quando um servidor de conteúdo 110 tenta iniciar um novo procedimento de reparo antientropia, preenche as

tabelas de estados repair_process e repair_sequence descritas acima, e gera uma nova sequência que define a sequência ou ordem na qual os servidores de conteúdo 110 realizam reparos de nós relacionados com o novo procedimento de reparo antientropia. Quando um servidor de conteúdo 110 tenta iniciar um novo procedimento de reparo antientropia, o servidor de conteúdo 110 solicita um bloqueio na tabela de estado repair_process de nível de agrupamento. Neste modo, múltiplos servidores de conteúdo 110 são impedidos de gravar simultaneamente novas tabelas de estados repair_process e repair_sequence para o próximo procedimento de reparo antientropia. Como um resultado, qualquer servidor de conteúdo 110 dentro do agrupamento 140 pode adquirir o bloqueio na tabela de estado repair_process e iniciar o próximo procedimento de reparo antientropia.

[034] Quando o servidor de conteúdo 110 adquire o bloqueio, o servidor de conteúdo 110 grava a tabela de estado repair_sequence máquina por máquina como uma operação em lote. Em uma gravação de operação em lote, todas as gravações são bem sucedidas ou nenhuma das gravações é executada. Em uma operação em lote, o servidor de conteúdo 110 é impedido de gravar uma sequência parcial para a tabela de estado repair_sequence e então falhar, resultando em uma tabela de estado repair_sequence que inclui sequências parciais relacionadas com múltiplos procedimentos de reparos antientropia. Adicionalmente, outros servidores de conteúdo 110 são impedidos de acessar a tabela de estado repair_sequence até que a operação em lote seja completada ou falhe. Como um resultado, outros servidores de conteúdo 110 têm a garantia de que a nova

tabela de estado repair_sequence está completa e consistente. Ao gravar a tabela de estado repair_sequence em uma operação em lote, outros servidores de conteúdo 110 são impedidos de ler uma tabela de estado repair_sequence gravada parcialmente e, como um resultado, realizar reparos de nós na sequência errada.

[035] Em resumo, qualquer servidor de conteúdo 110 pode iniciar um novo procedimento de reparo antientropia por meio de dois estados de coordenação: (1) adquirir um bloqueio e gravar para a tabela de estado repair_process; e (2) gravar a tabela de estado repair_sequence em uma operação em lote. A não ser estes dois estados de coordenação, os servidores de conteúdo 110 realizam o reparo de nó independentemente sem precisar de coordenação com os outros servidores de conteúdo 110 ou adquirir bloqueios a fim de prosseguir.

[036] Adicionalmente, cada servidor de conteúdo 110 divide o conjunto de partições escalonadas para reparo em subfaixas. Uma determinada subfaixa pode representar uma única partição, múltiplas partições ou uma parte de uma partição. Cada servidor de conteúdo 110 automaticamente divide e funde partições em subfaixas a fim de satisfazer um tempo de conclusão alvo para completar um reparo. Como um exemplo, um procedimento de reparo antientropia pode ter como um alvo completar um reparo em não mais que trinta minutos por subfaixa. Cada subfaixa é dimensionada automaticamente para satisfazer este tempo de conclusão alvo, com base no número de partições e no tamanho das partições armazenadas no servidor de conteúdo 110. De uma maneira geral, um conjunto ideal de tabelas de estados

repair_status para um servidor de conteúdo 110 em um agrupamento 140 que inclui uma quantidade grande de partições pequenas seria diferente de um conjunto ideal de tabelas de estados repair_status para um servidor de conteúdo 110 em um agrupamento 140 que inclui uma quantidade pequena de partições grandes.

[037] A fim de otimizar o conjunto de tabelas de estados repair_status, os servidores de conteúdo 110 realizam um procedimento de faixa de divisão adaptável. Isso seleciona automaticamente subfaixas com base no número de partições, no tamanho das partições e em outras métricas. Como um exemplo, considerar um agrupamento 140 que inclui um milhão de partições, onde cada partição é um par chave-valor. A fim de descobrir diferenças nas partições através de múltiplos servidores de conteúdo 110, cada servidor de conteúdo 110 incluído no agrupamento 140 calcula um cache criptográfico de cada um dos componentes e gera uma árvore de cache, referida neste documento como uma "árvore de Merkle". Em uma árvore de Merkle, cada folha inclui um valor de dispersão dos dados em uma partição ou parte de uma partição correspondente. Dadas duas árvores de Merkle, uma para um conjunto de partições em cada um de dois servidores de conteúdo 110, se folhas correspondentes nas duas árvores de Merkle tiverem o mesmo valor de dispersão, então os dados de fontes correspondentes são idênticos. Cada servidor de conteúdo 110 analisa a árvore de Merkle através dos servidores de conteúdo 110 para descobrir diferenças nos valores de dispersão armazenados em uma árvore de Merkle versus as dispersões armazenadas em uma outra árvore de Merkle. Uma diferença como esta

corresponde a uma diferença nos dados de fonte representados pelos valores de dispersão. Entretanto, gerar uma única árvore de Merkle para um milhão de partições pode consumir mais memória do que está disponível para realizar o procedimento de reparo antientropia.

[038] Um modo de resolver este problema é restringir o tamanho da árvore de Merkle a um número razoável de valores de dispersão. Entretanto, restringir o tamanho da árvore de Merkle neste modo pode resultar em uma árvore de Merkle onde cada valor de dispersão representa um número grande de partições. Se dois valores de dispersão correspondentes forem diferentes, então todas as partições representadas pelo valor de dispersão precisam ser comparadas a fim de descobrir a inconsistência de dados.

[039] Consequentemente, os servidores de conteúdo 110 realizam um procedimento de faixa de divisão adaptável, pelo qual o reparo de nó é dividido em subfaixas, cada subfaixa com uma árvore de Merkle diferente. Cada árvore de Merkle pode ter uma resolução de um para um, em que cada valor de dispersão representa uma única partição. As múltiplas árvores de Merkle são computadas concorrentemente. O procedimento de faixa de divisão adaptável determina o número, tamanho e resolução de cada árvore de Merkle com base em um tempo de conclusão alvo por árvore de Merkle, tal como trinta minutos.

[040] Este procedimento de faixa de divisão adaptável ajusta automaticamente as árvores de Merkle com base nas partições subjacentes no agrupamento 140. Por exemplo, um primeiro agrupamento 140 pode ter um milhão de partições onde cada partição é pequena. Este agrupamento 140

repararia rapidamente porque cada valor de dispersão pode ser computado rapidamente, e cada partição pode ser comparada rapidamente. Um segundo agrupamento 140 pode ter um milhão de partições onde cada partição tem um Gigabyte. Este agrupamento 140 repararia lentamente porque computar cada valor de dispersão pode levar um tempo longo, e comparar cada partição pode igualmente levar um tempo longo. Ao selecionar adaptativamente uma faixa de divisão apropriada, o tempo de conclusão alvo de trinta minutos por subfaixa pode ser alcançado. Esta seleção adaptativa de faixas de divisão permite recuperação mais rápida após uma falha do servidor de conteúdo 110.

[041] Em algumas modalidades, um procedimento de reparo antientropia pode ajustar automaticamente trabalho de reparo com base em dados históricos relacionados com um ou mais procedimentos de reparos antientropia executados anteriormente. Em tais modalidades, um servidor de conteúdo 110 que está iniciando um novo procedimento de reparo antientropia analisa os dados na tabela de estado repair_status, tal como reportado por cada servidor de conteúdo 110. Estes dados identificam cada subfaixa para cada servidor de conteúdo 110, o tamanho da subfaixa e a quantidade de tempo consumido para reparar essa subfaixa. As tabelas de estados para o novo procedimento de reparo antientropia podem então ser adaptadas com base nestes dados a fim de satisfazer um tempo de conclusão alvo. Como um exemplo, se o procedimento de reparo antientropia anterior é realizado em menos tempo que o tempo de conclusão alvo, então a faixa de divisão para novo procedimento de reparo antientropia pode ser baseada em um

tempo de conclusão alvo que é o dobro do tempo de conclusão real do procedimento de reparo antientropia anterior. Por outro lado, se o procedimento de reparo antientropia anterior é realizado em mais tempo que o tempo de conclusão alvo, então a faixa de divisão para novo procedimento de reparo antientropia pode ser baseada em um tempo de conclusão alvo que é uma porcentagem fixada menor que o tempo de conclusão real do procedimento de reparo antientropia anterior. Neste modo, configurações para procedimentos de reparos antientropia subsequentes são adaptadas às mudanças na natureza das partições ao longo do tempo.

[042] Um conjunto de opções de reparo é agora descrito. Estas opções de reparo fornecem controles de usuário para guiar configuração e execução de procedimentos de reparos antientropia, incluindo como o procedimento de faixa de divisão adaptável é realizado. As opções de reparo estão mostradas na Tabela 1 a seguir.

Tabela 1

Opção	Descrição	Padrão
tipo	O tipo de reparo a executar. Escolhas são completo e incremental.	completo
trabalhadores	Número de trabalhadores de reparo para executar reparos. Ajustar este parâmetro antes de ajustar paralelismo para melhorar velocidade de reparo.	máx(1, # núcleos / 2)

paralelismo	O grau de paralelismo ao calcular árvores de Merkle. Escolhas são sequencial, paralelo, dc_parallel.	sequencial
ganchos	O conjunto de ganchos pós-reparo que devem executar em um servidor de conteúdo uma vez que o servidor de conteúdo e todos os vizinhos tenham concluído reparo. Ganchos pós-reparo disponíveis incluem limpeza, compactação e opções conectáveis adicionais.	[“limpeza”]
split_range	O número de partições para dividir a tabela de estado repair_status para o propósito de subfaixa. Valor pode ser um valor inteiro, <integer>_dry_run ou adaptável. Se <integer>_dry_run, então o algoritmo adaptável inclui um campo split_range_adaptive no	adaptável

	mapa de configuração da tabela de estado repair_status para inspecionar antes de capacitação adaptável. Se adaptável, então o agrupamento determina automaticamente um valor ideal.	
interrepair_delay_minutes	O atraso entre execuções de reparos em minutos. O padrão depende do tipo de reparo sendo escalonado. Usar isto para reparos “deslocados” uns dos outros, por exemplo, se for conhecido que reparos levaram 2 dias, e o alvo é completar um reparo a cada 10 dias, então estabelecer um atraso de ~3 dias pode ser um bom valor.	completo -> 1.440 (24 horas) incremental -> 0
process_timeout_seconds	O número de segundos a esperar para que um outro servidor de conteúdo faça transição de estado (por exemplo,	1.800 (30 minutos)

	gerar uma sequência, iniciar reparo, tornar saudável, etc.). Estabelecer isto se muitos cancelamentos forem observados.	
repair_timeout_seconds	O número de segundos a esperar para que um único reparo de subfaixa termine. Estabelecer isto se muitos cancelamentos forem observados.	14.400 (4 horas)

[043] A opção de reparo "tipo" especifica o tipo de procedimento de reparo antientropia a executar. Em um reparo completo, os servidores de conteúdo 110 realizam um reparo em todas as partições. Em um reparo incremental, os servidores de conteúdo 110 realizam um reparo somente nos dados que tenham mudado. O tipo de reparo padrão é um reparo completo.

[044] A opção de reparo "trabalhadores" especifica o número de núcleos de processador, também referidos neste documento como "trabalhadores", nos quais é para executar o reparo. O número padrão de trabalhadores é o máximo de um núcleo e de o número de núcleos disponíveis dividido por dois. Este número padrão de trabalhadores executa reparo em pelo menos um núcleo de processador, mas não mais que na metade do número de núcleos disponíveis.

[045] A opção de reparo "paralelismo" especifica o grau de paralelismo empregado ao calcular árvores de Merkle para

executar o reparo. Para um agrupamento 140 que inclui três servidores de conteúdo 110 em cada um de três centros de dados distintos geograficamente, sequencial faz com que cada um dos nove servidores de conteúdo 110 construa uma árvore de Merkle sequencialmente, onde somente um servidor de conteúdo 110 constrói árvores de Merkle a qualquer momento. Paralelo faz com que os três servidores de conteúdo 110 em um determinado centro de dados construam simultaneamente árvores de Merkle. Centros de dados constroem árvores de Merkle sequencialmente, onde somente os servidores de conteúdo 110 em um centro de dados constroem árvores de Merkle a qualquer momento. Dc_parallel faz com que todos os nove servidores de conteúdo 110 em todos os três centros de dados construam simultaneamente árvores de Merkle. O padrão paralelismo é sequencial. Embora sequencial possa ser a mais lenta das três opções, sequencial é também a mais conservativa em termos de preservação de dados.

[046] A opção de reparo “ganchos” especifica o conjunto de ganchos pós-reparo a executar uma vez que o servidor de conteúdo 110 correspondente e os servidores de conteúdo 110 vizinhos tenham completado reparo de nó. Ganchos pós-reparo disponíveis incluem “limpeza”, “compactação” e quaisquer outros ganchos pós-reparo fornecidos pelo usuário. O gancho padrão é limpeza.

[047] A opção de reparo “split_range” especifica um número de partições para dividir a tabela de estado repair_status para o propósito de subfaixa. Se split_range especificar um número inteiro ‘n’, então o reparo de nó executa de tal maneira que cada partição é dividida em ‘n’

subfaixas. Se split_range especificar um número inteiro ‘n’ seguido por “_dry_run”, então o reparo de nó executa de tal maneira que cada partição é dividida em ‘n’ subfaixas e dados de diagnósticos adicionais são armazenados na tabela de estado repair_status. Se split_range especificar “adaptável”, então o reparo de nó executa um processo de faixa de divisão adaptável, tal como descrito adicionalmente neste documento. O split_range padrão é adaptável.

[048] A opção de reparo “interrepair_delay_minutes” especifica um número inteiro de minutos para atrasar entre a conclusão de um procedimento de reparo antientropia e o início do procedimento de reparo antientropia subsequente. Um valor interrepair_delay_minutes de “total” indica que o início do reparo subsequente ocorre 1.440 minutos (24 horas) após a conclusão do procedimento de reparo antientropia anterior. Um valor interrepair_delay_minutes de “incremental” indica que o início do procedimento de reparo antientropia subsequente ocorre imediatamente após a conclusão do procedimento de reparo antientropia anterior sem atraso.

[049] A opção de reparo “process_timeout_seconds” especifica um número inteiro de segundos de espera para que um outro servidor de conteúdo 110 mude de um estado para um outro estado. O process_timeout_seconds padrão é de 1.800 segundos (30 minutos).

[050] A opção de reparo “repair_timeout_seconds” especifica um número inteiro de segundos de espera para que um único reparo de subfaixa seja completado. O repair_timeout_seconds padrão é de 14.400 segundos (4

horas).

[051] Em várias modalidades, dois comandos suplementares, referidos neste documento como “nodetools”, estão disponíveis para verificar ou mudar o estado do procedimento de reparo executando atualmente. Os dois comandos suplementares incluem um comando repairstatus para verificar o histórico de reparo de um servidor de conteúdo 110 e um comando repairctl para controlar reparo do agrupamento 140. Cada um destes comandos nodetool é agora discutido com detalhes adicionais.

[052] A estrutura do comando repairstatus está mostrada na Tabela 2 a seguir:

001 NOME

002 nodetool repairstatus - Imprimir informação de histórico de reparo

003 SINOPSE

004 nodetool ... repairstatus

005 [--repair-id <repairid>] [--node-id <nodeid>]

006 [--keyspace <keyspace>] [--table <table>]

007 OPÇÕES

008 --repair-id <repairid>

009 Mostrar exatamente repairstatus para o identificador de reparo especificado

010 Se não especificado, então mostrar o status de reparo mais recente

011 --node-id <nodeid>

012 O identificador de nó para o qual recuperar status de reparo

013 --keyspace <keyspace>

```
014         Restringir ao keyspace especificado
015   --table <table>
016         Restringir à tabela de estado especificada.
```

Tabela 2

[053] As linhas 001 e 002 especificam o nome “nodetool repairstatus” como o comando para imprimir informação de histórico de reparo. As linhas 003 a 006 ilustram uma sinopse do comando repairstatus, incluindo quatro opções de comando tais como mostradas nas linhas 005 e 006 e descritas mais detalhadamente nas linhas 007 a 016. As linhas 008-010 ilustram a opção de comando repair-id. Se repair-id for especificado, então o comando repairstatus retorna o status para o identificador de reparo especificado. Se repair-id não for especificado, então o comando repairstatus retorna o status do reparo completo. As linhas 011-012 ilustram a opção de comando node-id. O comando repairstatus retorna o status para o identificador de nó especificado. As linhas 013-014 ilustram a opção de comando keyspace. O comando repairstatus retorna o status para o keyspace especificado. As linhas 015-016 ilustram a opção de comando de tabela. O comando repairstatus retorna o status para a tabela de estado especificada.

[054] Por exemplo, o comando “nodetool repairstatus” sem quaisquer opções de comando retornaria uma visão global do status de reparo mais recente. O comando “nodetool repairstatus --repair-id 12” retornaria o status de reparo do reparo com um identificador de 12. O comando “nodetool repairstatus --node-id 73ab7e49” retornaria o status de reparo do nó com um identificador de 73ab7e49.

[055] A estrutura do comando repairctl está mostrada na

Tabela 3 a seguir:

001 NOME

002 nodetool repairctl - Controlar reparo no agrupamento

003 SINOPSE

004 nodetool ... repairctl

005 [--stop-cluster] [--start-cluster]

006 [--cancel-running]

007 OPÇÕES

008 --stop-cluster

009 Pausar reparo no agrupamento total

010 Isto não cancela reparos executando ativos

011 --cancel-execution

012 Parar imediatamente quaisquer reparos executando em um determinado servidor de conteúdo

013 --start-cluster

014 Reiniciar reparo no agrupamento total.

Tabela 3

[056] As linhas 001 e 002 especificam o nome “nodetool repairctl” como o comando para controlar reparo em um agrupamento 140. As linhas 003 a 006 ilustram uma sinopse do comando repairctl, incluindo três opções de comando tais como mostradas nas linhas 005 e 006 e descritas mais detalhadamente nas linhas 007 a 014. As linhas 008-010 ilustram a opção de comando stop-cluster. Se presente, a opção stop-cluster faz com que o reparo no agrupamento total 140 seja pausado, sem cancelar a execução de reparos ativos. As linhas 011-012 ilustram a opção de comando cancel-execution. O cancel-execution interrompe imediatamente quaisquer reparos executando em um

determinado servidor de conteúdo 110. As linhas 013-014 ilustram a opção de comando start-cluster. O comando start-cluster reinicia reparo no agrupamento 140.

[057] Por exemplo, o comando "nodetool repairctl --stop-cluster" pausaria reparo no agrupamento 140 sem cancelar quaisquer reparos ativos. O comando "nodetool repairctl --start-cluster" reiniciaria reparo no agrupamento 140 pausado. O comando "nodetool repairctl --cancel-execution" pararia execução de qualquer reparo em um determinado servidor de conteúdo 110. Como um resultado, os servidores de conteúdo 110 remanescentes começariam a reparar. Esta opção de comando interromperia um servidor de conteúdo 110 impedido de prosseguir para não impedir outros servidores de conteúdo 110 de realizarem um reparo de nó. O comando "nodetool repairctl --stop-cluster --cancel-execution" pausaria reparo no agrupamento 140 e pararia execução de qualquer reparo em um determinado servidor de conteúdo 110.

[058] Em algumas modalidades, um procedimento de reparo antientropia pode suportar escalonamentos de reparos exclusivos. Em tais modalidades, reparos podem ser restringidos para iniciar e/ou executar somente durante certos períodos do tempo. Por exemplo, cada centro de dados em um agrupamento 140 pode especificar horas "fora de pico" que representam uma faixa especificada de tempo designado para operações de reparos. Se um tempo corrente cair na faixa especificada de tempo designado para operações de reparos, então um novo procedimento de reparo antientropia pode ser iniciado. De uma maneira geral, as técnicas descritas neste documento não consomem de modo excessivo

recursos de CPU, memória ou largura de banda de rede. Mesmo assim, alguns usuários podem desejar restringir o reparo fora do horário de pico a fim de minimizar impacto em outras solicitações de dados e tráfego de rede. Em outras modalidades, um usuário pode ser capaz de suspender um reparo ou um ou mais ganchos pós-reparo por um período de tempo especificado.

[059] Em algumas modalidades, um procedimento de reparo antientropia pode suportar múltiplos escalonamentos de reparos. Em tais modalidades, reparos podem ser de tipos diferentes e podem ser executados concorrentemente. O repair_sequence pode incluir um campo adicional que especifica qual o tipo de reparo e/ou opções de configuração para aplicar ao reparo atual, com base no tipo de reparo. Como um exemplo, reparos totais e reparos incrementais podem ser suportados. Em um reparo completo, os servidores de conteúdo 110 realizam um reparo em todas as partições. Em um reparo incremental, os servidores de conteúdo 110 realizam um reparo somente nos dados que foram mudados. Portanto, um reparo completo pode ser executado uma vez por mês enquanto que um reparo incremental pode ser executado uma vez por dia. Cada reparo incremental repararia partições que diferem por causa de gravações. Cada reparo completo repararia adicionalmente partições que diferem por causa de correção de dados. Nestas modalidades, duas sequências de reparos podem ser executadas simultaneamente, mas com configurações diferentes. Como um resultado, um reparo completo e um reparo incremental podem ser executados simultaneamente, contanto que um determinado servidor de conteúdo 110 execute um reparo de nó em um

determinado tempo.

[060] Em algumas modalidades, um procedimento de reparo antientropia para um agrupamento 140 grande pode ser executado como um reparo altamente concorrente, onde dois ou mais servidores de conteúdo 110 podem realizar concorrentemente um reparo de nó para o mesmo procedimento de reparo antientropia, referido neste documento como um reparo “deslocado” ou um reparo “concorrente”. Em tais modalidades, múltiplos servidores de conteúdo 110 separados que não compartilham quaisquer faixas executam um reparo de nó em paralelo. No ponto em que um servidor de conteúdo 110 determina se o servidor de conteúdo 110 é o próximo em sequência, o servidor de conteúdo 110 alternativamente pode determinar se prosseguir para reparo de nó não impactaria negativamente quaisquer servidores de conteúdo 110 vizinhos.

[061] Em algumas modalidades, um servidor de conteúdo 110 pode adquirir um bloqueio na tabela de estado repair_process, mas pode falhar para prosseguir para gerar uma tabela de estado repair_sequence para iniciar o próximo procedimento de reparo antientropia. Em tais modalidades, um ou mais outros servidores de conteúdo 110 no agrupamento 140 podem monitorar progresso do servidor de conteúdo 110 tentando adquirir o bloqueio. Se o servidor de conteúdo 110 tentando adquirir o bloqueio falhar para gerar uma nova tabela de estado repair_sequence dentro de um período de tempo limite configurável, tal como trinta minutos, então um outro servidor de conteúdo 110 pode iniciar o próximo procedimento de reparo antientropia e cancelar o procedimento de reparo antientropia que falhou.

[062] Em algumas modalidades, um servidor de conteúdo 110 pode determinar que o servidor de conteúdo 110 é o próximo em sequência, mas pode falhar para realizar o reparo de nó. Em tais modalidades, o servidor de conteúdo 110 pode gerar e monitorar continuamente mensagens de pulsação para a linha correspondente na tabela de estado repair_sequence para assegurar que o servidor de conteúdo 110 está progredindo no reparo de nó. Outros servidores de conteúdo 110 também podem monitorar as mensagens de pulsação. Se o servidor de conteúdo 110 tentando atualmente um reparo de nó não atualizar a mensagem de pulsação dentro de um período de tempo limite configurável, tal como trinta minutos, então um outro servidor de conteúdo 110 que é o próximo na tabela de estado repair_sequence pode cancelar todos os reparos em andamento do servidor de conteúdo 110 impedido de prosseguir. Este servidor de conteúdo 110 então marca o status do servidor de conteúdo 110 impedido de prosseguir como CANCELADO na tabela de estado repair_sequence e prossegue com o reparo de nó.

[063] Em algumas modalidades, um determinado servidor de conteúdo 110 pode dispender uma quantidade excessiva de tempo e recursos de CPU ao realizar um reparo de nó ou pode gerar uma quantidade excessiva de tráfego de rede ao realizar um reparo de nó. Para resolver esta questão, o servidor de conteúdo 110 realiza uma função de faixa de divisão adaptável para dividir adaptativamente trabalho de reparo em subfaixas de trabalho que são adaptadas para o tamanho e número de partições no servidor de conteúdo 110. As subfaixas são dimensionadas de maneira que cada subfaixa seja completada dentro de um período de tempo limite

configurável, tal como trinta minutos. Se um reparo em uma subfaixa particular exceder o período de tempo limite por uma quantidade especificada, então o reparo nessa subfaixa pode ser cancelado e reprogramado.

[064] Em algumas modalidades, a base de dados pode reiniciar enquanto um servidor de conteúdo 110 está realizando um reparo de nó. Em tais modalidades, o servidor de conteúdo 110 pode reiniciar reparo de nó assim que a base de dados for reiniciada. Por causa de cada subfaixa de um reparo ser dimensionada para ser completada dentro de um período de tempo limite configurável, tal como trinta minutos, a quantidade de trabalho perdido por causa de um reinício de base de dados é minimizada. Uma vez que a base de dados seja reiniciada, os servidores de conteúdo 110 que estavam em um estado INICIADO anteriormente restabelecem transição para um estado CANCELADO na tabela de estado repair_sequence e reiniciam reparo na mesma tabela de estado e/ou na mesma subfaixa tal como quando o restabelecimento de base de dados ocorreu. Este processo assegura que o reparo de nó é completado na sua totalidade.

[065] Em algumas modalidades, um servidor de conteúdo 110 pode ficar impedido de prosseguir ao realizar ganchos pós-reparo. De uma maneira geral, reparos não precisam ser atrasados por causa de falha ao completar ganchos pós-reparo. Portanto, esta questão pode ser resolvida com um período de tempo limite agressivo para ser aplicado para os servidores de conteúdo 110 no estado REPAIR_HOOK_RUNNING. Se a quantidade de tempo que um servidor de conteúdo 110 gasta realizando ganchos pós-reparo exceder o período de tempo limite, então o servidor de conteúdo 110 pode ser

cancelado e reiniciado.

[066] Em algumas modalidades, servidores de conteúdo 110 podem ser adicionados durante execução de um reparo. Porque novos servidores de conteúdo 110 não impactam a sequência do reparo atual, o reparo atual continua a executar com a sequência existente. O reparo subsequente então adiciona os novos servidores de conteúdo 110 à próxima tabela de estado repair_sequence.

[067] Em algumas modalidades, os servidores de conteúdo 110 podem ser terminados ou se tornam indisponíveis durante um reparo. Em tais modalidades, um ou mais servidores de conteúdo 110 podem monitorar a saúde de outros servidores de conteúdo 110 no agrupamento 140. Se uma subfaixa para um determinado servidor de conteúdo 110 estiver indisponível por um período de tempo limite configurável, tal como trinta minutos, então o status do servidor de conteúdo 110 é estabelecido para FALHADO e o reparo prossegue.

[068] A figura 3 é um diagrama de blocos de um servidor de controle 120 que pode ser implementado em combinação com a infraestrutura de rede 100 da figura 1, de acordo com várias modalidades da presente invenção. Tal como mostrado, o servidor de controle 120 inclui, sem limitação, um processador 304, um disco de sistema 306, uma interface de dispositivos de entrada/saída (I/O) 308, uma interface de rede 310, uma interconexão 312 e uma memória de sistema 314.

[069] O processador 304 é incluído para ser representativo de uma única CPU, múltiplas CPUs, uma única CPU tendo múltiplos núcleos de processamento e outros mais. O processador 304 é configurado para recuperar e executar

instruções de programação, tais como uma aplicação de controle 317, armazenadas na memória de sistema 314. De modo similar, o processador 304 é configurado para armazenar dados de aplicação (por exemplo, bibliotecas de software) e recuperar dados de aplicação da memória de sistema 314 e de uma base de dados 318 armazenada no disco de sistema 306. A interconexão 312 é configurada para facilitar transmissão de dados entre o processador 304, o disco de sistema 306, a interface de dispositivos de entrada/saída 308, a interface de rede 310 e a memória de sistema 314. A interface de dispositivos de entrada/saída 308 é configurada para transmitir dados de entrada e dados de saída entre os dispositivos de entrada/saída 316 e o processador 304 por meio da interconexão 312. O disco de sistema 306 pode incluir uma ou mais unidades de discos rígidos, dispositivos de armazenamento de estado sólido e outros mais. O disco de sistema 206 é configurado para armazenar uma base de dados 318 de informação associada com os servidores de conteúdo 110, a(s) fonte(s) de preenchimento 130 e os arquivos 218.

[070] A memória de sistema 314 inclui uma aplicação de controle 317 configurada para acessar informação armazenada na base de dados 318 e processar a informação para determinar o modo no qual arquivos específicos 218 serão reproduzidos através dos servidores de conteúdo 110 incluídos na infraestrutura de rede 100. A aplicação de controle 317 também pode ser configurada para receber e analisar características de desempenho associadas com um ou mais de os servidores de conteúdo 110 e/ou os dispositivos de pontos de extremidades 115.

[071] A figura 4 é um diagrama de blocos de um dispositivo de ponto de extremidade 115 que pode ser implementado em combinação com a infraestrutura de rede 100 da figura 1, de acordo com várias modalidades da presente invenção. Tal como mostrado, o dispositivo de ponto de extremidade 115 pode incluir, sem limitação, um processador 410, um subsistema gráfico 412, uma interface de dispositivos de entrada/saída 414, uma unidade de armazenamento de massa 416, uma interface de rede 418, uma interconexão 422 e um subsistema de memória 430.

[072] O processador 410 é incluído para ser representativo de uma única CPU, múltiplas CPUs, uma única CPU tendo múltiplos núcleos de processamento e outros mais. Em algumas modalidades, o processador 410 é configurado para recuperar e executar instruções de programação armazenadas no subsistema de memória 430. De modo similar, o processador 410 é configurado para armazenar e recuperar dados de aplicação (por exemplo, bibliotecas de software) residindo no subsistema de memória 430. A interconexão 422 é configurada para facilitar transmissão de dados, tais como instruções de programação e dados de aplicação, entre o processador 410, o subsistema gráfico 412, a interface de dispositivos de entrada/saída 414, o armazenamento de massa 416, a interface de rede 418 e o subsistema de memória 430.

[073] Em algumas modalidades, o subsistema gráfico 412 é configurado para gerar quadros de dados de vídeo e transmitir os quadros de dados de vídeo para o dispositivo de exibição 450. Em algumas modalidades, o subsistema gráfico 412 pode ser integrado em um circuito integrado, junto com o processador 410. O dispositivo de exibição 450

pode compreender quaisquer dispositivos tecnicamente exequíveis para gerar uma imagem para exibição. Por exemplo, o dispositivo de exibição 450 pode ser fabricado usando tecnologia de tela de cristal líquido (LCD), tecnologia de raios catódicos e tecnologia de exibição de diodos emissores de luz (LED). Uma interface de dispositivos de entrada/saída (I/O) 414 é configurada para receber dados de entrada dos dispositivos de entrada/saída de usuário 452 e transmitir os dados de entrada para o processador 410 por meio da interconexão 422. Por exemplo, os dispositivos de entrada/saída de usuário 452 podem compreender uma ou mais teclas, um teclado e um mouse ou outro dispositivo indicador. A interface de dispositivos de entrada/saída 414 inclui também uma unidade de saída de áudio configurada para gerar um sinal elétrico de saída de áudio. Os dispositivos de entrada/saída de usuário 452 incluem um alto-falante configurado para gerar uma saída acústica em resposta ao sinal elétrico de saída de áudio. Em modalidades alternativas, o dispositivo de exibição 450 pode incluir o alto-falante. Uma televisão é um exemplo de um dispositivo conhecido na técnica que pode exibir quadros de vídeo e gerar uma saída acústica.

[074] Uma unidade de armazenamento de massa 416, tal como uma unidade de disco rígido ou unidade de armazenamento de memória flash, é configurada para armazenar dados não voláteis. Uma interface de rede 418 é configurada para transmitir e receber pacotes de dados por meio da rede 105. Em algumas modalidades, a interface de rede 418 é configurada para se comunicar usando o padrão Ethernet bem conhecido. A interface de rede 418 é acoplada

ao processador 410 por meio da interconexão 422.

[075] Em algumas modalidades, o subsistema de memória 430 inclui instruções de programação e dados de aplicação que compreendem um sistema operacional 432, uma interface de usuário 434 e uma aplicação de reprodução 436. O sistema operacional 432 realiza funções de gerenciamento de sistema tais como gerenciar dispositivos de hardware incluindo a interface de rede 418, a unidade de armazenamento de massa 416, a interface de dispositivos de entrada/saída 414 e o subsistema gráfico 412. O sistema operacional 432 também fornece modelos de gerenciamento de processo e memória para a interface de usuário 434 e para a aplicação de reprodução 436. A interface de usuário 434, tal como uma metáfora de janela e objeto, fornece um mecanismo para interação de usuário com o dispositivo de ponto de extremidade 108. Pessoas versadas na técnica reconhecerão os vários sistemas operacionais e interfaces de usuário que são bem conhecidos na técnica e adequados para incorporação ao dispositivo de ponto de extremidade 108.

[076] Em algumas modalidades, a aplicação de reprodução 436 é configurada para solicitar e receber conteúdo do servidor de conteúdo 105 por meio da interface de rede 418. Adicionalmente, a aplicação de reprodução 436 é configurada para interpretar o conteúdo e apresentar o conteúdo por meio do dispositivo de exibição 450 e/ou dos dispositivos de entrada/saída de usuário 452.

[077] As figuras 5A-5B mostram um fluxograma de etapas de método para realizar procedimentos de reparos antientropia distribuídos através de uma pluralidade dos servidores de conteúdo 110 em uma rede de base de dados

distribuída, de acordo com várias modalidades da presente invenção. Embora as etapas de método sejam descritas em combinação com os sistemas das figuras 1-4, pessoas de conhecimento comum na técnica entenderão que qualquer sistema configurado para realizar as etapas de método, em qualquer ordem, está dentro do escopo da presente invenção.

[078] Tal como mostrado, um método 500 começa na etapa 502, onde uma aplicação de reparo 219 executando em um servidor de conteúdo 110 determina se existe atualmente um procedimento de reparo antientropia em andamento. Mais especificamente, a aplicação de reparo 219 lê a tabela de estado repair_process ampla de agrupamento para determinar se o procedimento de reparo antientropia mais recente mostra um status de completo ou em andamento.

[079] Se um procedimento de reparo antientropia estiver atualmente em andamento, então o método 500 prossegue para a etapa 504, onde a aplicação de reparo 219 determina se o servidor de conteúdo 110 é o próximo em sequência para realizar um reparo, tal como indicado pela tabela de estado repair_sequence. Se todos os servidores de conteúdo 110 anteriores mostrados na tabela de estado repair_sequence tiverem um status de FALHADO ou TERMINADO, então o servidor de conteúdo 110 é o próximo em sequência. Em tais casos, o método 500 prossegue para a etapa 506, onde a aplicação de reparo 219 realiza um reparo em partições armazenadas no servidor de conteúdo 110. Mais especificamente, a aplicação de reparo 219 realiza um reparo para cada partição armazenada no servidor de conteúdo 110. Se uma partição tiver sido dividida em múltiplas subfaixas, então a aplicação de reparo 219 realiza o reparo para uma subfaixa

de cada vez. Após a aplicação de reparo 219 completar o reparo para todas as partições e subfaixas armazenadas no servidor de conteúdo 110, o método 500 então termina.

[080] Se, entretanto, na etapa 504, o servidor de conteúdo 110 não for o próximo em sequência, então o método prossegue para a etapa 508, onde a aplicação de reparo 219 determina se o servidor de conteúdo 110 atual e todos os servidores de conteúdo 110 vizinhos do servidor de conteúdo 110 atual completaram os respectivos reparos. Se o servidor de conteúdo 110 atual ou pelo menos um servidor de conteúdo 110 vizinho do servidor de conteúdo 110 atual não tiver completado reparos, então o método termina. Se, entretanto, o servidor de conteúdo 110 atual e todos os servidores de conteúdo 110 vizinhos completaram os respectivos reparos, então o método 500 prossegue para a etapa 510, onde a aplicação de reparo 219 executa ganchos pós-reparo. Tal como descrito adicionalmente neste documento, tais ganchos pós-reparo realizam operações de limpeza e/ou de compactação assim como outras tarefas de manutenção, tal como transmitir uma notificação para uma aplicação de monitoramento de que o servidor de conteúdo 110 atual completou reparo. O método 500 então termina.

[081] Retornando para a etapa 502, se um procedimento de reparo antientropia não estiver atualmente em andamento, então o método 500 prossegue para a etapa 512, onde a aplicação de reparo 219 determina se um novo procedimento de reparo antientropia está pronto para iniciar. Em particular, a aplicação de reparo 219 determina que não existem restrições adicionais com relação a iniciar um novo procedimento de reparo antientropia, tal como um intervalo

mínimo de tempo entre dois procedimentos de reparos antientropia sucessivos ou uma restrição de hora do dia para executar procedimentos de reparos antientropia somente fora de horário de pico. Se um novo procedimento de reparo antientropia não estiver pronto para iniciar, então o método 500 termina. Se, entretanto, um novo procedimento de reparo antientropia estiver pronto para iniciar, então o método prossegue para a etapa 514, onde a aplicação de reparo 219 tenta adquirir um bloqueio na tabela de estado repair_process. Se a aplicação de reparo 219 não for bem sucedida ao adquirir o bloqueio na tabela de estado repair_process, então o método 500 termina. Se, entretanto, a aplicação de reparo 219 for bem sucedida ao adquirir o bloqueio na tabela de estado repair_process, então o método 500 prossegue para a etapa 518, onde a aplicação de reparo 219 gera a tabela de estado repair_sequence para o próximo procedimento de reparo antientropia. A aplicação de reparo 219 armazena a tabela de estado repair_sequence em uma operação em lote. O método 500 então termina.

[082] Além disso, tal como descrito adicionalmente neste documento, cada servidor de conteúdo 110 em um agrupamento 140 realiza as etapas do método 500 periodicamente, tal como uma vez a cada dois minutos.

[083] Em resumo, um procedimento de reparo antientropia executa através dos nós em rede de base de dados distribuída, onde nenhum nó é designado antecipadamente como um coordenador de reparo antientropia. Cada nó realiza periodicamente uma técnica, onde o nó primeiro determina se um procedimento de reparo antientropia está em andamento. Se um procedimento de reparo antientropia estiver em

andamento, então o nó determina se esse nó é o próximo em sequência para realizar o reparo. Se o nó for o próximo em sequência, então o nó repara as réplicas e tabelas de estados associadas e sai. Se o nó não for o próximo em sequência, então o nó determina se o nó tem permissão para realizar procedimentos pós-reparo relacionados com o nó. Se o nó tiver permissão, então o nó realiza os procedimentos pós-reparo e sai. De outro modo, o nó sai sem realizar os procedimentos pós-reparo.

[084] Se, por outro lado, o nó determinar que um procedimento de reparo antientropia não está em andamento, então o nó determina se um novo procedimento de reparo antientropia deve ser iniciado. Se um novo reparo antientropia não puder ser iniciado, então o nó sai. De outro modo, o nó tenta iniciar um novo procedimento de reparo antientropia e, se bem sucedido, gera uma nova sequência que define a ordem de sequência em que os nós realizam reparos de nós relacionados com o novo procedimento de reparo antientropia. O nó então sai.

[085] Pelo menos uma vantagem técnica das técnicas divulgadas em relação à técnica anterior é que procedimentos de reparos antientropia escalam automaticamente para o número de nós e para o tamanho variado das réplicas em uma rede de base de dados distribuída. Como um resultado, o procedimento de reparo antientropia se adapta a cada subconjunto de dados armazenados em um nó particular a fim de reduzir a quantidade de acesso a disco, recursos de CPU e largura de banda de rede consumida, em relação às técnicas convencionais. Além disso, usuários são aliviados de

projetar e manter manualmente procedimentos de reparos antientropia à medida que o número de nós e o tamanho de réplicas mudam ao longo do tempo. Uma outra vantagem técnica das técnicas divulgadas em relação à técnica anterior é que o procedimento de reparo antientropia é distribuído através dos nós. Adicionalmente, uma vez que os nós falhos reiniciam execução, o procedimento de reparo antientropia pode reiniciar relativamente na mesma condição de quando a falha ocorreu. Consequentemente, menos ou nenhum progresso de reparo é perdido quando um ou mais outros nós falham durante um reparo, em relação às abordagens de técnica anterior. Estas vantagens técnicas representam um ou mais melhoramentos tecnológicos em relação às abordagens de técnica anterior.

[086] 1. Em algumas modalidades, um método implementado por computador compreende: determinar, por um primeiro nó incluído em uma pluralidade de nós e antes de todos os outros nós incluídos na pluralidade de nós, que um primeiro procedimento de reparo antientropia terminou; determinar, pelo primeiro nó, que um segundo procedimento de reparo antientropia está pronto para começar; gerar um escalonamento para executar uma ou mais operações associadas com o segundo procedimento de reparo antientropia; e gravar o escalonamento em uma estrutura de dados de escalonamentos de reparos compartilhados para iniciar o segundo procedimento de reparo antientropia através de múltiplos nós incluídos na pluralidade de nós.

[087] 2. O método implementado por computador de acordo com a cláusula 1, em que determinar que o segundo procedimento de reparo antientropia está pronto para

começar compreende determinar que um tempo corrente está dentro de uma faixa especificada de tempo designado para operações de reparos.

[088] 3. O método implementado por computador de acordo com a cláusula 1 ou cláusula 2, compreendendo adicionalmente: determinar, por um segundo nó incluído na pluralidade de nós, que um terceiro procedimento de reparo antientropia está pronto para começar; gerar um escalonamento para executar uma ou mais operações associadas com o terceiro procedimento de reparo antientropia; e gravar o escalonamento em uma segunda estrutura de dados de escalonamentos de reparos compartilhados para iniciar o terceiro procedimento de reparo antientropia através de múltiplos nós incluídos na pluralidade de nós.

[089] 4. O método implementado por computador de acordo com qualquer uma das cláusulas 1-3, em que: o segundo procedimento de reparo antientropia compreende um procedimento de reparo antientropia completo, o terceiro procedimento de reparo antientropia compreende um procedimento de reparo antientropia incremental, e determinar que o terceiro procedimento de reparo antientropia está pronto para começar compreende determinar que um quarto procedimento de reparo antientropia que compreende um reparo incremental terminou.

[090] 5. O método implementado por computador de acordo com qualquer uma das cláusulas 1-4, compreendendo adicionalmente: determinar, por um segundo nó incluído na pluralidade de nós, que o segundo procedimento de reparo antientropia está em andamento; determinar que o segundo nó

é o próximo em sequência a ser reparado; e reparar pelo menos uma partição de dados inconsistente que reside no segundo nó.

[091] 6. O método implementado por computador de acordo com qualquer uma das cláusulas 1-5, compreendendo adicionalmente: determinar, por um segundo nó incluído na pluralidade de nós, que o segundo procedimento de reparo antientropia está em andamento; determinar que o segundo nó é independente de todos os outros nós incluídos na pluralidade de nós que estão realizando correntemente um reparo associado com o segundo procedimento de reparo antientropia; e reparar pelo menos uma partição de dados inconsistente que reside no segundo nó.

[092] 7. O método implementado por computador de acordo com qualquer uma das cláusulas 1-6, compreendendo adicionalmente: determinar, por um segundo nó incluído na pluralidade de nós, que o segundo procedimento de reparo antientropia está em andamento; determinar que o segundo nó realizou um reparo associado com o segundo procedimento de reparo antientropia; determinar que todos os outros nós incluídos na pluralidade de nós que são interdependentes com o segundo nó realizaram um reparo associado com o segundo procedimento de reparo antientropia; e realizar, pelo segundo nó, um procedimento de pós-reparo associado com o segundo procedimento de reparo antientropia.

[093] 8. O método implementado por computador de acordo com qualquer uma das cláusulas 1-7, em que realizar o procedimento de pós-reparo compreende eliminar uma partição que não é mais necessária após o segundo procedimento de reparo antientropia estar completo.

[094] 9. O método implementado por computador de acordo com qualquer uma das cláusulas 1-8, em que realizar o procedimento de pós-reparo compreende realizar uma operação de compactação em uma ou mais partições associadas com o segundo nó para reduzir latência ao acessar a uma ou mais partições.

[095] 10. O método implementado por computador de acordo com qualquer uma das cláusulas 1-9, em que realizar o procedimento de pós-reparo compreende transmitir uma mensagem para uma aplicação de monitoramento indicando que o segundo nó realizou o reparo associado com o segundo procedimento de reparo antientropia.

[096] 11. O método implementado por computador de acordo com qualquer uma das cláusulas 1-10, compreendendo adicionalmente: determinar que um número de partições associadas com o segundo procedimento de reparo antientropia excede um nível limiar; e dividir trabalho associado com o segundo procedimento de reparo antientropia em uma pluralidade de subfaixas.

[097] 12. O método implementado por computador de acordo com qualquer uma das cláusulas 1-11, compreendendo adicionalmente: determinar que um tamanho de uma ou mais partições associadas com o segundo procedimento de reparo antientropia excede um nível limiar; e dividir trabalho associado com o segundo procedimento de reparo antientropia em uma pluralidade de subfaixas.

[098] 13. O método implementado por computador de acordo com qualquer uma das cláusulas 1-12, compreendendo adicionalmente: determinar que um tempo de conclusão para o segundo procedimento de reparo antientropia excede um nível

limiar; e dividir trabalho associado com o segundo procedimento de reparo antientropia em uma pluralidade de subfaixas com base no tempo de conclusão.

[099] 14. Em algumas modalidades, um ou mais meios de armazenamento não transitório legíveis por computador incluem instruções que, quando executadas por um ou mais processadores, induzem o um ou mais processadores para realizar as etapas de: determinar que um primeiro procedimento de reparo antientropia terminou; determinar, por um primeiro nó incluído em uma pluralidade de nós e antes de todos os outros nós incluídos na pluralidade de nós, que um segundo procedimento de reparo antientropia está pronto para começar; gerar um escalonamento para executar operações associadas com o segundo procedimento de reparo antientropia; e gravar o escalonamento em uma estrutura de dados de escalonamentos de reparos compartilhados para iniciar o segundo procedimento de reparo antientropia através de múltiplos nós incluídos na pluralidade de nós.

[0100] 15. O um ou mais meios de armazenamento não transitório legíveis por computador de acordo com a cláusula 14, em que determinar que o segundo procedimento de reparo antientropia está pronto para começar compreende determinar que um tempo corrente está dentro de uma faixa especificada de tempo designado para operações de reparos.

[0101] 16. O um ou mais meios de armazenamento não transitório legíveis por computador de acordo com a cláusula 14 ou cláusula 15, em que cada nó incluído na pluralidade de nós realiza sequencialmente um reparo associado com o segundo procedimento de reparo

antientropia.

[0102] 17. O um ou mais meios de armazenamento não transitório legíveis por computador de acordo com qualquer uma das cláusulas 14-16, em que: cada nó incluído em um primeiro subconjunto de nós incluídos na pluralidade de nós realiza um reparo associado com o segundo procedimento de reparo antientropia em paralelo uns com os outros; cada nó incluído em um segundo subconjunto de nós incluídos na pluralidade de nós realiza um reparo associado com o segundo procedimento de reparo antientropia em paralelo uns com os outros; e os nós incluídos no primeiro subconjunto de nós realizam o reparo associado com o segundo procedimento de reparo antientropia sequencialmente aos nós incluídos no segundo subconjunto de nós.

[0103] 18. O um ou mais meios de armazenamento não transitório legíveis por computador de acordo com qualquer uma das cláusulas 14-17, em que cada nó incluído na pluralidade de nós realiza um reparo associado com o segundo procedimento de reparo antientropia em paralelo uns com os outros.

[0104] 19. Em algumas modalidades, um dispositivo de computação, compreende: uma memória que inclui instruções; e um processador que é acoplado à memória e, ao executar as instruções, é configurado para: determinar que um primeiro procedimento de reparo antientropia terminou; determinar, por um primeiro nó incluído em uma pluralidade de nós e antes de todos os outros nós incluídos na pluralidade de nós, que um segundo procedimento de reparo antientropia está pronto para começar; gerar um escalonamento de reparo associado com o segundo

procedimento de reparo antientropia; e gravar o escalonamento de reparo em um armazenamento de dados para iniciar o segundo procedimento de reparo antientropia através de múltiplos nós incluídos na pluralidade de nós.

[0105] 20. O dispositivo de computação de acordo com a cláusula 19, em que o processador, ao executar as instruções, é configurado adicionalmente para designar, em um segundo nó incluído na pluralidade de nós, um número de núcleos de processamento para realizar um reparo associado com o segundo procedimento de reparo antientropia, em que o número de núcleos de processamento é no máximo uma metade de um número de núcleos de processador disponíveis dentro do segundo nó.

[0106] Qualquer uma e todas as combinações de qualquer um dos elementos de reivindicações relatados em qualquer uma das reivindicações e/ou quaisquer elementos descritos neste pedido, em qualquer modo, estão incluídas no escopo considerado da presente invenção e proteção.

[0107] As descrições das várias modalidades foram apresentadas para o propósito de ilustração, mas não pretendem ser exaustivas ou limitadas às modalidades divulgadas. Muitas modificações e variações estarão aparentes para as pessoas de conhecimento comum na técnica sem divergir do escopo e espírito das modalidades descritas.

[0108] Aspectos das presentes modalidades podem ser incorporados como um sistema, método ou produto de programa de computador. Portanto, aspectos da presente divulgação podem ter a forma de uma modalidade totalmente de hardware, uma modalidade totalmente de software (incluindo firmware,

software residente, microcódigo, etc.) ou de uma modalidade combinando aspectos de software e de hardware que de uma maneira geral todos podem ser referidos neste documento como um “módulo” ou “sistema”. Além disso, aspectos da presente divulgação podem ter a forma de um produto de programa de computador incorporado em um ou mais meios legíveis por computador tendo código de programa legível por computador incorporado ao mesmo.

[0109] Qualquer combinação de um ou mais meios legíveis por computador pode ser utilizada. O meio legível por computador pode ser um meio de sinal legível por computador ou um meio de armazenamento legível por computador. Um meio de armazenamento legível por computador pode ser, por exemplo, mas não limitado a isto, um sistema, aparelho ou dispositivo eletrônico, magnético, ótico, eletromagnético, infravermelho ou semicondutor, ou qualquer combinação adequada dos indicados acima. Mais exemplos específicos (uma lista não exaustiva) do meio de armazenamento legível por computador incluiriam o seguinte: uma conexão elétrica tendo um ou mais fios, um disquete de computador portátil, um disco rígido, uma memória de acesso aleatório (RAM), uma memória somente de leitura (ROM), uma memória somente de leitura programável e apagável (EPROM ou memória Flash), uma fibra ótica, uma memória somente de leitura de disco compacto portátil (CD-ROM), um dispositivo de armazenamento ótico, um dispositivo de armazenamento magnético, ou qualquer combinação adequada dos indicados acima. No contexto deste documento, um meio de armazenamento legível por computador pode ser qualquer meio tangível que pode conter ou armazenar um programa para uso

por ou em conexão com um sistema, aparelho ou dispositivo de execução de instruções.

[0110] Aspectos da presente divulgação estão descritos acima com referência para ilustrações de fluxogramas e/ou diagramas de blocos de métodos, aparelho (sistemas) e produtos de programas de computador de acordo com modalidades da divulgação. Será entendido que cada bloco das ilustrações de fluxogramas e/ou diagramas de blocos, e combinações de blocos nas ilustrações de fluxogramas e/ou diagramas de blocos, pode ser implementado por meio de instruções de programa de computador. Estas instruções de programa de computador podem ser fornecidas para um processador de um computador de uso geral, computador de uso especial ou de outro aparelho de processamento de dados programável para produzir uma máquina, de tal maneira que as instruções, as quais executam por meio do processador do computador ou de outro aparelho de processamento de dados programável, capacitam a implementação das funções/procedimentos especificados no bloco ou blocos de fluxograma e/ou diagrama de blocos. Tais processadores, sem limitação, podem ser processadores de uso geral, processadores de uso especial, processadores de aplicação específica ou programáveis em campo.

[0111] O fluxograma e diagramas de blocos nas figuras ilustram a arquitetura, funcionalidade e operação de implementações possíveis de sistemas, métodos e produtos de programas de computador de acordo com várias modalidades da presente divulgação. Neste aspecto, cada bloco no fluxograma ou nos diagramas de blocos pode representar um módulo, segmento ou parte de código, o qual compreende uma

ou mais instruções executáveis para implementar a(s) função(s) lógica(s) especificada(s). Também deve ser notado que, em algumas implementações alternativas, as funções notadas no bloco podem ocorrer fora da ordem notada nas figuras. Por exemplo, dois blocos mostrados em sucessão, de fato, podem ser executados substancialmente de forma concorrente, ou os blocos algumas vezes podem ser executados na ordem inversa, dependendo da funcionalidade envolvida. Também deve ser notado que cada bloco dos diagramas de blocos e/ou da ilustração de fluxograma, e combinações de blocos nos diagramas de blocos e/ou na ilustração de fluxograma, pode ser implementado por meio de sistemas baseados em hardware de uso especial que realizam as funções ou procedimentos especificados, ou de combinações de hardware de uso especial e instruções de computador.

[0112] Embora o exposto anteriormente esteja direcionado para modalidades da presente divulgação, outras e modalidades adicionais da divulgação podem ser imaginadas sem divergir do escopo básico da mesma, e o escopo da mesma é determinado pelas reivindicações que se seguem.

REIVINDICAÇÕES EMENDADAS

1. Método implementado por computador, caracterizado pelo fato de que compreende:

determinar, por um primeiro nó incluído em uma pluralidade de nós e antes de todos os outros nós incluídos na pluralidade de nós, que um primeiro procedimento de reparo antientropia terminou;

determinar, pelo primeiro nó, que um segundo procedimento de reparo antientropia está pronto para começar;

gerar um escalonamento para executar uma ou mais operações associadas com o segundo procedimento de reparo antientropia; e

gravar o escalonamento em uma estrutura de dados de escalonamentos de reparos compartilhados para iniciar o segundo procedimento de reparo antientropia através de múltiplos nós incluídos na pluralidade de nós.

2. Método implementado por computador, de acordo com a reivindicação 1, caracterizado pelo fato de que determinar que o segundo procedimento de reparo antientropia está pronto para começar compreende determinar que um tempo corrente está dentro de uma faixa especificada de tempo designado para operações de reparos.

3. Método implementado por computador, de acordo com a reivindicação 1, caracterizado pelo fato de que compreende adicionalmente:

determinar, por um segundo nó incluído na pluralidade de nós, que um terceiro procedimento de reparo antientropia está pronto para começar;

gerar um escalonamento para executar uma ou mais

operações associadas com o terceiro procedimento de reparo antientropia; e

gravar o escalonamento em uma segunda estrutura de dados de escalonamentos de reparos compartilhados para iniciar o terceiro procedimento de reparo antientropia através de múltiplos nós incluídos na pluralidade de nós.

4. Método implementado por computador, de acordo com a reivindicação 3, caracterizado pelo fato de que:

o segundo procedimento de reparo antientropia compreende um procedimento de reparo antientropia completo,

o terceiro procedimento de reparo antientropia compreende um procedimento de reparo antientropia incremental, e

determinar que o terceiro procedimento de reparo antientropia está pronto para começar compreende determinar que um quarto procedimento de reparo antientropia que compreende um reparo incremental terminou.

5. Método implementado por computador, de acordo com a reivindicação 1, caracterizado pelo fato de que compreende adicionalmente:

determinar, por um segundo nó incluído na pluralidade de nós, que o segundo procedimento de reparo antientropia está em andamento;

determinar que o segundo nó é o próximo em sequência a ser reparado; e

reparar pelo menos uma partição de dados inconsistente que reside no segundo nó.

6. Método implementado por computador, de acordo com a reivindicação 1, caracterizado pelo fato de que compreende adicionalmente:

determinar, por um segundo nó incluído na pluralidade de nós, que o segundo procedimento de reparo antientropia está em andamento;

determinar que o segundo nó é independente de todos os outros nós incluídos na pluralidade de nós que estão realizando correntemente um reparo associado com o segundo procedimento de reparo antientropia; e

reparar pelo menos uma partição de dados inconsistente que reside no segundo nó.

7. Método implementado por computador, de acordo com a reivindicação 1, caracterizado pelo fato de que compreende adicionalmente:

determinar, por um segundo nó incluído na pluralidade de nós, que o segundo procedimento de reparo antientropia está em andamento;

determinar que o segundo nó realizou um reparo associado com o segundo procedimento de reparo antientropia;

determinar que todos os outros nós incluídos na pluralidade de nós que são interdependentes do segundo nó realizaram um reparo associado com o segundo procedimento de reparo antientropia; e

realizar, pelo segundo nó, um procedimento de pós-reparo associado com o segundo procedimento de reparo antientropia.

8. Método implementado por computador, de acordo com a reivindicação 7, caracterizado pelo fato de que realizar o procedimento de pós-reparo compreende eliminar uma partição que não é mais necessária após o segundo procedimento de reparo antientropia estar completo.

9. Método implementado por computador, de acordo com a reivindicação 7, caracterizado pelo fato de que realizar

o procedimento de pós-reparo compreende realizar uma operação de compactação em uma ou mais partições associadas com o segundo nó para reduzir latência ao acessar a uma ou mais partições.

10. Método implementado por computador, de acordo com a reivindicação 7, caracterizado pelo fato de que realizar o procedimento de pós-reparo compreende transmitir uma mensagem para uma aplicação de monitoramento indicando que o segundo nó realizou o reparo associado com o segundo procedimento de reparo antientropia.

11. Método implementado por computador, de acordo com a reivindicação 1, caracterizado pelo fato de que compreende adicionalmente:

determinar que um número de partições associadas com o segundo procedimento de reparo antientropia excede um nível limiar; e

dividir trabalho associado com o segundo procedimento de reparo antientropia em uma pluralidade de subfaixas.

12. Método implementado por computador, de acordo com a reivindicação 1, caracterizado pelo fato de que compreende adicionalmente:

determinar que um tamanho de uma ou mais partições associadas com o segundo procedimento de reparo antientropia excede um nível limiar; e

dividir trabalho associado com o segundo procedimento de reparo antientropia em uma pluralidade de subfaixas.

13. Método implementado por computador, de acordo com a reivindicação 1, caracterizado pelo fato de que compreende adicionalmente:

determinar que um tempo de conclusão para o segundo

procedimento de reparo antientropia excede um nível limiar; e

dividir trabalho associado com o segundo procedimento de reparo antientropia em uma pluralidade de subfaixas com base no tempo de conclusão.

14. Um ou mais meios de armazenamento não transitório legíveis por computador, caracterizados pelo fato de que incluem instruções que, quando executadas por um ou mais processadores, induzem o um ou mais processadores para realizar as etapas de:

determinar que um primeiro procedimento de reparo antientropia terminou;

determinar, por um primeiro nó incluído em uma pluralidade de nós e antes de todos os outros nós incluídos na pluralidade de nós, que um segundo procedimento de reparo antientropia está pronto para começar;

gerar um escalonamento para executar operações associadas com o segundo procedimento de reparo antientropia; e

gravar o escalonamento em uma estrutura de dados de escalonamentos de reparos compartilhados para iniciar o segundo procedimento de reparo antientropia através de múltiplos nós incluídos na pluralidade de nós.

15. Dispositivo de computação, caracterizado pelo fato de que compreende:

uma memória que inclui instruções; e

um processador que é acoplado à memória e, ao executar as instruções, é configurado para:

determinar que um primeiro procedimento de reparo antientropia terminou;

determinar, por um primeiro nó incluído em uma pluralidade de nós e antes de todos os outros nós incluídos na pluralidade de nós, que um segundo procedimento de reparo antientropia está pronto para começar;

gerar um escalonamento de reparo associado com o segundo procedimento de reparo antientropia; e

gravar o escalonamento de reparo em um armazenamento de dados para iniciar o segundo procedimento de reparo antientropia através de múltiplos nós incluídos na pluralidade de nós.

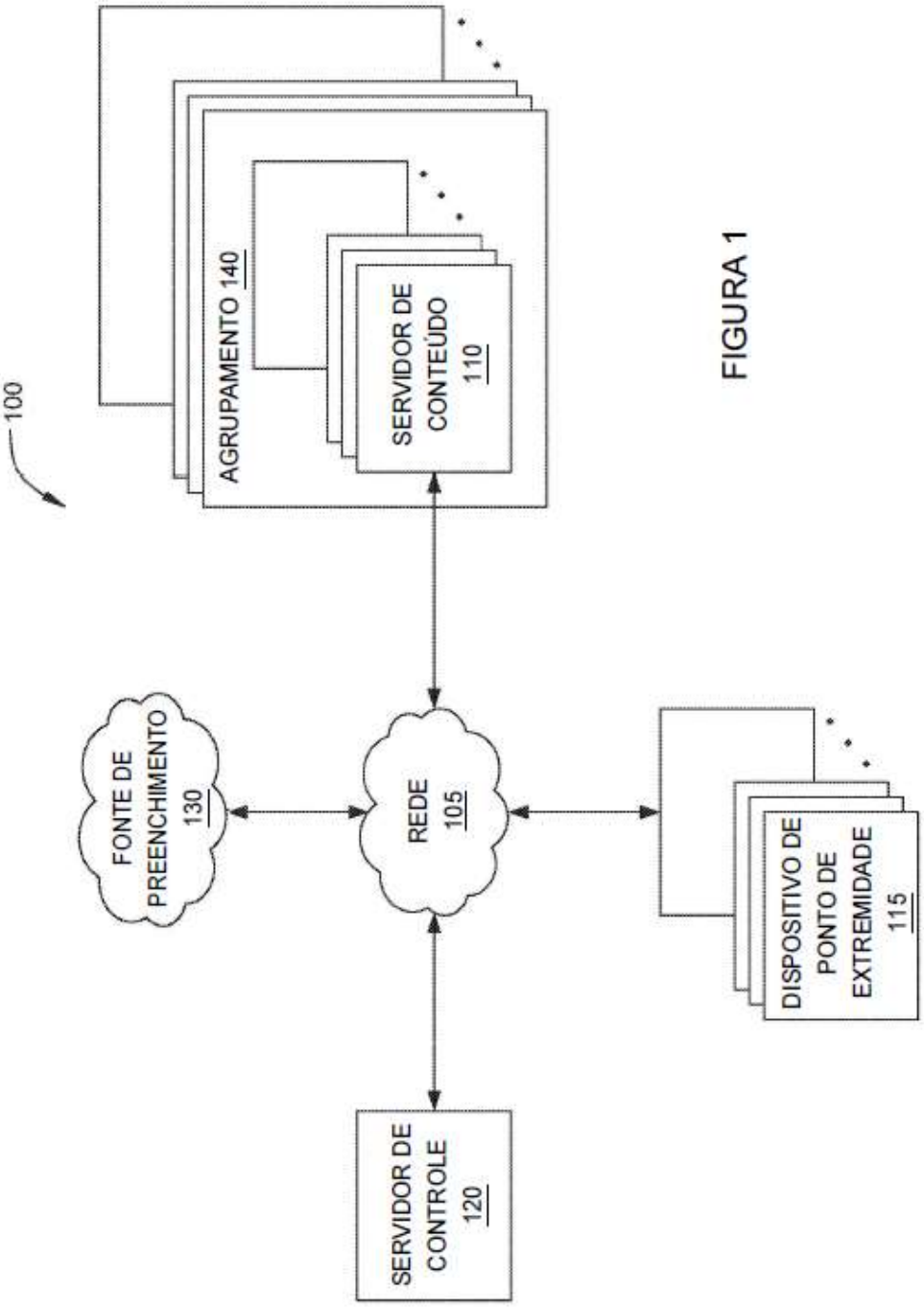

100
AGRUPAMENTO 140
SERVIDOR DE CONTEÚDO 110
FONTE DE PREENCHIMENTO 130
REDE 105
DISPOSITIVO DE PONTO DE EXTREMIDADE 115
SERVIDOR DE CONTROLE 120

FIGURA 1

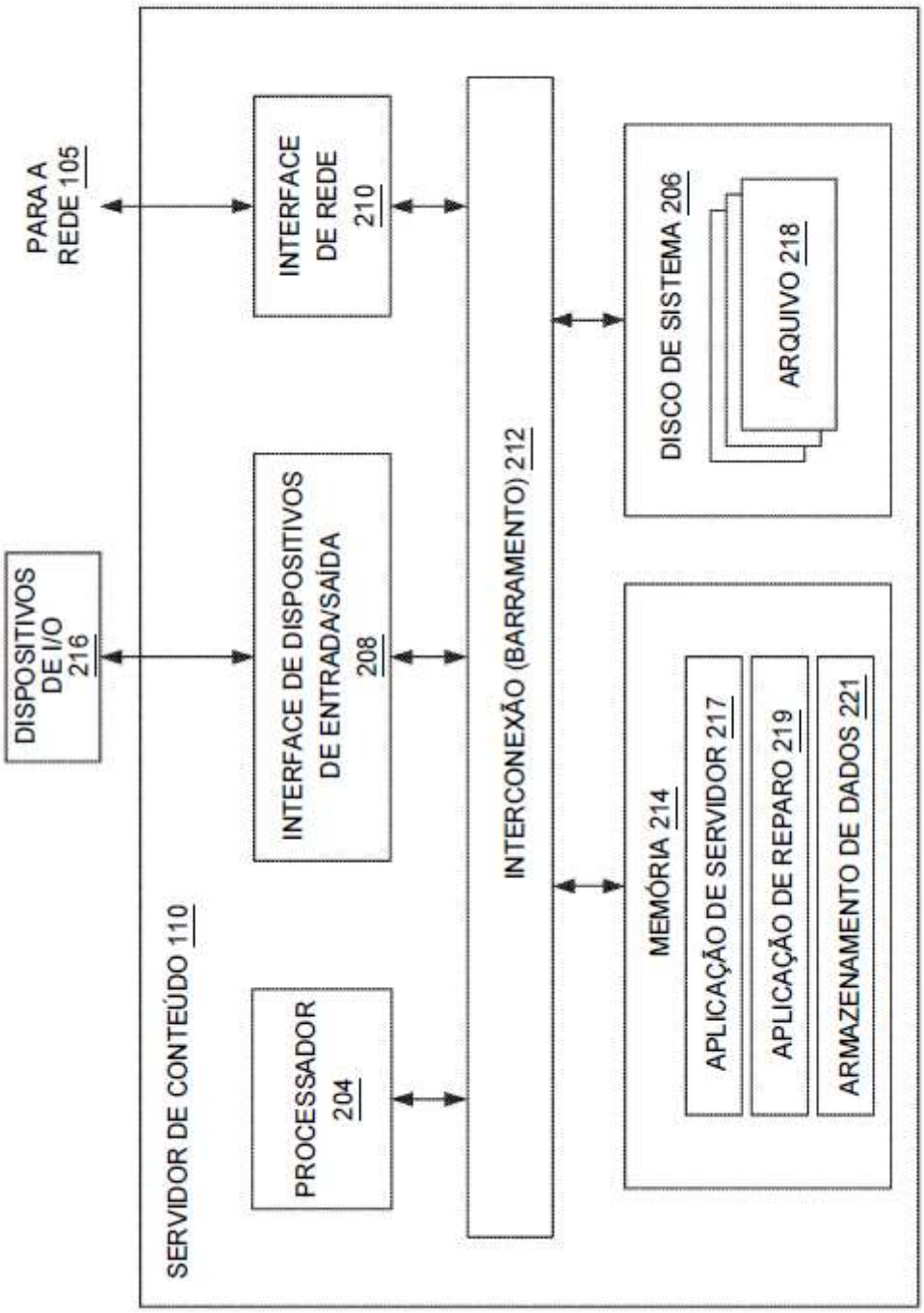

PARA A REDE 105
INTERFACE DE REDE 210
DISPOSITIVOS DE I/O 216
INTERFACE DE DISPOSITIVOS DE ENTRADA/SAÍDA 208
INTERCONEXÃO (BARRAMENTO) 212
DISCO DE SISTEMA 206
ARQUIVO 218
SERVIDOR DE CONTEÚDO 110
PROCESSADOR 204
MEMÓRIA 214
APLICAÇÃO DE SERVIDOR 217
APLICAÇÃO DE REPARO 219
ARMAZENAMENTO DE DADOS 221

FIGURA 2

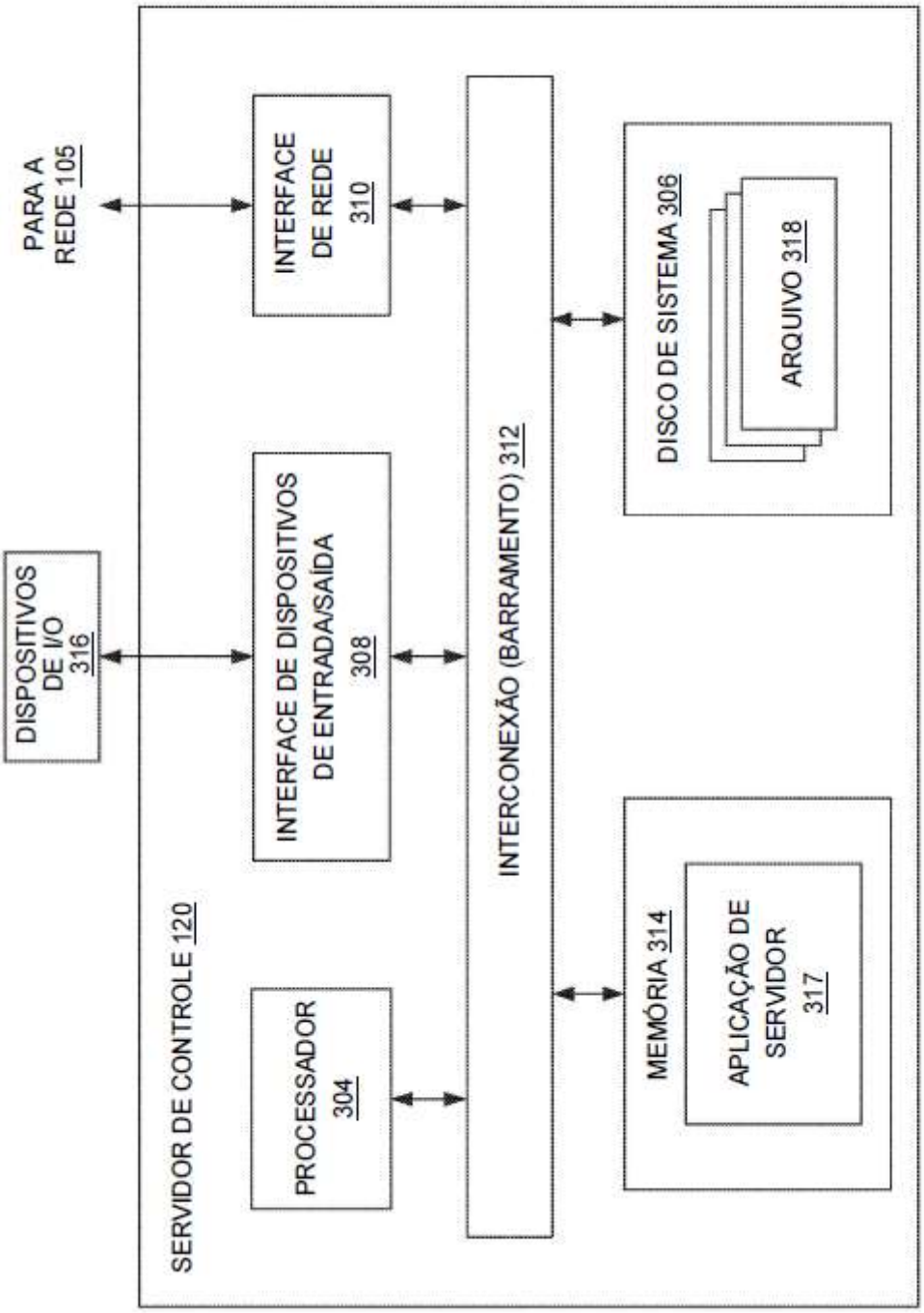

PARA A REDE 105
INTERFACE DE REDE 310
DISPOSITIVOS DE I/O 316
INTERFACE DE DISPOSITIVOS DE ENTRADA/SAÍDA 308
INTERCONEXÃO (BARRAMENTO) 312
DISCO DE SISTEMA 306
ARQUIVO 318
SERVIDOR DE CONTROLE 120
PROCESSADOR 304
MEMÓRIA 314
APLICAÇÃO DE SERVIDOR 317

FIGURA 3

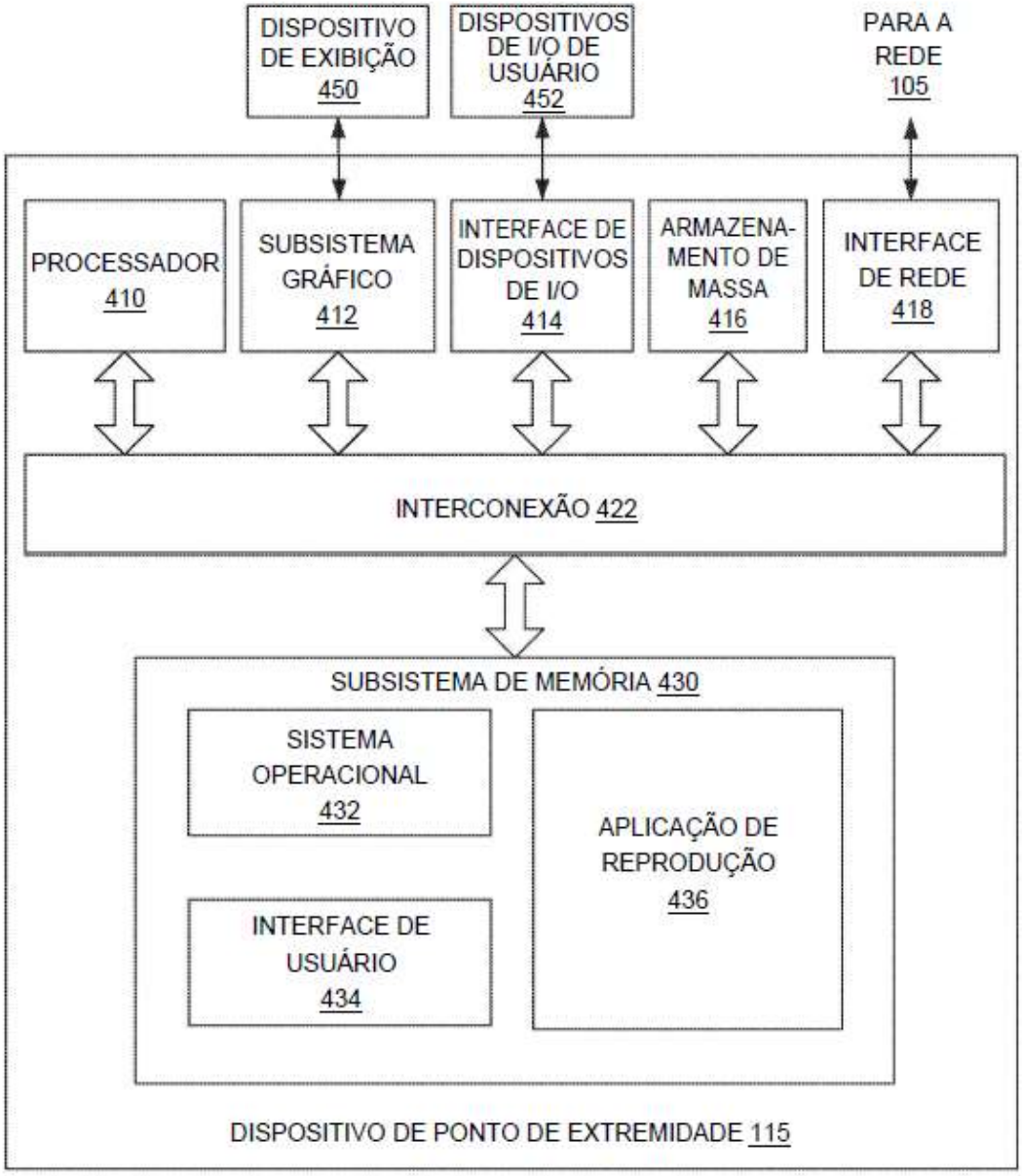

DISPOSITIVO DE EXIBIÇÃO 450
DISPOSITIVOS DE I/O DE USUÁRIO 452
PARA A REDE 105
PROCESSADOR 410
SUBSISTEMA GRÁFICO 412
INTERFACE DE DISPOSITIVOS DE I/O 414
ARMAZENA-MENTO DE MASSA 416
INTERFACE DE REDE 418
INTERCONEXÃO 422
SUBSISTEMA DE MEMÓRIA 430
SISTEMA OPERACIONAL 432
INTERFACE DE USUÁRIO 434
APLICAÇÃO DE REPRODUÇÃO 436
DISPOSITIVO DE PONTO DE EXTREMIDADE 115

FIGURA 4

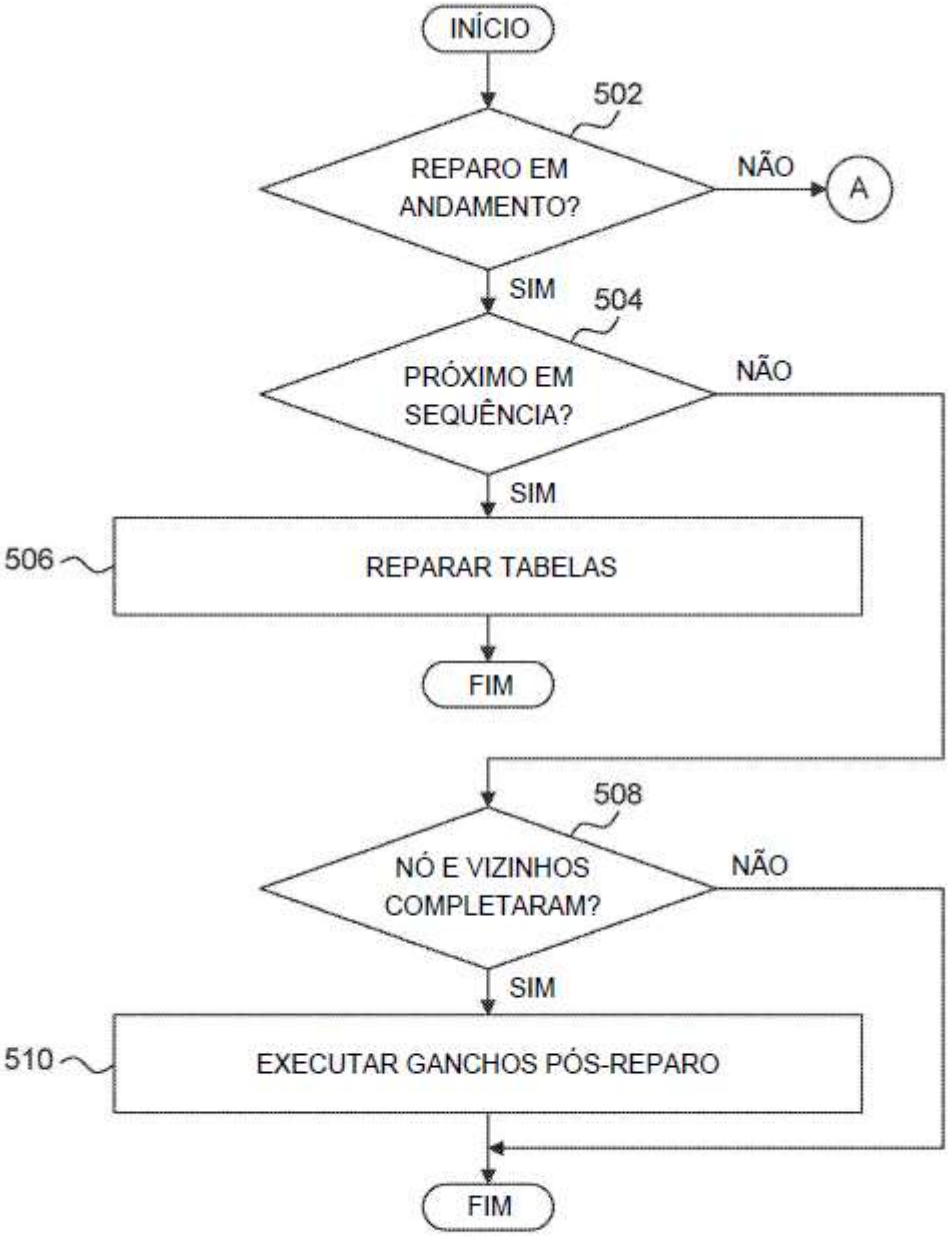

INÍCIO
502
REPARO EM ANDAMENTO?
NÃO
A
SIM
504
PRÓXIMO EM SEQUÊNCIA?
NÃO
SIM
506
REPARAR TABELAS
FIM
508
NÓ E VIZINHOS COMPLETARAM?
NÃO
SIM
510
EXECUTAR GANCHOS PÓS-REPARO
FIM

FIGURA 5A

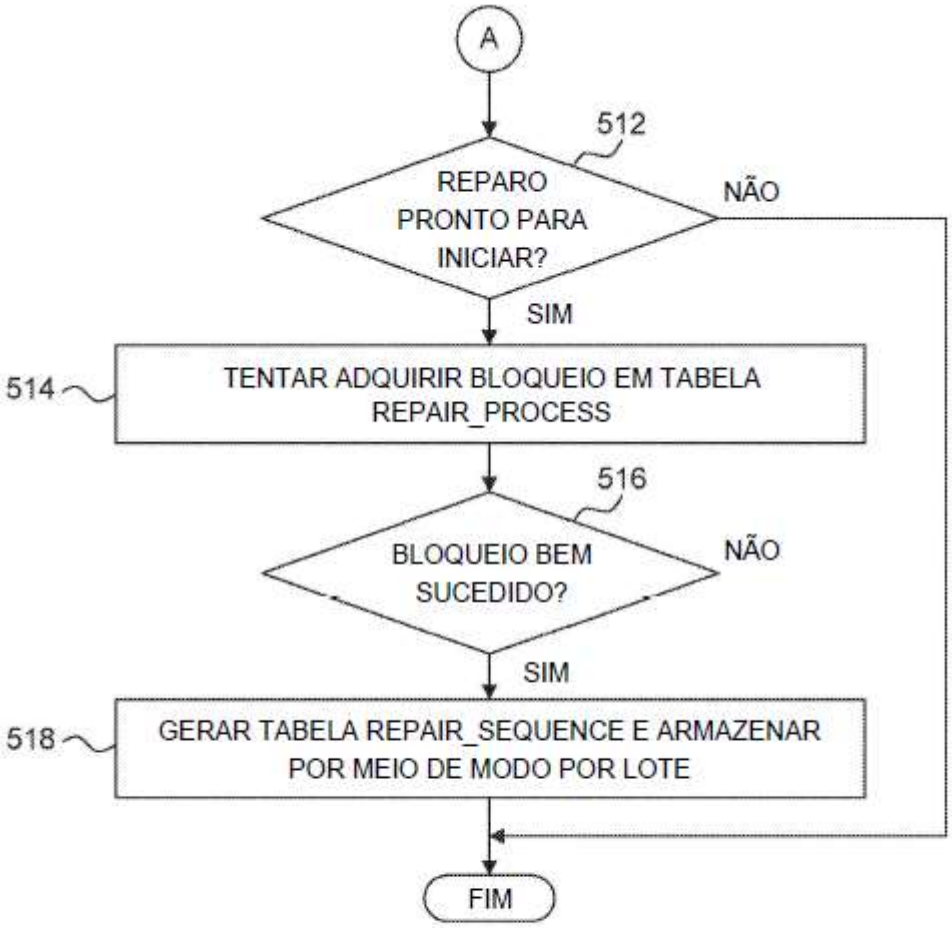

A
512
REPARO PRONTO PARA INICIAR?
NÃO
SIM
514
TENTAR ADQUIRIR BLOQUEIO EM TABELA REPAIR_PROCESS
516
BLOQUEIO BEM SUCEDIDO?
NÃO
SIM
518
GERAR TABELA REPAIR_SEQUENCE E ARMAZENAR POR MEIO DE MODO POR LOTE
FIM

FIGURA 5B

RESUMO

"TÉCNICAS PARA PROJETO DE REPARO ANTIENTROPIA ESCALONADO"

Várias modalidades da invenção revelada neste documento fornecem técnicas para realizar procedimentos de reparos antientropia distribuídos através de uma pluralidade de nós em uma rede de base de dados distribuída. Um nó incluído em uma pluralidade de nós dentro da rede de base de dados distribuída determina, antes de todos os outros nós incluídos na pluralidade de nós, que um primeiro procedimento de reparo antientropia terminou. O nó determina que um segundo procedimento de reparo antientropia está pronto para começar. O nó gera um escalonamento para executar uma ou mais operações associadas com o segundo procedimento de reparo antientropia. O nó grava o escalonamento em uma estrutura de dados de escalonamentos de reparos compartilhados para iniciar o segundo procedimento de reparo antientropia através de múltiplos nós incluídos na pluralidade de nós. Cada um dos nós incluídos na pluralidade de nós então realiza um reparo de nó com base no escalonamento.