

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

(21) **PI0711335-8 A2**



★ B R P I 0 7 1 1 3 3 5 A 2 ★

(22) Data de Depósito: 26/04/2007
(43) Data da Publicação: 30/08/2011
(RPI 2121)

(51) *Int.Cl.:*
G06F 12/16
G06F 15/16

(54) Título: **CRIANDO EFICIENTEMENTE
FREQUÊNTES CÓPIAS DE SEGURANÇA
CONSISTENTES DE APLICATIVO**

(30) Prioridade Unionista: 29/05/2006 IN 1283/DEL/2006

(73) Titular(es): Microsoft Corporation

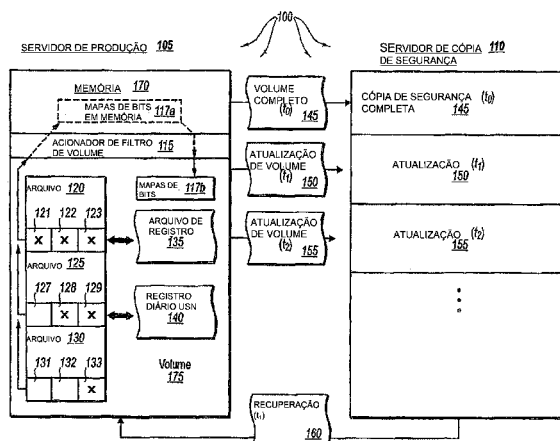
(72) Inventor(es): Abid Ali, Amit Singla, Karandeep Singh Anand,
Manoj K. Valiyaparambil, Robert M. Fries, Vinay S. Badami, Vivek
Sahasranaman

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

(86) Pedido Internacional: PCT US2007010304 de 26/04/2007

(87) Publicação Internacional: WO WO2007/139647 de
06/12/2007

(57) **Resumo:** CRIANDO EFICIENTEMENTE FREQUENTES CÓPIAS DE SEGURANÇA CONSISTENTES DE APLICATIVO. Os dados podem ser protegidos em um servidor de produção de um modo virtualmente contínuo, sem necessariamente impor severas restrições no aplicativo(s) fonte. Por exemplo, um servidor de produção pode criar uma cópia de segurança consistente de aplicativo de um ou mais volumes, as cópias de segurança correspondentes a um primeiro instante no tempo. Um acionador de filtro de volume pode monitorar mudanças de dados usando um mapa de bits em memória, enquanto um arquivo de registro e/ou registro diário de número de sequência de atualização pode manter rastreio de arquivos que foram modificados. As atualizações de volume são também consistentes para um instante (no futuro) no tempo. No próximo ciclo de cópia, tal como a cada poucos minutos (entretanto configurados), o acionador de filtro de volume passa cada mapa de bits em memória ao disco físico no servidor de produção. O servidor de produção então envia as atualizações ao servidor de cópia de segurança, que assim armazena cópias de segurança consistentes de aplicativo para o volume para múltiplos instantes de tempo.



"CRIANDO EFICIENTEMENTE FREQUENTES CÓPIAS DE SEGURANÇA CONSISTENTES DE APLICATIVO"

Fundamentos da Invenção

Como os sistemas computadorizados têm aumentado em popularidade, então têm
5 as necessidades de armazenar e fazer cópias de segurança eletrônicas de arquivos e outras
comunicações criadas pelos usuários e aplicativos associados com eles. Em geral, os sis-
temas de computador e dispositivos relacionados criam arquivos para uma variedade de
razões, tal como no caso geral de criar um documento de processamento de palavras em
um ambiente de trabalho, bem como criar um arquivo usado para propósitos de bancos de
10 dados mais sofisticados. Em adição, muitos desses documentos podem incluir produto de
trabalho valioso, ou informação sensível que deveria ser protegida. Um apreciará, portanto,
que há uma variedade de razões porque uma organização desejará copiar arquivos eletrôni-
cos em uma base regular, e desse modo criar uma restauração confiável de um arquivo ori-
ginalmente criado quando necessário.

15 Apesar de algumas das conveniências permitidas por muitos sistemas de cópias de
segurança convencionais, os mecanismos usados por muitos sistemas convencionais são
freqüentemente menos eficientes do que ótimos. Por exemplo, a capacidade de criar uma
cópia de segurança consistente de aplicativo pode ser um componente importante de alguns
sistemas de cópia de segurança. Uma cópia de segurança consistente de aplicativo (bem
20 como uma consistente de sistema de arquivos) é basicamente um conjunto copiado de da-
dos que é consistente em estado de arquivo para um ponto particular no tempo. Por exem-
plo, se um administrador de cópia de segurança copiou todos os dados em um dado volume
de servidor de produção, mesmo à medida que os dados podem estar no processo de se-
rem escritos ou atualizados, o estado de arquivo da cópia de segurança correspondente
25 pode não necessariamente ser consistente para um único ponto no tempo. Criar uma cópia
de segurança consistente de aplicativo, portanto, geralmente envolve o esforço adicional de
reconciliar o estado do arquivo.

No entanto, um pode apreciar que pode haver qualquer número de dificuldades as-
sociadas com a criação de cópias de segurança de dados consistentes de aplicativo. Por
30 exemplo, mecanismos convencionais para criar cópias de segurança geralmente envolvem
um programa de aplicativo, tal como um componente de cópia de segurança de um progra-
ma de correio eletrônico ou de aplicativo de banco de dados, chamando uma ou mais Inter-
faces de Programa de Aplicativo ("APIs") de cópia de segurança e/ou restauração. Em parti-
cular, o componente de cópia de segurança deve dizer às APIs para congelar as gravações
35 em certos dados de disco específicos no servidor de produção, e então criam uma cópia de
segurança (isto é, "réplica") dos dados. Infelizmente, geralmente não há modo simples de
copiar componentes para descrever seus dados às APIs de cópia de segurança no servidor

de produção. Ademais, complicando essa dificuldade é o fato de que pode ser às vezes um grande número de APIs de cópia de segurança que podem necessitar ser referenciadas durante um processo de cópia de segurança.

Em adição, o fato de que um aplicativo particular que criou certos dados está solicitando os serviços de cópia de segurança freqüentemente implica que menos dos dados de servidor de produção devem ser copiados em qualquer dado tempo. Por exemplo, os mecanismos de cópia de segurança convencionais são freqüentemente específicos de aplicativo com relação aos dados sendo copiados. Tais abordagens de cópia de segurança específicas de aplicativo freqüentemente envolvem executar múltiplas ocorrências do dado aplicativo durante o processo de cópia de segurança. Um apreciará, entretanto, que executar múltiplas ocorrências de um dado aplicativo pode ser ineficiente para um número de razões, se de uma perspectiva de gasto de custos ou de recursos.

Além disso, mesmo usando aplicativos para fornecer serviços de cópia de segurança pode ser às vezes ineficaz desde que as cópias de segurança específicas de aplicativo geralmente não fornecem cópias de ponto no tempo dos dados de aplicativo sem significativos gastos de recursos. Isso pode significar que um aplicativo pode não ser capaz de fornecer uma cópia de ponto no tempo consistente de aplicativo com alta freqüência (assim fornecendo pontos de recuperação altamente granulares) sem sobrecarregar ambos o servidor de produção e o servidor de cópia de segurança. Assim, cópias de segurança convencionais executadas por um aplicativo particular tipicamente não podem fornecer um "equipamento reserva" de cópias de segurança de ponto no tempo que são somente uns poucos minutos mais antigas.

Essa ausência geral de configurabilidade granular pode ser estender a uma ampla variedade de outros eventos em sistemas de cópias de segurança. Por exemplo, sistemas de cópias de segurança convencionais podem ser difíceis de configurar para tipos de arquivos, pastas específicas, ou localizações de pastas em um volume particular. Assim, pode haver dificuldades associadas com o fato de ter dados de servidor de produção de cópias de segurança de sistemas de cópias de segurança convencionais com melhor granularidade do que somente um ou mais volumes, ou somente arquivos inteiros, como oposto a cópias de segurança de somente aquelas partes dos arquivos que foram realmente modificados. Esses e outros problemas similares freqüentemente significam que o servidor de produção e o servidor de cópia de segurança são configurados para copiar e transferir mais dados entre eles do que o necessário, o que é claro pode afetar o desempenho do sistema e a largura de banda da rede. Em particular, o servidor de produção pode estar copiando e transferindo dados de arquivo que não mudaram, bem como arquivos inteiros que somente mudaram em pequenas partes. Por causa disso, o servidor de cópia de segurança pode também necessitar devotar mais capacidade de armazenamento do que necessário para fazer cópias de

segurança dos dados de servidor de produção.

Um apreciará, portanto, que cada um dos fatores mencionados acima (ou combinações desses) podem negativamente afetar os Objetivos do Ponto de Recuperação ("RPO"), que geralmente se referem a até aonde no tempo os dados necessitam ser recuperados de modo a uma organização reinicializar operações depois de um desastre. Os fatores mencionados acima podem também afetar negativamente os Objetivos do Tempo de Recuperação (RTO), que geralmente se referem a quanto tempo passará depois de um desastre antes que os dados necessários para reinicializar as operações podem ser recuperados. Ou seja, os sistemas de cópia de segurança convencionais são geralmente mal equipados para fornecer pontos de recuperação relativamente altos, particularmente em uma quantidade relativamente rápida de tempo sem sobrecargas indevidas em recursos de sistema.

Os sistemas de cópias de segurança presentes, portanto, encaram um número de dificuldades que podem ser abordadas.

Sumário da Invenção

As implementações da presente invenção resolvem um ou mais problemas na técnica com sistemas, métodos, e produtos de programa de computador configurados pelo menos em parte para otimizar objetivos de ponto de recuperação e objetivos de tempo de recuperação em sistemas de cópia de segurança. Por exemplo, em pelo menos uma implementação, as economias de recurso em um servidor de produção podem ser alcançadas monitorando mudanças para volumes de servidor de produção com um acionador de filtro de volume. Em adição, a largura de banda e os recursos de servidor de cópia de segurança podem ser usados eficientemente transferindo-se primariamente somente as mudanças incrementais (por exemplo, bytes, ou faixas de bytes de mudanças) a um servidor de cópia de segurança desde o ultimo ciclo de cópia. Como será apreciado mais completamente aqui, tais otimizações podem fornecer a capacidade de copiar os dados do servidor de produção de um modo virtualmente contínuo (ou quase contínuo) sem significantes drenos nos recursos do servidor de produção, recursos do servidor de cópia de segurança, e/ou preocupações de largura de banda de rede.

Por exemplo, um método a partir da perspectiva de um servidor de produção de replicar dados do servidor de produção de um modo consistente virtualmente contínuo pode envolver enviar uma cópia de dados de volume de um ou mais volumes de um servidor de produção a um servidor de cópia de segurança. Em tal caso, a cópia enviada de dados para o volume(s) geralmente será consistente (isto é, consistente de aplicativo ou consistente de sistema de arquivos) para um primeiro instante de tempo. Em adição, o método pode envolver identificar uma ou mais mudanças nos dados de volume via um ou mais arquivos de registro de volume. O método pode adicionalmente envolver, mediante a identificação de um evento de ciclo de réplica, salvar uma ou mais mudanças de dados em um ou mais arquivos

de registro de volume. Geralmente, uma ou mais mudanças de dados também serão consistentes por um segundo (isto é, subsequente) instante de tempo. Ainda ademais, o método pode envolver enviar ao servidor de cópia de segurança uma cópia de uma ou mais mudanças. Como tal, o servidor de cópia de segurança terá uma cópia dos dados de um ou mais volumes, onde os dados são válidos para um primeiro instante de tempo e um segundo instante de tempo.

Em contraste, um método a partir da perspectiva de um servidor de cópia de segurança para replicar os dados do servidor de produção de uma maneira consistente virtualmente contínua, pode envolver receber cópias de segurança de um ou mais volumes a partir de um servidor de produção. Em tal caso, cópias de segurança de um ou mais volumes são consistentes para um instante inicial de tempo. O método pode também envolver receber uma ou mais atualizações de cópia de segurança consistente de aplicativo, pelo menos uma das quais é uma atualização consistente a pelo menos uma das cópias de segurança de um ou mais volumes para um subsequente instante de tempo. Em adição, o método pode envolver receber uma solicitação de recuperação para dados que são válidos de acordo com o subsequente instante de tempo.

Além disso, o método pode também envolver identificar os dados solicitados para o subsequente instante de tempo em um ou mais volumes de servidor de cópia de segurança. Em tal caso, os dados solicitados incluem pelo menos uma parte de pelo menos uma atualização de cópia de segurança consistente de aplicativo. Em adição, o método pode envolver enviar os dados solicitados que são válidos para o subsequente instante de tempo ao servidor de produção.

Este sumário é fornecido para introduzir uma seleção de conceitos de uma forma simplificada que são adicionalmente descritos abaixo na Descrição Detalhada. Este sumário não pretende identificar características chave ou características essenciais do assunto reivindicado, nem pretende ser usado como um auxílio em determinar o escopo do assunto reivindicado.

Características e vantagens adicionais de implementações exemplificadas da invenção serão apresentadas na descrição que segue, e em parte estarão óbvias a partir da descrição, ou podem ser aprendidas pela prática de tais implementações exemplificadas. As características e vantagens de tais implementações podem ser realizadas e obtidas por meio dos instrumentos e combinações particularmente apontados nas reivindicações em anexo. Essas e outras características se tornarão mais completamente aparentes a partir da seguinte descrição e reivindicações em anexo, ou podem ser aprendidas pela prática de tais implementações exemplificadas como apresentado aqui.

Breve Descrição dos Desenhos

De modo a descrever a maneira na qual as vantagens e características da invenção

citadas acima e outras tais podem ser obtidas, uma descrição mais particular da invenção brevemente descrita acima será abordada por referência a modalidades específicas dessa que são ilustradas nos desenhos em anexo. O entendimento desses desenhos representa somente modalidades típicas da invenção e não são, portanto, consideradas como limitante de seu escopo, a invenção será descrita e explicada com especificidade adicional e detalhes através do uso dos desenhos em anexo nos quais:

A Figura 1A ilustra um diagrama de visão geral arquitetural de acordo com uma implementação da presente invenção na qual um servidor de produção cria cópias de segurança consistente de aplicativo (ou de sistema de arquivos) e envia aquelas cópias de segurança a um servidor de cópia de segurança;

A Figura 1B ilustra um diagrama de visão geral de acordo com uma implementação da presente invenção na qual um acionador de filtro de volume monitora mudanças em um volume usando memória de sistema e um ou mais discos físicos; e

A Figura 2 ilustra fluxogramas de métodos compreendendo uma seqüência de ações executadas a partir da perspectiva de um servidor de produção e um servidor de cópia de segurança de acordo com implementações da presente invenção.

Descrição Detalhada da Invenção

A presente invenção se estende a sistemas, métodos e produtos de programa de computador configurados pelo menos em parte para otimizar objetivos de ponto de recuperação e objetivos de tempo de recuperação em sistemas de cópia de segurança. Por exemplo, em pelo menos uma implementação, as economias de recursos em um servidor de produção podem ser alcançados monitorando mudanças nos volumes do servidor de produção com um acionador de filtro de volume. Em adição, a largura de banda de rede e os recursos de servidor de cópia de segurança podem ser usados eficientemente transferindo-se primariamente somente as mudanças incrementais (por exemplo, bytes, ou faixas de bytes de mudanças) a um servidor de cópia de segurança desde o último ciclo de réplica. Como será apreciado mais completamente aqui, tais otimizações podem fornecer a capacidade de fazer cópias de segurança de dados do servidor de produção de um modo virtualmente contínuo (ou quase contínuo) sem significativos drenos nos recursos do servidor de produção, recursos do servidor de cópia de segurança, e/ou largura de banda de rede.

Como será apreciado mais completamente a partir da seguinte especificação e reivindicações, as implementações da presente invenção podem alcançar uma ampla faixa de "objetivos de tempo de recuperação" renovando o servidor de cópia de segurança com um "instantâneo completo" dos dados do servidor de produção. Em adição, as implementações da presente invenção incluem um acionador de filtro de volume que pode ser implementado no servidor de produção. Como será apreciado mais completamente aqui, o acionador de filtro de volume pode ser configurado para monitorar mudanças em bytes (e/ou blocos de

bytes) no volume(s) de servidor de produção. O servidor de produção pode então ser configurado para enviar um instantâneo inteiro (ou cópia de segurança) enviando somente aqueles bytes mudados (ou blocos de bytes) ao servidor de cópia de segurança. Como tal, o uso de um acionador de filtro de volume pode suavizar a sobrecarga em recursos que devem, de outra forma, ser consumidos quando movendo um instantâneo completo de dados do servidor de produção para um servidor de cópia de segurança.

Além disso, e como um resultado dessas e outras características, o servidor de produção pode fornecer a um servidor de cópia de segurança, cópias de segurança virtualmente contínuas de vários instantâneos (isto é, cópias de segurança iniciais e cópias virtuais subseqüentes) “consistentes” (isto é, consistente de aplicativo e/ou consistente de sistema de arquivos) que são colocadas juntas no tempo. Em adição, desde que cada atualização à cópia de segurança é consistente de aplicativo (e/ou consistente de sistema de arquivo), ou válida para um instante específico de tempo, a diferença entre cada atualização também será consistente de aplicativo. Como tal, as implementações da presente invenção fornecem a um usuário a capacidade de recuperar uma ampla faixa de dados consistentes de aplicativo (por exemplo, a partir do nível de arquivo, nível de banco de dados, e até nível de servidor de produção inteiro) com granularidade razoavelmente alta (por exemplo, somente uns poucos minutos mais antigo) com muito menos sobrecarga do que de outra forma deve ser necessário.

Geralmente, há uma variedade de formas de acordo com as implementações da presente invenção de implementar serviços de cópia de segurança consistentes contínuos. Em pelo menos um sentido muito básico, criar uma cópia de segurança consistente inclui criar uma cópia de linha de base (por exemplo, 145) de um ou mais volumes (por exemplo, 175), e então suplementar essa cópia de linha de base com atualizações consistentes incrementais (por exemplo, 150, 155) a um ou mais volumes. Por exemplo, a Figura 1A mostra que o servidor de produção 105 cria pelo menos uma cópia de segurança de linha de base (por exemplo, 145) de dados no volume(s) selecionado(s) 175. Em adição a simplesmente criar a cópia de linha de base 145, um administrador de cópia de segurança no sistema 100 deve usar qualquer número ou tipo de mecanismos para tornar a cópia 145 consistente.

Em uma implementação, o administrador de cópia de segurança deve usar um agente de réplica (não mostrado), instalado no servidor de cópia de segurança 110 e/ou servidor de produção 105 para guiar processos de réplica. Durante um ciclo de réplica, por exemplo, o agente de réplica deve ser configurado para instruir quaisquer um ou mais escritores de aplicativo apropriados no servidor de produção 105 a momentaneamente manter atividades de escrita em quaisquer um ou mais volumes (por exemplo, uma base de dados pode abranger vários volumes, e vários aplicativos podem usar o mesmo volume com dife-

rentes programações de réplica) para um único ponto no tempo. (Para uma cópia de segurança de compartilhamento de arquivo, um escritor de aplicativo pode nem estar envolvido). Isso permite que o agente de réplica desse modo crie uma cópia de segurança de único ponto no tempo (isto é, “cópia virtual”) de um ou mais volumes.

5 O agente de réplica pode também fornecer instruções a cada escritor de aplicativo para executar certas funções em seus dados de interesse, para desse modo assegurar que todos os dados e metadados são consistentes para o ponto no tempo do ciclo de réplica. Para aplicativos mais simples que não têm um escritor de aplicativo ou plug-in correspondente associado com eles, o agente de réplica deve ser configurado para simplesmente ins-
10 truir aqueles aplicativos a congelar e desligar durante o ciclo de réplica. Os agentes, componentes, e funções mencionados acima para criar cópias de segurança consistente podem ser fornecidos em pelo menos uma implementação no ambiente MICROSOFT, por exemplo, com um Serviço de Cópia Virtual de Volume (“VSS”).

Em qualquer evento, tendo congelado as escritas de interesse em um volume parti-
15 cular (ou volumes) e para um instante particular de tempo, o servidor de produção 105 pode então obter e enviar uma cópia do volume(s) (ou alternativamente somente aquelas pastas selecionadas, arquivos, ou tipos de arquivos) de interesse. Por exemplo, a Figura 1A mostra que o servidor de produção 105 pode fornecer essa cópia de linha de base inicial 145 ao
servidor de cópia de segurança 110. Geralmente, o servidor de produção 105 pode fornecer
20 a cópia de linha de base 145 qualquer número de formas. Em uma implementação, por exemplo, o servidor de produção 105 simplesmente envia a cópia 145 pela conexão de rede. Em outras implementações, tal como onde a largura de banda da rede pode ser mais limitada, um administrador de cópia de segurança pode transferir a cópia de volume para fita (ou
outro nó de armazenamento intermediário – não mostrado) e mais tarde conecta essa fita ao
25 servidor de cópia de segurança 110. Entretanto executado, o servidor de produção 105 fornece ao servidor de cópia de segurança 110 pelo menos uma cópia de linha de base consistente de todos os dados (isto é, válido pelo tempo “t0”) para o volume, pasta, ou conjunto de arquivos de interesse que compartilha o mesmo ciclo de réplica.

Depois de fornecer uma ou mais cópias de linha de bases de um ou mais volumes
30 do servidor de produção, o servidor de cópia de segurança 110 pode continuar a receber atualizações para a cópia(s) de segurança de linha de base. Por exemplo, o servidor de cópia de segurança 110 pode continuar a copiar o servidor de produção 105 em uma ampla variedade de programações de réplica configuráveis, tal como aquelas na faixa na ordem de
aproximadamente 5-10 minutos, 10-15 minutos, 15-30 minutos, ou seus similares. Geral-
35 mente, o nível ao qual o administrador de cópia de segurança configura os ciclos de réplica será o nível de granularidade ao qual um pode acessar “pontos de recuperação” particulares.

Como previamente discutido, ordinariamente um nível relativamente alto de granularidade na acessibilidade de cópias de segurança de ponto no tempo poderia ser proibitivamente caro em recursos para alguns sistemas de cópia de segurança. Assim, para criar o nível acima mencionado de granularidade de “pontos de recuperação” sem necessariamente comprometer o “tempo tomado para recuperação” (por exemplo, sem incorrer em significativa sobrecarga), as implementações da presente invenção podem fornecer um número de importantes componentes e funções.

Em uma implementação discutida mais completamente aqui, por exemplo, um acionador de filtro de volume 115 pode ser usado para monitorar mudanças iterativas em qualquer de um ou mais volumes (por exemplo, 175) no servidor de produção 105, com ou mapas de bits em memória, ou tornando bytes alterados particulares (ou blocos de bytes) em um arquivo de registro de volume em disco. Geralmente, um acionador de filtro de volume (por exemplo, 115) será independente de como “instantâneos” baseados em hardware ou em software são implementados no servidor de produção 105. Um apreciará, entretanto, que um acionador de filtro de volume 115 não é necessariamente exigido, e funções similares podem ser executadas por outros componentes, como também discutido aqui. Em implementações adicionais ou alternativas, o servidor de produção 105 pode também monitorar mudanças no volume (por exemplo, mudanças nos arquivos 120, 125, 130, etc.) através do uso de mecanismos de monitoramento de cópia virtual convencionais, e/ou registros diários de Número de Sequência de Atualização (isto é, “registro diário USN” 140), ou seus similares. Com particular referência ao ambiente operacional MICROSOFT, por exemplo, tais componentes são fornecidos através do uso combinado de um Serviço de Cópia Virtual de Volume (“VSS”) e um registro diário USN.

Geralmente, um arquivo de registro de volume (por exemplo, 135) pode compreender todas as mudanças em um volume durante um ciclo de réplica específico (por exemplo, deslocamento de volume, comprimento de mudança de dados) para cada escrita no volume, e/ou mapa de bits de mudanças em memória. Quando um ciclo de réplica específico ocorre, um arquivo de registro de volume existente (por exemplo, 135) é congelado, e um novo arquivo de registro de volume (não mostrado) pode ser criado para reunir mudanças para o próximo ciclo de réplica. Em uma implementação, as mudanças de nível de volume podem ser enviadas diretamente ao servidor de cópia de segurança 110, sem qualquer informação correlata adicional. As atualizações correspondentes enviadas ao servidor de cópia de segurança 110 devem ser então aplicadas na réplica como “byte n” (ou “bloco de bytes n mudado para n+1”). Em implementações adicionais ou alternativas, os dados de volume no servidor de produção 105 podem também ser correlacionados com a informação de registro diário USN (ou componente relacionado) 140.

Em particular, um registro diário USN (por exemplo, 140) compreende informação

estampada no tempo de tais atividades com relação aos dados de nome de arquivo no sistema de arquivos. Componentes similares ou idênticos ao registro diário USN 140 são também referidos como filtros de mudança, ou registros diários de mudança, ou seus similares. Referência específica a um “registro diário USN 140” aqui, portanto, é feita primariamente a título de conveniência. Em qualquer evento, e com relação à atividade do sistema de arquivos, o servidor de produção 105 pode combinar dados do arquivo de registro de volume (por exemplo, 135) e dados de registro diário de mudança (por exemplo, registro diário USN 140) para correlacionar o tempo, o tipo de atividade, e o nome do arquivo, etc. das várias escritas no volume 175.

Ao longo dessas linhas, o registro diário 140 pode também ser usado em conjunto com o arquivo de registro de volume 135 para correlacionar tais coisas como o endereço de uma mudança de byte particular ou “bloco de bytes” (bem como arquivos correspondentes para a mudança). Em particular, cada arquivo em um volume pode ser pensado como um conjunto aberto de bytes endereçáveis, bem como um conjunto aberto de blocos de bytes de comprimento fixo endereçáveis. Em alguns casos, monitorar e transferir blocos de bytes (ao invés de bytes individuais) pode ser uma forma mais eficiente de monitorar e transferir mudanças, bem como determinar quanto espaço pode ser necessário para propósitos de cópia de segurança. Em particular, isto é, em parte, como blocos de bytes representam um nível de granularidade que é usualmente um pouco menos do que o de um arquivo inteiro, mas maior do que o de um único byte. Como tal, a Figura 1A mostra que o servidor de produção 105 registra as várias mudanças de byte ou de bloco de bytes em seus arquivos a serem protegidos.

Por exemplo, a Figura 1A mostra que o servidor de produção 105 registra que bytes (ou “blocos de bytes”) 121, 122, e 123 de arquivo 120 mudaram (por exemplo, 120 é um novo arquivo) desde que o último ciclo de réplica (por exemplo, últimos 5, 10, 15, 20, 25 ou 30 minutos, etc.). Similarmente, embora o arquivo 125 compreenda bytes (ou blocos de bytes) 127, 128 e 129, somente os bytes 128 e 129 mudaram; e onde o arquivo 130 compreende bytes 131, 132, e 133, somente o byte 133 mudou desde o último ciclo de réplica. Geralmente, o servidor de produção 105 pode registrar esses bytes alterados em uma cópia virtual somente de leitura do volume, pasta, ou arquivos relevantes. Como previamente mencionado, o servidor de produção 105 pode também armazenar esses bytes alterados para um arquivo de registro de volume como mapas de bits em memória (por exemplo, usando um bit por bloco de dados no volume) que são mais tarde passados para o disco físico durante a réplica. Entretanto registrado ou monitorado, e em um tempo apropriado (isto é, o próximo ciclo de réplica), o servidor de produção 105 pode então preparar somente essas mudanças de arquivo (isto é, 121, 122, 123, 128, 129, 133, etc.) a serem enviadas o servidor de cópia de segurança 110.

Nesse exemplo particular, as mudanças (isto é, 121, 122, 123, 128, 129, 133, etc.) são válidas para o ponto no tempo mais recente (isto é, " t_1 "), e assim consistentes (isto é, consistentes de aplicativo ou consistentes de sistema de arquivos). Notavelmente, como esses bytes (ou blocos de bytes) representam a diferença entre duas cópias consistentes de segurança de ponto no tempo, o aplicativo desses bytes (ou blocos de bytes) no servidor de cópia de segurança 110 pode também ser consistente. Depois de identificar essas mudanças de dados, o servidor de produção 105 pode enviar essas mudanças como atualizações 150 (para tempo t_1) ao servidor de cópia de segurança 110. Similarmente, o servidor de produção 105 pode identificar e enviar cada próximo conjunto de mudanças de dados incrementais (isto é, mudanças de dados registradas para um ou mais volumes) em uma próxima atualização incremental para o próximo ciclo de réplica. Por exemplo, a Figura 1A também mostra que o servidor de produção 105 prepara e envia a atualização 155 (para o tempo t_2) ao servidor de cópia de segurança 110, e assim por diante.

Um apreciará que a partir do ponto no qual o servidor de produção 105 lê os dados modificados e prepara a mensagem 150, pode haver mudanças adicionais nos dados do volume, que poderia tornar os dados sendo lidos de outra forma inconsistentes (isto é, consistentes para o tempo " t_1 " e válidos para um tempo subsequente). Como previamente discutido com relação à cópia de linha de base 145, portanto, um serviço de cópia virtual de volume (ou outro mecanismo tipo VSS) pode ser usado em pelo menos uma implementação para ler dados somente para o instante particular congelado de tempo, e não quaisquer mudanças subsequentes a essa. Isso pode ajudar a assegurar que as atualizações de instantâneo 150 (bem como 155, etc.) são consistentes até o instante de tempo indicado no qual as operações do instantâneo (também referido como atualização de cópia de segurança ou atualização) começaram.

Mediante o recebimento, o servidor de cópia de segurança 110 pode armazenar cada cópia de segurança e atualização(ões) correspondente em um volume de réplica particular. Em uma implementação, o servidor de cópia de segurança 110 armazena as cópias de segurança e atualizações na mesma alocação de volume do mesmo meio de armazenamento. Em outras implementações, o servidor de cópia de segurança 110 (e/ou servidores de cópia de segurança adicionais ou nós de armazenamento) pode armazenar as cópias de segurança e correspondentes atualizações em volumes separados, e até em meios de armazenamento separados, entretanto desejados pelo administrador de cópia de segurança.

Em alguns casos, e devido à natureza das cópias de segurança iterativas virtualmente contínuas do servidor de produção 105, um administrador de cópia de segurança pode necessitar sincronizar um número de aspectos dos dados com os dados do servidor de produção 105. Por exemplo, pode haver casos de falha durante um ciclo de réplica, e/ou por vários ciclos de réplica, tal como pode ser devido a falha de rede, sobrecarga de registro

(por exemplo, re-arranjo de registro diário USN, etc.). Em uma implementação, portanto, um administrador de cópia de segurança pode executar uma validação ou correção criando um novo instantâneo completo básico (por exemplo, similar a 145) do servidor de produção 105. O administrador de cópia de segurança pode então executar (por exemplo, via o servidor de

5 produção 105) uma comparação de soma de verificação (ou outra validação) entre um instantâneo dos dados no servidor de produção 105, e os dados no servidor de cópia de segurança 110. Quaisquer dados errantes no servidor de cópia de segurança 110 podem então ser fixos, se necessário.

Com particular referência aos componentes operacionais WINDOWS, por exemplo,

10 essa soma de verificação pode ser executada em pelo menos uma implementação usando Compressão Diferencial Remota ("RDC") usada em WINDOWS SERVER 2003. Em alguns casos, o uso de um tipo de mecanismo RDC pode ser preferencial em um ambiente de Rede de Área Ampla ("WAN"). Em outra implementação, tal como pode ser preferencial em uma Rede de Área Local ("LAN"), o administrador de cópia de segurança pode dividir cada

15 arquivo em um instantâneo em conjuntos de "blocos" (por exemplo, blocos de bytes) e então computa somas de verificação para cada bloco.

Em qualquer evento, devido em parte ao nível de granularidade de réplica fornecido pelas implementações da presente invenção, se um usuário solicita uma versão particular de um arquivo (ou outra representação de dados) que é somente uns poucos minutos mais

20 antigo (por exemplo, necessário a partir de uma falha de computador pessoal recente), o usuário pode enviar uma solicitação ao servidor de cópia de segurança 110 para essa versão particular de um arquivo. Por exemplo, um usuário deve solicitar uma cópia particular de arquivo 120 que estava válida há 5 minutos atrás (por exemplo, " t_0 ", ou antes das atualizações 121, 122, 123). Similarmente, um administrador pode solicitar (não mostrado) uma re-

25 produção inteira de volume ou volumes 175.

Mediante o recebimento da solicitação, e dependendo da natureza da solicitação, o servidor de cópia de segurança 110 pode então encontrar os dados solicitados como apropriado. Por exemplo, com relação a dados de sistema de arquivos básicos, cada atualização de volume 175 poderia conter uma cópia completa de dados solicitados. Assim, o servidor

30 de cópia de segurança 110 deve somente necessitar identificar o tempo solicitado pelo usuário, identificar os dados na atualização correspondente para esse tempo, e então fornecer uma cópia desses dados de volta ao usuário (por exemplo, mensagem de recuperação 160).

Em outros casos, tal como com correio ou outros tipos de dados de aplicativo de banco de dados, cada atualização incremental (por exemplo, 150, 155) recebida pelo servidor de cópia de segurança 110 deve somente conter uma atualização incremental para os

35 dados solicitados. Assim, o servidor de cópia de segurança 110 poderia ser configurado para reproduzir cada atualização incremental a partir do ponto de recuperação solicitado de

volta à última linha de base completa. O servidor de cópia de segurança 110 pode então combinar os dados solicitados identificados durante a reprodução (por exemplo, 145, 150, 155, ou t_{0-n}), até alcançar o tempo especificado na solicitação. Quando todos os dados relevantes a partir da cópia de segurança original e atualizações correspondentes são combinados e preparados, o servidor de cópia de segurança 110 pode então enviar a resposta de recuperação (por exemplo, 160), que é válida de acordo com o tempo solicitado. Por exemplo, a Figura 1A mostra que o servidor de cópia de segurança 110 envia a resposta 160, que indica que os dados recuperados são válidos pelo tempo " t_1 ".

Na implementação anterior, o servidor de cópia de segurança 110 pode, portanto, necessitar suporte de aplicativo para reproduzir as atualizações incrementais. Em outra implementação, a cópia completa de linha de base e quaisquer atualizações incrementais correspondentes entre a linha de base completa e o ponto no tempo solicitado podem simplesmente ser copiadas de volta ao servidor de produção 105. Um escritor de aplicativo convencional (por exemplo, um escritor de aplicativo em uma estrutura de serviço de cópia virtual) no servidor de produção 105 pode então reproduzir os registros na cópia de segurança completa.

Geralmente, o tempo que decorre entre a solicitação por dados particulares e a resposta correspondente pode ser uma função de pelo menos duas partes:

1. Tempo para transferir os dados do servidor de cópia de segurança 110 ao servidor de produção 105; e

2. Tempo para o servidor de cópia de segurança 110 (por exemplo, via um agente de cópia de segurança relevante) para completar a recuperação.

É claro, o tempo para transferir dados do alvo à fonte é geralmente uma função da largura de banda da rede disponível, bem como velocidades de disco e uso de recursos no servidor de cópia de segurança 110 e no servidor de produção 105. Em contraste, o tempo para criar uma recuperação particular é tipicamente uma função do tempo exigido para recuperar uma cópia completa dos dados de servidor de produção a partir de uma dada linha de base, e o tempo exigido para identificar e reproduzir atualizações acumuladas (por exemplo, " t_{n-1} ") acumuladas a partir da linha de base de modo a recuperar um ponto específico no tempo. Um apreciador, portanto, que o tempo de recuperação pode ser amplamente melhorado limitando a quantidade de atualizações que o servidor de cópia de segurança 110 (ou servidor de produção 105) tem que reproduzir novamente para qualquer dada solicitação de recuperação, tal como criando cópias completas de linha de base periódica (por exemplo, 145).

Como previamente mencionado, uma forma de limitar a quantidade de atualizações incrementais que o servidor de cópia de segurança 110 deve necessitar reproduzir pode envolver criar um novo instantâneo de linha de base "completo" periodicamente. Como criar

e enviar um novo instantâneo completo ao servidor de cópia de segurança 110 pode ser dispendioso em recursos (por exemplo, largura de banda de rede, recursos do servidor de produção 105, e a quantidade de espaço em disco do servidor de cópia de segurança necessária) em alguns casos, as implementações da presente invenção também fornecem a criação de “instantâneos completos inteligentes”. Esses instantâneos completos inteligentes são efetivamente uma demarcação de linha de base de um ponto no tempo pré-determinado. Por exemplo, cada período pré-determinado, tal como a cada duas semanas, o servidor de cópia de segurança 110 pode desenrolar duas semanas a importância de atualizações incrementais (por exemplo, 150, 155, etc.) junto com a última cópia de linha de base de dados (por exemplo, 145, ou mais nova), e assim criar essencialmente uma nova cópia “ t_0 ” de dados do servidor de produção 105.

De modo a desenrolar cada uma dessas atualizações incrementais juntas eficientemente, o servidor de cópia de segurança 110 pode ser configurado para monitorar todas as escritas para o volume do servidor de produção 105 desde o último instantâneo completo. Em pelo menos uma implementação, por exemplo, o servidor de cópia de segurança 110 implementa o acionador de filtro de volume 115 no servidor de produção 105 para monitorar mudanças no volume (isto é, um ou mais volumes), e armazena essas escritas na memória 170 do servidor de produção 105 durante cada ciclo de réplica. Por exemplo, a Figura 1A (ver também Figura 1B) mostra que o acionador de filtro de volume 115 faz interface entre o volume 175 e a memória 170 no servidor de produção 105. Como será entendido mais completamente na Figura 1B, cada hora em que uma mudança é feita nos dados do volume, o acionador de filtro de volume 115 pode gravar essa mudança (ou conjunto de mudanças) em um arquivo de registro de volume 135. Em pelo menos uma implementação, essas mudanças são gravadas na memória de sistema 170 como um mapa de bits em memória (por exemplo, 117a) para cada um dos um ou mais volumes. Quando o servidor de produção 105 está pronto para um ciclo de réplica particular, o acionador de filtro de volume 115 pode então passar todos os mapas de bits em memória ao volume 175, e o servidor de produção 105 pode então enviar os dados correspondentes ao servidor de cópia de segurança 110.

Por exemplo, a Figura 1B mostra que a memória 170 foi usada para reunir dados de instantâneo correspondentes a vários mapas de bits em memória. Em particular, a Figura 1B mostra que o acionador de filtro de volume 115 identifica certas mudanças de arquivo (isto é, mudanças de arquivos 121, 122, 123, etc.), e subseqüentemente armazena essas mudanças como mapas de bits em memória correspondentes 193, 195, etc. em alocação de memória 190a. Nesse exemplo particular, o acionador de filtro de volume 115 armazena todas as mudanças aos um ou mais volumes correspondentes desde o último ciclo de réplica (isto é, instantâneo 185a – “ t_2 ”), e, como tal, cada mapa de bits 193, 195, etc. é válido para o instante de tempo mais recente no instantâneo correspondente (isto é, 190^a, instan-

tâneo “ t_3 ”).

Quando o ciclo de réplica é disparado, tal como devido às instruções recebidas a partir de um agente de réplica, o acionador de filtro de volume 115 transfere todos os mapas de bits para instantâneo 190a (isto é, os mapas de bits 193, 195, etc.) à alocação 190b do volume 175 apropriada. Por exemplo, a Figura 1B mostra que as partes de instantâneo de memória 180a e 185a foram esvaziadas desde o ciclo de réplica para o qual elas foram geradas já passou. Além disso, as alocações 180b, 185b, etc. do volume 175 correspondente agora contêm “todos os mapas de bits” 183, 187 que foram previamente armazenados em partes de memória 180a, 185a, respectivamente. No fim, quando um instantâneo correspondente ao conjunto de mapas de bits é apagado, os mapas de bits (por exemplo, 183, 187) podem permanecer no volume (por exemplo, 175).

Esses mapas de bits em memória (por exemplo, 193, 195) podem ser criados e implementados em um número de diferentes formas. Em uma implementação, por exemplo, o servidor de cópia de segurança 110 obtém um instantâneo de cópia virtual (por exemplo, instantâneo 150) do volume de servidor de produção 105, um Controle de Entrada/Saída (“IOCTL”) pode ser enviado ao provedor de cópia virtual (software ou hardware). Esse IOCTL pode ser interceptado pelo acionador de filtro de volume 115 para dividir o mapa de bits ativo. Em resposta, e durante a criação da cópia virtual, o acionador de filtro de volume 115 sincroniza a divisão criando um conjunto congelado de mapa de bits (por exemplo, 180a/189b, 185a/185b) e um novo conjunto ativo de mapa de bits (por exemplo, 190a/190b). Em outra implementação alternativa, uma entidade que é capaz de esgotar a área de diferença de cópia virtual (por exemplo, área de diferença VSS) poderia estar ciente das mudanças de nível de volume e também ciente do sistema de arquivo/USN. Como tal, essa entidade poderia fornecer uma abstração que forneceria o conjunto de arquivos alterados e o conjunto de mudanças nos arquivos. Uma réplica apropriada ou aplicativo de cópia de segurança pode então usar essa infra-estrutura para alcançar a réplica.

Quando um ciclo de réplica é disparado, o acionador de filtro de volume 115 passa os mapas de bits congelados para o disco de modo a reduzir o uso da memória 170. O acionador de filtro de volume 115 pode também expor um ou mais IOCTLs que podem ser consultados para todas as mudanças que ocorreram desde o instantâneo mais recente (por exemplo, “ t_{n-1} ”). Em uma implementação, consultar esses IOCTLs retorna todos os mapas de bits que foram acumulados desde o instantâneo mais recente. Um apreciará que usando os mapas de bits em memória como descrito aqui para monitorar mudanças podem ser muito eficientes, pelo menos em parte desde que os mapas de bits em memória não tendem a afetar significativamente os recursos do servidor de produção 105.

Em uma implementação alternativa, e para identificar cada uma dessas mudanças monitoradas, o servidor de cópia de segurança 110 pode também identificar o conjunto de

arquivos que são mudados usando, por exemplo, o registro diário USN 140 (ou usando outros metadados de arquivo monitorados). O servidor de cópia de segurança 110 (ou componente relevante) pode então consultar o sistema de arquivos do servidor de produção 105 para as extensões de arquivo que cada arquivo alterado ocupa. Uma interseção entre as

5 extensões de arquivo consultado a partir do sistema de arquivos e as extensões de arquivo que o acionador de filtro de volume 115 relata pode fornecer as extensões do arquivo que mudou desde o último ciclo de réplica, e assim permite que certos arquivos (por exemplo, arquivos de base de dados) sejam excluídos de certos processos de réplica. O servidor de cópia de segurança 110 (ou componente relevante) pode então repetir esse processo para

10 cada arquivo alterado (ou como relatado por um registro diário USN ou por um documento de metadados similarmente configurado).

Com particular referência a um registro diário USN para monitorar mudanças no servidor de produção 105, um apreciará que um Número de Referência de Arquivo (“FRN”) no servidor de produção 105 deve não combinar com o FRN para o mesmo arquivo arma-

15 zenado no servidor de cópia de segurança 110. Assim, de modo a enviar as mudanças monitoradas do servidor de produção 105 ao servidor de cópia de segurança 110, pode ser importante em alguns casos computar um caminho combinado correto até o arquivo alterado particular. Por exemplo, um volume deve ter as seguintes mudanças, que envolvem modificar um caminho no servidor de produção 105 para o arquivo “y.txt”:

- 20 1)Modify C:\a\b\c\y.txt
- 2)Rename C:\a\b to C:\p\q\r
- 3)Modify C:\p\q\r\b\c\y.txt
- 4)Delete C:\a

No exemplo ilustrado imediatamente acima, o caminho original “a\b” no servidor de

25 produção 105 é renomeado como caminho “p\q\r”, e o diretório “a” do caminho original é apagado. Isso deixa o caminho no servidor de produção 105 até o arquivo “y.txt” como sendo “C:\p\q\r\b\c\y.txt”. No entanto, essas mudanças de caminho para “y.txt” no servidor de produção 105 podem não automaticamente resultar em mudanças ao caminho no servidor de cópia de segurança 110. Em particular, o caminho até y.txt no servidor de cópia de segu-

30 rança 110 geralmente permanecerá “a\b\c\y.txt”.

Na hora de um instantâneo, o servidor de produção 105 deve também gravar no registro diário USN as seguintes mudanças mostradas abaixo, por exemplo, registros 1-5:

- 1){rsn = modify,FRN = yFRN, parentFRN = cFRN, nome de arquivo = y.txt}
- 2){rsn = rename-old,FRN = bFRN, parentFRN = aFRN, nome de arquivo = b}

35 3){rsn = rename-new,FRN = bFRN, parentFRN = rFRN, nome de arquivo = y.txt}

- 4){rsn = modify,FRN = yFRN, parentFRN = cFRN, nome de arquivo = y.txt}
- 5){rsn = delete,FRN = aFRN, parentFRN = root, nome de arquivo = a}

Em adição ao exemplo anterior, o estado do volume do servidor de produção relevante 105 na hora da réplica é mostrado como "C:\p\q\r\b\c\y.txt".

De modo a enviar mudanças para o registro 1 ao servidor de cópia de segurança 110 nesse exemplo particular, o servidor de produção 105 necessita restaurar o caminho de arquivo para "y.txt" a partir do servidor de cópia de segurança 110. Especificamente, devido à mudança de caminho descrita acima, o caminho para "y.txt" no servidor de produção 105 no instantâneo para "c-FRN" é C:\p\q\r\b\c\y.txt, que é diferente de seu caminho no servidor de cópia de segurança 110 (isto é, C:\a\b\c\y.txt). As implementações da presente invenção podem resolver esse problema com pelo menos duas abordagens alternativas. Em uma implementação, por exemplo, o registro diário USN pode simplesmente restaurar e armazenar metadados de caminho a partir do servidor de cópia de segurança 110 em um banco de dados relacional, e assim caminhos de arquivos continuamente correlatos no servidor de produção 105 e no servidor de cópia de segurança 110.

Em uma implementação alternativa, o servidor de produção 105 pode varrer o registro diário USN duas vezes. No primeiro passo, o servidor de produção 105 pode correlacionar essa informação através de varreduras iterativas (ou "passos") do registro diário USN 140. Em uma varredura, por exemplo, o servidor de produção 105 armazena em cache cada renomeação de pasta. Em um segundo passo, o servidor de produção 105 pode computar o caminho correspondente no servidor de cópia de segurança 110 baseado nas renomeações em cache e caminhos atuais. Por exemplo, no fim de um primeiro passo, o servidor de produção 105 deve armazenar em cache a seguinte informação sobre os diretórios apagados e/ou renomeados:

{FRN = bFRN, replica_parent = aFRN, replica_name = b}

{FRN = aFRN, replica_parent = root, replica_name = a}

Para computar o caminho de arquivo para y.txt, o servidor de produção 105 primeiro identifica que o FRN pai (isto é, "Número de Referência de Arquivo") para y.txt no registro "1" é cFRN. Em uma próxima etapa, o servidor de produção 105 computa o nome de arquivo para cFRN, bem como o nome de arquivo do FRN pai. O servidor de produção 105 então procura na cache antes de consultar o sistema de arquivos. Como, nesse exemplo, o cache não tem entrada de nome de arquivo para o cFRN pai, o servidor de produção 105 consulta o sistema de arquivos, e determina que o nome do arquivo é c, e que o pai é bFRN.

O servidor de produção 105 então computa o nome de arquivo de bFRN, bem com o nome de arquivo pai correspondente do bFRN. Como antes, o servidor de produção 105 primeiro procura antes de consultar o sistema de arquivos. Como a cache tem uma entrada para bFRN neste exemplo, o servidor de produção 105 determina que o nome de arquivo é b, e que o pai é aFRN. O servidor de produção então computa o nome de arquivo de aFRN e o nome de arquivo pai de aFRN. Novamente, o servidor de produção 105 primeiro procura

na cache antes de consultar o sistema de arquivos, e como a cache tem uma entrada para aFRN nesse exemplo, o servidor de produção 105 determina que o nome de arquivo de aFRN é “a”, e o FRN pai é “root”.

5 Por fim, o servidor de produção 105 computa o caminho final como “c:\a\b\c\y.txt”. A seguir, quando o registro 3 (rename-new) é processado no segundo passo, a cache é atualizada para o novo nome de arquivo pai como segue.

{FRN = bFRN, replica_parent = rFRN, replica_name = b}

{FRN = aFRN, replica_parent = root, replica_name = a}

10 Quando o servidor de produção 105 processa o registro 4, apesar do registro ser idêntico ao registro 1, o caminho computado para y.txt é agora “C:\p\q\r\b\c\y.txt”. O caminho computado é diferente nesse caso particular desde que o “bFRN” pai na cache é agora “rFRN”. Conseqüentemente, o texto anterior ilustra como o algoritmo de dois passos pode auxiliar a otimizar a quantidade de dados que o servidor de produção 105 transfere ao servidor de cópia de segurança 110. Em particular, o texto anterior descreve múltiplas formas
15 nas quais o servidor de produção 105 pode identificar na hora do segundo passo quaisquer arquivos criados, bem como quaisquer arquivos que são modificados e então apagados, e assim apropriadamente caminhos de arquivo correlatos com o servidor de cópia de segurança 110.

Conseqüentemente, as Figuras 1A-1B e o texto correspondente fornecem um número de sistemas, componentes e mecanismos para eficientemente copiar os dados de um
20 servidor de produção de um modo virtualmente contínuo. Em adição ao anterior, as implementações da presente invenção podem também ser descritas em termos de fluxogramas de métodos tendo uma seqüência de ações para alcançar os resultados particulares. Em particular, a Figura 2 ilustra fluxogramas a partir da perspectiva do servidor de produção 105 e do servidor de cópia de segurança 110 de acordo com as implementações da presente
25 invenção para copiar e recuperar os dados. As ações ilustradas na Figura 2 são descritas abaixo com relação aos componentes e diagramas ilustrados nas Figuras 1A-1B.

Como uma questão preliminar, referência é às vezes feita aqui a um “primeiro”, “segundo” ou “terceiro” evento (ou uma “ocorrência”) em uma seqüência de tempo. Um apreciá,
30 rá, entretanto, que tais designações são meramente para diferenciar ocorrências exclusivas em uma seqüência contínua, tal que o “primeiro” evento ou ocorrência é não somente diferente de uma “segunda” ou “terceira” ocorrência, mas também ocorre em algum ponto antes da “segunda” e/ou “terceira” ocorrências. Por exemplo, a criação e o envio de uma cópia “completa” de linha de base de dados (por exemplo, 145) poderiam também ser considera-
35 dos uma segunda ou terceira (ou mais) ocorrência com relação a algum evento anterior (não mostrado), mesmo que descrito primariamente como uma primeira ocorrência com relação à atualização 150. Similarmente, a atualização 150 poderia ser descrita como uma ocorrência

em um “primeiro” instante no tempo com relação à atualização 155 (por exemplo, uma “segunda” ocorrência) e assim por diante, apesar desses termos serem primariamente descritos aqui como “segundo” e “terceiro” instantes de tempo nos exemplos ilustrados com relação à linha de base completa 145. Tal uso de termo no sentido relativo para eventos sequenciais também se aplica ao uso aqui dos termos “inicial” ou “subseqüente”.

Por exemplo, a Figura 2 mostra que um método a partir da perspectiva do servidor de produção 105 para replicar os dados do servidor de produção de um modo consistente de aplicativo (isto é, ou de sistema de arquivos), virtualmente contínuo, tal que os dados recentes possam ser facilmente recuperados a partir do servidor de cópia de segurança, compreende uma ação 200 de enviar uma cópia consistente de dados de volume a um servidor de cópia de segurança. A ação 200 inclui enviar uma cópia de dados de volume a partir de um servidor de produção a um servidor de cópia de segurança, onde os dados são consistentes para um primeiro instante de tempo. Por exemplo, o servidor de produção 105 (por exemplo, em resposta ao agente de réplica ou outro comando a partir do servidor de cópia de segurança 110) cria uma cópia de segurança do volume inteiro 145 que é consistente para um ponto no tempo particular (por exemplo, “ t_0 ”). Em uma implementação, isso envolve chamar todos os escritores de aplicativo no servidor de produção 105 para congelar e iniciar preparações para cópia de segurança. O servidor de produção 105 então copia todos os dados no volume (ou em arquivos, pastas ou tipos de arquivos específicos, etc.), e prepara esses dados para serem enviados (e armazenados no) ao servidor de cópia de segurança 110.

Similarmente, a Figura 2 mostra que o método a partir da perspectiva do servidor de cópia de segurança 110 de replicar os dados do servidor de produção de um modo consistente e virtualmente contínuo, tal que os dados recentes possam ser facilmente recuperados a partir do servidor de cópia de segurança, compreende uma ação 240 de receber uma cópia de segurança de volume consistente a partir de um servidor de produção. A ação 240 inclui receber cópias de segurança de um ou mais volumes a partir de um servidor de produção, onde as cópias de segurança de um ou mais volumes são consistentes (isto é, consistente de aplicativo ou de sistema de arquivos) para um instante inicial de tempo. Por exemplo, a Figura 1A mostra que o servidor de cópia de segurança 110 recebe e armazena a cópia de segurança completa 145, que é recebida por uma rede a partir do servidor de produção 105, ou recebida a partir de uma unidade de fita ou outro nó de armazenamento que é em algum ponto conectado com o servidor de cópia de segurança 110. Em particular, um administrador de cópia de segurança pode obter um instantâneo de hardware do servidor de produção 105, e então conectar o instantâneo ao servidor de cópia de segurança 110.

Em adição, a Figura 2 mostra que o método a partir da perspectiva do servidor de produção 105 compreende uma ação 210 de identificar uma ou mais mudanças nos dados

de volume. A ação 210 inclui identificar uma ou mais mudanças nos dados de volume via um ou mais arquivos de registro de volume. Por exemplo, como mostrado nas Figuras 1A e 1B, o acionador de filtro de volume 115 pode rastrear mudanças nos arquivos 120, 125 e 130 e armazenar aquelas mudanças no arquivo de registro de volume 135. Em uma implementação, essas uma ou mais mudanças podem alternativamente ser armazenadas na memória 170 como mapas de bits em memória sendo substituídos no arquivo de registro de volume 135.

A Figura 2 também mostra que o método a partir da perspectiva do servidor de produção 105 compreende uma ação 220 de salvar uma ou mais atualizações consistentes em disco. A ação 220 inclui, mediante identificar arquivos de registro de volume, onde uma ou mais mudanças de dados são consistentes para um segundo instante de tempo. Por exemplo, mediante identificar um evento de ciclo de réplica, salvar uma ou mais mudanças de dados em um ou mais arquivos de registro de volume, onde uma ou mais mudanças de dados são consistentes para um segundo instante de tempo. Por exemplo, mediante identificar um disparador de ciclo de réplica (por exemplo, de um agente de réplica), o acionador de filtro de volume passa os mapas de bits 193, 195 na alocação de memória 190a para o espaço em disco físico alocado 190b (por exemplo, também correspondendo com o arquivo de registro de volume 135). Essa alocação de disco físico 190b assim compreende um instantâneo para um ponto no tempo que é diferente do instantâneo 187 armazenado na alocação em disco 185b (isto é, “todos os mapas de bits” 187).

Além disso, a Figura 2 mostra que o método a partir da perspectiva do servidor de produção 105 compreende uma ação 230 de enviar uma cópia das atualizações consistentes ao servidor de cópia de segurança durante a réplica. A ação 230 inclui enviar ao servidor de cópia de segurança uma cópia de uma ou mais mudanças de dados, tal que o servidor de cópia de segurança tem uma cópia dos dados que são válidos para um primeiro instante de tempo e um segundo instante de tempo. Por exemplo, em adição a passar a atualização completa 145 ao servidor de cópia de segurança 110, o servidor de produção também envia atualizações de instantâneo consistentes 150 e 155, que são válidas para diferentes pontos no tempo (isto é, “ t_1 ”, “ t_2 ”, etc.).

Conseqüentemente, a Figura 2 mostra que o método a partir da perspectiva do servidor de cópia de segurança 110 compreende uma ação 250 de receber uma ou mais atualizações consistentes. A ação 250 inclui receber uma ou mais atualizações de cópia de segurança consistentes, pelo menos uma das quais é uma atualização consistente a pelo menos uma das uma ou mais cópia de segurança do volume para um instante subsequente de tempo. Por exemplo, o servidor de cópia de segurança 110 recebe qualquer das atualizações de instantâneo consistentes 150, 155, tal que o servidor de cópia de segurança agora tem cópias de dados para vários pontos incrementais no tempo (isto é, “ t_1 ”, “ t_2 ”, etc.) compa-

rado à cópia de segurança completa 145 (“ t_0 ”, ou alguma outra linha de base completa).

Em adição, a Figura 2 mostra que o método a partir da perspectiva do servidor de cópia de segurança 110 compreende uma ação 260 de receber uma solicitação de recuperação. A ação 260 inclui receber uma solicitação de recuperação para dados que são válidos de acordo com o instante subsequente de tempo. Por exemplo, o servidor de cópia de segurança 110 recebe uma solicitação (não mostrada) por um arquivo particular que é válido para o tempo “ t_1 ”, que é um arquivo encontrado em ambas a cópia de segurança completa de linha de base 145 e a atualização 150. Além disso, a Figura 2 mostra que o método a partir da perspectiva do servidor de cópia de segurança 110 compreende uma ação 270 de identificar os dados válidos solicitados para o subsequente instante de tempo. A ação 270 inclui identificar os dados solicitados para o subsequente instante de tempo em um ou mais volumes do servidor de cópia de segurança, onde os dados solicitados incluem pelo menos uma parte de pelo menos uma atualização de cópia de segurança consistente.

Com relação aos dados da base de dados, por exemplo, o servidor de cópia de segurança 110 deve desenrolar juntas cópias do arquivo em cada ponto no tempo anterior e atual que é válido para a solicitação; ou seja, o servidor de cópia de segurança 110 combina cópias do arquivo de tempos “ t_0 ” e “ t_1 ”. Por outro lado com os dados do sistema de arquivos, cada subsequente atualização (por exemplo, 150, 155) pode incluir uma cópia atualizada completa do arquivo solicitado, e, como tal, o servidor de cópia de segurança 110 pode somente necessitar identificar os dados solicitados na atualização mais recente (ou alguma outra atualização subsequente) para o ponto no tempo solicitado. Assim, dependendo do tipo de dados solicitados, o servidor de cópia de segurança 110 pode necessitar identificar cada atualização incremental a partir do ponto solicitado voltando à última linha de base completa, ou pode necessitar simplesmente identificar os dados solicitados na última atualização de ponto no tempo.

Como tal, a Figura 2 também mostra que o método a partir da perspectiva do servidor de cópia de segurança 110 compreende uma ação 280 de retornar os dados solicitados. A ação 280 inclui enviar os dados solicitados que são válidos para o instante subsequente de tempo ao servidor de produção. Por exemplo, com o caso dos dados do banco de dados, o servidor de cópia de segurança 110 fornece a recuperação 160, que inclui ambos os dados completos da linha de base (por exemplo, 145) bem como os dados de atualização incremental (por exemplo, 150, 155), enquanto com os dados do sistema de arquivos, o servidor de cópia de segurança 110 fornece uma recuperação que tem pelo menos os dados de arquivo a partir da atualização solicitada de ponto no tempo. A resposta 160 do servidor de cópia de segurança 110 pode também fornecer uma indicação para qual tempo de recuperação a resposta é válida. Como mostrado na Figura 1A, os dados de recuperação 160 mostram que os dados são válidos para o tempo “ t_1 ”.

Conseqüentemente, os diagramas, componentes, e métodos de acordo com as implementações da presente invenção fornecem um número de vantagens sobre os sistemas de cópia de segurança convencionais. Em particular, as implementações da presente invenção fornecem formas de copiar os dados que são agnósticos da fonte de dados (por exemplo, via o acionador de filtro de volume 115), não necessariamente exigem uma versão do aplicativo executando em um servidor de cópia de segurança, e representam réplica virtualmente contínua. Como previamente descrito, essas otimizações são baseadas pelo menos em parte em rastrear ou monitorar mudanças com baixa sobrecarga em um nível de volume usando um acionador de filtro de volume (por exemplo, 115), e também executar ciclos de réplica no nível de arquivo usando, por exemplo, um registro diário USN, tal que as inclusões/exclusões em nível de arquivo possam ainda ser aplicadas.

Em adição, as implementações da presente invenção fornecem a criação de cópias de linha de base completa dos dados do servidor de produção em um servidor de cópia de segurança de uma maneira otimizada, e sem necessariamente exigir a transferência dos dados completos através de uma rede. Minimizar a quantidade de dados subseqüentes que são transferidos a um servidor de cópia de segurança pode fornecer uma redução substancial nos drenos de recursos potenciais na rede e nos servidores de produção. Como tal, as implementações da presente invenção adicionalmente fornecem um número de formas alternativas de alcançar os objetivos de tempo de recuperação rigorosos. As implementações da presente invenção também fornecem um número de formas para rastrear mudanças de dados (por exemplo, em nível de byte ou bloco de bytes) com sobrecarga de baixo desempenho, e rastrear mudanças de dados de uma maneira que é independente do sistema de arquivos e instantâneos de hardware/software. Além disso, as implementações da presente invenção também fornecem uma ou mais formas de reconstruir informação de caminho (tal como com réplica baseada em USN) sem necessariamente exigir estado persistente nos servidores de produção relevantes.

As modalidades da presente invenção podem compreender um computador de propósito especial ou de propósito geral incluindo vários hardwares de computador, como discutido mais detalhadamente abaixo. As modalidades dentro do escopo da presente invenção também incluem meios legíveis por computador para executar ou ter instruções executáveis por computador ou estruturas de dados armazenadas nestes. Tais meios legíveis por computador podem ser quaisquer meios disponíveis que podem ser acessados por um computador de propósito geral ou de propósito específico.

A título de exemplo, e não limitação, tais meios legíveis por computador podem compreender RAM, ROM, EEPROM, CD-ROM ou outro armazenamento de disco óptico, armazenamento de disco magnético ou outros dispositivos de armazenamento magnético, ou qualquer outro meio que pode ser usado para executar ou armazenar código de progra-

ma desejado na forma de instruções executáveis por computador ou estruturas de dados e que pode ser acessado por um computador de propósito geral ou de propósito específico. Quando informação é transferida ou fornecida por uma rede ou outra conexão de comunicações (ou por fio, sem fio, ou uma combinação das duas) a um computador, o computador vê
5 apropriadamente a conexão como um meio legível por computador. Assim, qualquer tal conexão é apropriadamente chamada meio legível por computador. As combinações do acima deveriam também ser incluídas no escopo dos meios legíveis por computador.

As instruções executáveis por computador compreendem, por exemplo, instruções e dados que levam um computador de propósito geral, computador de propósito específico,
10 ou dispositivo de processamento de propósito específico a executar uma certa função ou grupo de funções. Embora o assunto em questão tenha sido descrito em linguagem específica a características estruturais e/ou ações metodológicas, entende-se que o assunto em questão definido nas reivindicações em anexo não está necessariamente limitado às características ou ações específicas descritas acima. De preferência, as características e ações
15 específicas descritas acima são representadas como formas exemplificadas de implementar as reivindicações.

A presente invenção pode ser incorporada em outras formas específicas sem abandonar seu espírito e características essenciais. As modalidades descritas são para serem consideradas em todos os aspectos somente como ilustrativas e não restritivas. O escopo da invenção é, portanto, indicado pelas reivindicações em anexo ao invés de pela descrição anterior. Todas as mudanças que vêm no significado e faixa de equivalência das reivindicações são abrangidas em seu escopo.
20

REIVINDICAÇÕES

1. Em um servidor de produção (105) em um ambiente computadorizado (100) no qual um ou mais servidores de produção copiam dados a serem protegidos em um ou mais volumes (175) em um ou mais servidores de cópia de segurança (110), método para replicar os dados do servidor de produção em um modo consistente virtualmente contínuo, tal que os dados recentes possam ser facilmente recuperados a partir do servidor de cópia de segurança, **CARACTERIZADO** pelo fato de que compreende as ações de:

enviar (200) uma cópia de dados para um ou mais volumes de um servidor de produção (105) a um servidor de cópia de segurança (110), onde os dados são consistentes para um primeiro instante de tempo (145);

identificar (210) uma ou mais mudanças (121, 122, 123) nos dados para um ou mais volumes (175) em um ou mais arquivos de registro de volume (135);

mediante identificar um evento de ciclo de réplica, salvar (220) uma ou mais mudanças de dados em um ou mais arquivos de registro de volume, onde uma ou mais mudanças de dados são consistentes para um segundo instante de tempo (150); e

enviar (230) ao servidor de cópia de segurança uma cópia de uma ou mais mudanças de dados para um ou mais volumes, tal que o servidor de cópia de segurança tem uma cópia de dados para um ou mais volumes que são válidos para um primeiro instante de tempo (145) e um segundo instante de tempo (150).

2. Método, de acordo com a reivindicação 1, **CARACTERIZADO** adicionalmente pelo fato de que compreende salvar mudanças de dados em nível de arquivo para o volume em um de um filtro de mudanças, um registro diário de mudanças, ou um registro diário USN (140).

3. Método, de acordo com a reivindicação 2, **CARACTERIZADO** adicionalmente pelo fato de que compreende uma ação de correlacionar um ou mais arquivos de registro de volume (135) com um do filtro de mudanças, registro diário de mudanças, ou registro diário USN (140) para identificar um ou mais arquivos alterados (120) que correspondem a uma ou mais mudanças de dados em cada arquivo alterado.

4. Método, de acordo com a reivindicação 1, **CARACTERIZADO** adicionalmente pelo fato de que compreende uma ação de fazer uma ou mais mudanças de dados (121, 122, 123) em qualquer um de um nível de byte ou um nível de bloco de bytes nos um ou mais arquivos de registro de volume.

5. Método, de acordo com a reivindicação 1, **CARACTERIZADO** adicionalmente pelo fato de que compreende uma ação de identificar as mudanças nos dados para um ou mais volumes via um ou mais mapas de bits em memória (117a), onde um ou mais mapas de bits em memória correspondem ao segundo instante no tempo (150).

6. Método, de acordo com a reivindicação 5, **CARACTERIZADO** adicionalmente

pelo fato de que compreende uma ação de salvar um ou mais mapas de bits em memória (187) nos um ou mais arquivos de registro de volume durante o evento de ciclo de réplica, onde um ou mais arquivos de registro de volume são armazenados em disco (117b, 185b).

7. Método, de acordo com a reivindicação 5, **CHARACTERIZADO** adicionalmente pelo fato de que compreende as ações de:

congelar um ou mais mapas de bits em memória correspondendo ao segundo instante de tempo; e

criar um novo conjunto de um ou mais mapas de bits em memória (193, 195) correspondendo a novas escritas em um ou mais arquivos alterados para o terceiro instante de tempo (155).

8. Método, de acordo com a reivindicação 1, **CHARACTERIZADO** pelo fato de que um acionador de filtro de volume (115) recebe uma ou mais mudanças de dados e as aplica a um ou mais arquivos de registro de volume (135).

9. Método, de acordo com a reivindicação 1, **CHARACTERIZADO** pelo fato de que uma ou mais mudanças de dados que são consistentes para o primeiro e o segundo instante de tempo são pelo menos uma dentre consistente de aplicativo ou consistente de sistema de arquivos.

10. Método, de acordo com a reivindicação 1, **CHARACTERIZADO** adicionalmente pelo fato de que compreende uma ação de mudar um caminho de arquivo de um arquivo (20) correspondendo a qualquer das uma ou mais mudanças de dados (121, 122, 123) no servidor de produção, tal que o caminho de arquivo no servidor de produção (105) é diferente de um caminho para o arquivo no servidor de cópia de segurança (110).

11. Método, de acordo com a reivindicação 10, **CHARACTERIZADO** adicionalmente pelo fato de que compreende uma etapa para correlacionar os caminhos para o arquivo (120) no servidor de produção (105) e no servidor de cópia de segurança (110), tal que novas mudanças no arquivo possam ser enviadas ao servidor de cópia de segurança com uma mudança no caminho para o arquivo.

12. Método, de acordo com a reivindicação 11, **CHARACTERIZADO** pelo fato de que a etapa para correlacionar os caminhos de arquivos para o arquivo (120) compreende as ações de:

identificar um caminho inicial para o arquivo (120); e

armazenar o caminho de arquivo inicial para o arquivo no servidor de produção (105).

13. Método, de acordo com a reivindicação 11, **CHARACTERIZADO** pelo fato de que a etapa para correlacionar os caminhos de arquivo para o arquivo (120) compreende as ações de:

varrer um registro diário USN (140) pelo menos uma primeira vez para armazenar

em cache a mudança no caminho de arquivo no servidor de produção;

varrer um registro diário USN pelo menos uma segunda vez para identificar o caminho de arquivo inicial no servidor de produção (105); e

computar um caminho ajustado para o arquivo no servidor de cópia de segurança (110) baseado na primeira e na segunda varredura.

14. Método, de acordo com a reivindicação 1, **CHARACTERIZADO** adicionalmente pelo fato de que compreende a ação de enviar uma nova atualização de dados de volume no servidor de produção ao servidor de cópia de segurança, onde a nova atualização é consistente para um terceiro instante de tempo (155), e onde o tempo decorrido entre o segundo (150) e o terceiro (155) instantes de tempo é configurável para qualquer um de menos de aproximadamente uma hora, ou menos do que aproximadamente meia hora.

15. Método, de acordo com a reivindicação 14, **CHARACTERIZADO** adicionalmente pelo fato de que compreende uma ação de enviar uma solicitação ao servidor de cópia de segurança (110) por uma cópia de um ou mais arquivos, onde a solicitação ao servidor de cópia de segurança por uma cópia de um ou mais arquivos, onde a solicitação inclui uma indicação de que um ou mais arquivos são válidos para um do segundo (150) ou do terceiro (155) instante de tempo.

16. Método, de acordo com a reivindicação 15, **CHARACTERIZADO** adicionalmente pelo fato de que compreende uma ação de receber uma resposta de recuperação (160) a partir de um servidor de cópia de segurança (110), onde a resposta de recuperação inclui uma cópia completa de dados para os um ou mais arquivos selecionados como do segundo ou do terceiro instante de tempo.

17. Em um servidor de cópia de segurança (110) em um ambiente computadorizado (100) no qual um ou mais servidores de cópia de segurança copiam dados de um ou mais volumes de um ou mais servidores de produção (105), método de replicar dados de um ou mais volumes do servidor de produção (175) de um modo consistente e virtualmente contínuo, tal que os dados recentes podem ser facilmente recuperados a partir do servidor de cópia de segurança, **CHARACTERIZADO** adicionalmente pelo fato de que compreende as ações de:

receber (240) uma ou mais cópias de segurança de volume a partir de um servidor de produção (105), onde uma ou mais cópias de segurança de volume são consistentes para um instante inicial de tempo (145);

receber (250) uma ou mais atualizações de cópia de segurança consistentes, pelo menos uma das quais é uma atualização consistente de aplicativo para pelo menos uma ou mais cópias de segurança de volume para um instante subsequente de tempo (150, 155);

receber (260) uma solicitação de recuperação para dados que são válidos de acordo com o instante subsequente de tempo;

identificar (270) os dados solicitados para o instante subsequente de tempo em um ou mais volumes de servidor de cópia de segurança, onde os dados solicitados incluem pelo menos uma parte de pelo menos uma atualização de cópia de segurança consistente; e

5 enviar (280) os dados solicitados (160) que são válidos para o instante subsequente de tempo ao servidor de produção.

18. Método, de acordo com a reivindicação 16, **CARACTERIZADO** adicionalmente pelo fato de que compreende as ações de:

identificar uma cópia completa de linha de base (145) dos dados solicitados;

10 identificar cada uma das uma ou mais atualizações de cópia de segurança consistentes (150, 155) para os dados solicitados que foram recebidos entre o recebimento da cópia completa de linha de base e o recebimento de pelo menos uma atualização de cópia de segurança para o instante subsequente de tempo; e

15 combinar a cópia completa de linha de base dos dados solicitados com as uma ou mais atualizações de cópia de segurança consistentes identificadas para os dados solicitados.

19. Método, de acordo com a reivindicação 18, **CARACTERIZADO** pelo fato de que uma ou mais atualizações de volume e uma ou mais atualizações de cópia de segurança consistentes são pelo menos um dentre consistente de aplicativo ou consistente de sistema de arquivos.

20 20. Em um servidor de produção (105) em um ambiente computadorizado (100) no qual um ou mais servidores de produção copiam dados a serem protegidos em um ou mais servidores de cópia de segurança (110), um produto de programa de computador tendo instruções executáveis por computador armazenadas nele que, quando executadas, levam um ou mais processadores no servidor de produção a executar um método de replicar os dados do servidor de produção em um modo consistente de aplicativo virtualmente contínuo, tal que os dados recentes possam ser facilmente recuperados a partir do servidor de cópia de segurança, **CARACTERIZADO** pelo fato de que compreende as ações de:

30 enviar (200) uma cópia de dados para um ou mais volumes a partir de um servidor de produção (105) a um servidor de cópia de segurança (110), onde os dados são consistentes para um primeiro instante de tempo (145);

identificar (210) uma ou mais mudanças (121, 122, 123) nos dados para um ou mais volumes (175) em um ou mais arquivos de registro de volume (135);

35 mediante identificar um evento de ciclo de réplica, salvar (220) uma ou mais mudanças de dados em um ou mais arquivos de registro de volume, onde uma ou mais mudanças de dados são consistentes para um segundo instante de tempo (150); e

enviar (230) ao servidor de cópia de segurança uma cópia de uma ou mais mudanças de dados para um ou mais volumes, tal que o servidor de cópia de segurança tem uma

cópia de dados para um ou mais volumes que são válidos para um primeiro instante de tempo (145) e um segundo instante de tempo (150).

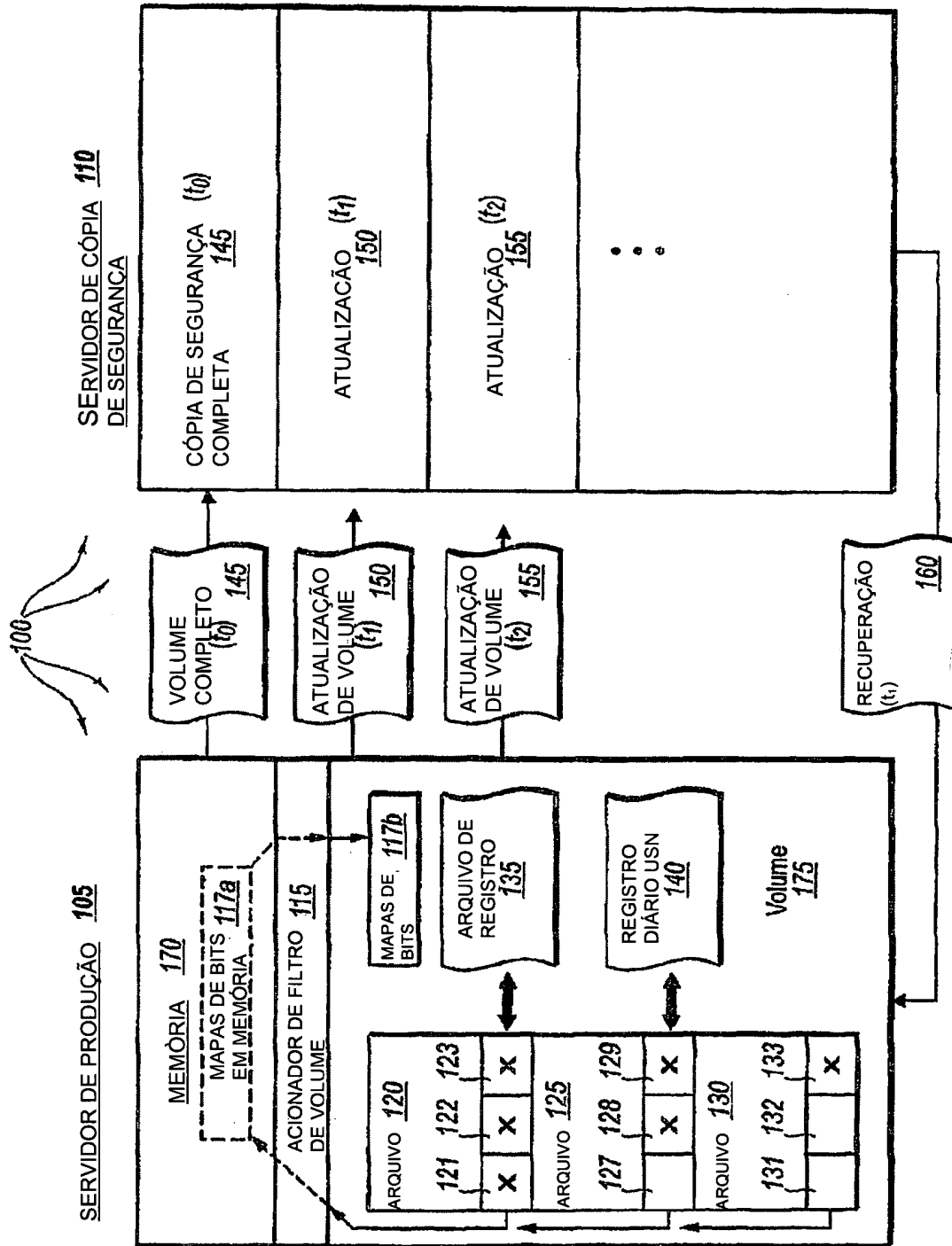


FIG. 1A

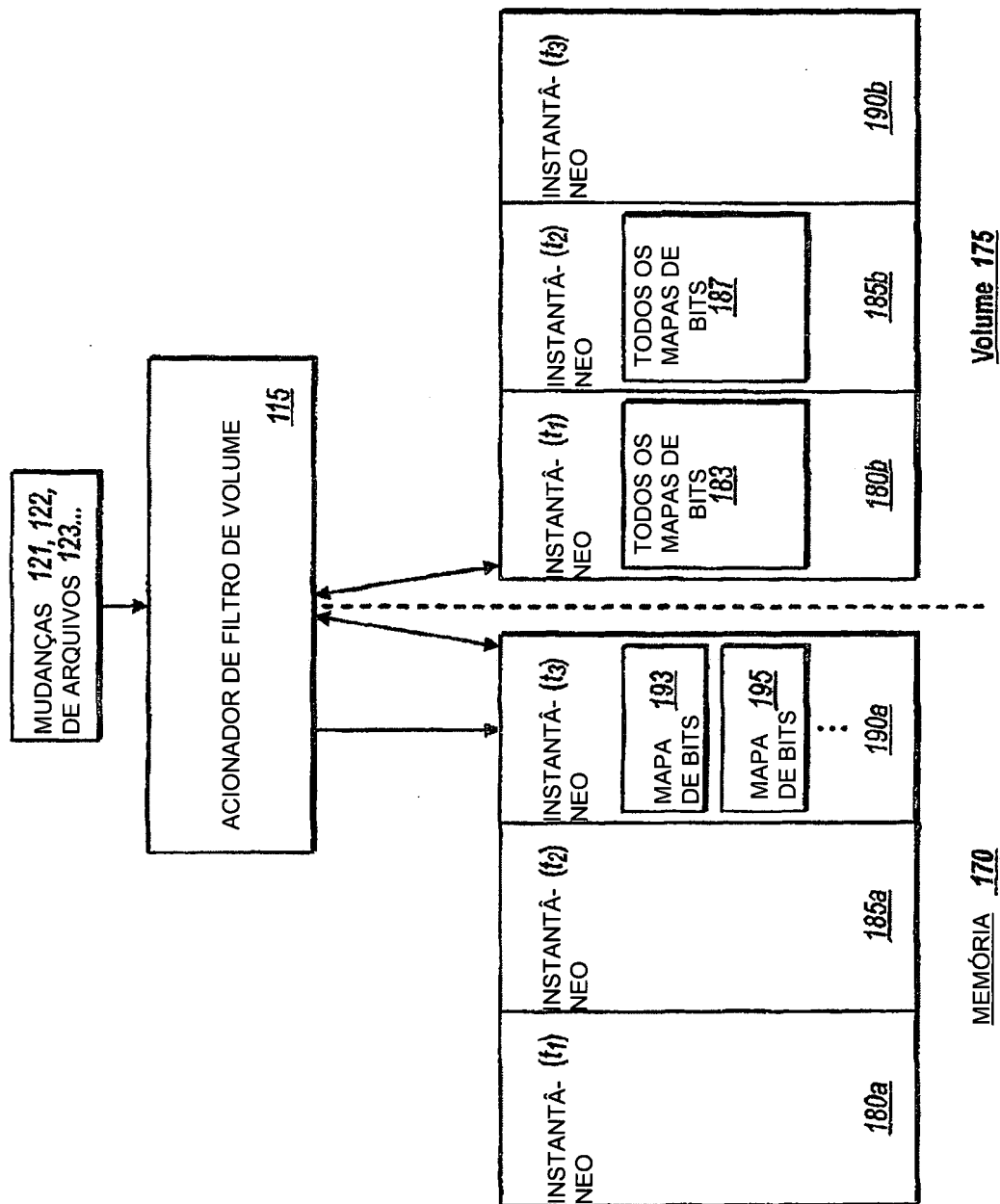
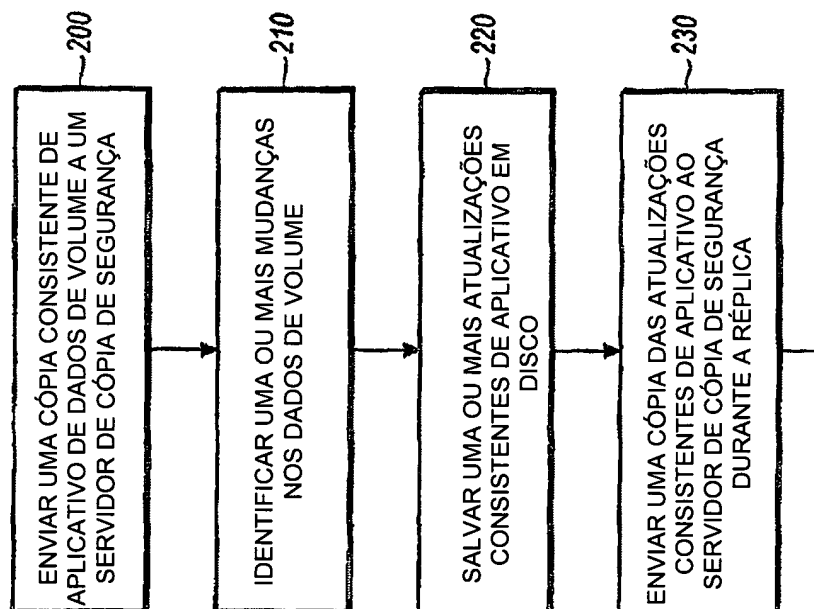


FIG. 1B

SERVIDOR DE PRODUÇÃO 105



SERVIDOR DE CÓPIA 110 DE SEGURANÇA

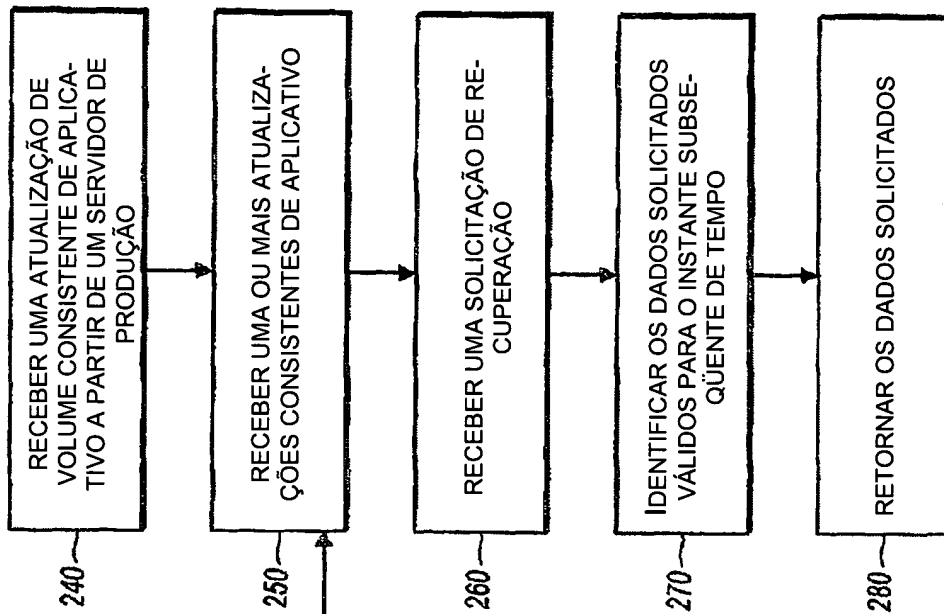


FIG. 2

RESUMO**"CRIANDO EFICIENTEMENTE FREQUENTES CÓPIAS DE SEGURANÇA CONSISTENTES DE APLICATIVO"**

Os dados podem ser protegidos em um servidor de produção de um modo virtualmente contínuo, sem necessariamente impor severas restrições no aplicativo(s) fonte. Por exemplo, um servidor de produção pode criar uma cópia de segurança consistente de aplicativo de um ou mais volumes, as cópias de segurança correspondentes a um primeiro instante no tempo. Um acionador de filtro de volume pode monitorar mudanças de dados usando um mapa de bits em memória, enquanto um arquivo de registro e/ou registro diário de número de sequência de atualização pode manter rastreamento de arquivos que foram modificados. As atualizações de volume são também consistentes para um instante (no futuro) no tempo. No próximo ciclo de cópia, tal como a cada poucos minutos (entretanto configurados), o acionador de filtro de volume passa cada mapa de bits em memória ao disco físico no servidor de produção. O servidor de produção então envia as atualizações ao servidor de cópia de segurança, que assim armazena cópias de segurança consistentes de aplicativo para o volume para múltiplos instantes de tempo.