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

(21) **PI0708199-5 A2**

BRPI0708199A2

(22) Data de Depósito: 13/02/2007
(43) Data da Publicação: 17/05/2011
(RPI 2106)

(51) Int.Cl.:
G06F 15/16
G06F 17/40
G06F 17/00

(54) Título: **ARMAZENAMENTO PONTO A PONTO CONFIÁVEL E EFICIENTE**

(30) Prioridade Unionista: 22/02/2006 US 11/359.276

(73) Titular(es): Microsoft Corporation

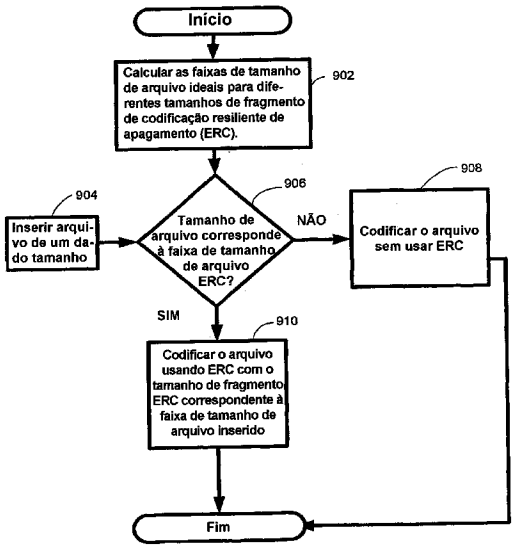
(72) Inventor(es): Jin Li

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

(86) Pedido Internacional: PCT US2007004048 de 13/02/2007

(87) Publicação Internacional: WO 2007/100509de 07/09/2007

(57) Resumo: ARMAZENAMENTO PONTO A PONTO CONFIÁVEL E EFICIENTE E divulgado um sistema de armazenamento com codificação adaptativa que usa codificação resiliente de apagamento (ERC) adaptativa que muda o número de fragmentos usados para codificação de acordo com o tamanho do arquivo distribuído. ERC adaptativa pode melhorar enormemente a eficiência e a confiabilidade do armazenamento P2P. Inúme ros procedimentos para aplicações de armazenamento P2P também podem ser impíementados. Em uma modalidade, pequenos arquivos de dados dinâmicos são desviados para os pares mais confiáveis ou mesmo para um servidor, enquanto que grandes arquivos estáticos são armazenados utilizando-se a capacidade de armazenamento dos pares não confiáveis. Também, para contribuição e benefício equilibrados, um par deve hospedar a mesma quantidade de conteúdo armazenada na rede P2P. Em decorrência disto, pares não confiáveis podem distribuir menos dados, e pares confiáveis podem distribuir mais. Também, arquivos menores são atribuidos com um custo de distribuição mais alto, e os arquivos maiores são atribuidos com um custo de distribuição mais baixo.



"ARMAZENAMENTO PONTO A PONTO CONFIÁVEL E EFICIENTE"

ANTECEDENTES DA INVENÇÃO

Em uma aplicação Ponto a Ponto (P2P), pares trazem consigo larguras de banda de rede e/ou recursos de armazenamento em disco rígido quando eles ingressam no serviço P2P. À medida que a demanda em um sistema P2P cresce, a capacidade do sistema também cresce. Isto é exatamente o contrário de um sistema cliente-servidor, em que a capacidade do servidor é fixa e paga pelo provedor do sistema cliente-servidor. Em decorrência disto, um sistema P2P é mais econômico de executar do que um sistema cliente-servidor, e é superior em virtude de ser escalável.

Em um sistema P2P, o par contribui não somente com a largura de banda, mas também com o espaço de armazenamento para servir os outros pares. O espaço de armazenamento coletivo contribuído pelos pares forma uma nuvem de armazenamento distribuída. Dados podem ser armazenados na nuvem e ser recuperados a partir dela. O armazenamento P2P pode ser usado por inúmeras aplicações. Uma é a cópia de segurança distribuída. O par pode fazer cópia de segurança dos seus próprios dados na nuvem P2P. Quando um par falhar, os dados podem ser restaurados a partir da nuvem. Uma outra aplicação P2P é acesso distribuído a dados. Em virtude de o cliente poder recuperar dados simultaneamente a partir de múltiplos pares controladores, a recuperação P2P pode ter maior rendimento se comparado com a recuperação de dados a partir de uma única fonte. Uma outra aplicação é visualização de filme sob demanda. Um servidor de mídia pode disseminar a nuvem P2P com arquivos de filme com direito de posse. Quando um cliente estiver visualizando o filme, ele pode transmitir o filme em fluxo contínuo tanto a partir da nuvem P2P quanto a partir do servidor, assim, reduzindo a carga do servidor, reduzindo o tráfego na espinha dorsal da rede e melhorando a qualidade do filme em fluxo contínuo.

Embora os pares na rede P2P possam agir como servidores, eles diferem de servidores da Internet / bases de dados comerciais em um importante aspecto: confiabilidade. Em virtude de, usualmente, um par ser um computador ordinário que suporta a aplicação P2P com seu espaço livre em disco rígido e largura de banda ociosa, ele é muito menos confiável do que um servidor típico. O usuário pode escolher desligar um computador par ou a aplicação P2P de tempos em tempos. Necessidades compulsórias, por exemplo, transferência / carregamento de grandes arquivos, podem subalimentar o par em relação à largura de banda necessária para a atividade P2P. O computador par pode estar fora de linha em função da necessidade de atualizar ou consertar software / hardware ou em função de um ataque de vírus. O hardware do computador e a ligação de rede do par também são inerentemente muito menos confiáveis do que um computador servidor típico e suas ligações de rede comerciais, que são projetados para confiabilidade. Embora agrupamentos comerciais servidor / servidor sejam projetados para confiabilidade "seis nove" (com uma taxa de falha

de 10^{-6}, nesta taxa, cerca de 30 segundos de tempo ocioso é permitido a cada ano), um bom par cliente pode ter confiabilidade somente "dois nove" (uma taxa de falha de 10^{-2} ou cerca de 15 minutos de tempo ocioso a cada dia), e não é incomum que pares tenham somente 50 % (metade do tempo ocioso) ou até 10 % de confiabilidade (ocioso em 90 % do tempo).

A maior parte das aplicações P2P, por exemplo, cópia de segurança e recuperação de dados P2P, deseja manter o mesmo nível de confiabilidade para armazenamento P2P do que aquele do servidor (confiabilidade "seis nove"). O desafio está em como construir um armazenamento P2P confiável e eficiente usando mínima largura de banda e recursos de armazenamento dos pares.

SUMÁRIO DA INVENÇÃO

São apresentados um sistema e método de armazenamento com codificação adaptativa para armazenar dados de forma eficiente e confiável em uma rede Ponto a Ponto (P2P). Os sistema e método de armazenamento com codificação adaptativa ajustam inúmeros fragmentos para codificação resiliente de apagamento (ERC), o número de fragmentos ERC, com base no tamanho do arquivo armazenado e distribuído.

Inúmeras modalidades do sistema de armazenamento com codificação adaptativa empregam procedimentos para melhorar a eficiência e confiabilidade de uma rede P2P. Por exemplo, em uma modalidade, pequenos dados dinâmicos são desviados para pares mais confiáveis ou mesmo para um servidor, se o suporte ao componente do servidor estiver disponível. Também, em uma outra modalidade, para uma rede P2P equilibrada, é permitido que pares que não são confiáveis e estão distribuindo arquivos menores distribuam menos dados.

Percebe-se que, embora as limitações expostas nos sistemas de armazenamento e de distribuição ponto a ponto existentes descritos na seção de Antecedentes da Invenção possam ser resolvidas por uma implementação em particular do sistema de armazenamento com codificação adaptativa de acordo com a presente invenção, estes sistema e processo não são, de nenhuma maneira, limitados às implementações que resolvem exatamente qualquer uma e todas as desvantagens expostas. Em vez disto, os presentes sistema e processo têm uma aplicação muito mais ampla, como ficará evidente a partir das descrições que seguem.

Também percebe-se que este Sumário é fornecido para introduzir uma seleção de conceitos de uma forma simplificada que são descritos adicionalmente a seguir na Descrição Detalhada. Não pretende-se que este Sumário identifique recursos chaves ou recursos essenciais do assunto em questão reivindicado, nem pretende-se que seja usado como um auxílio na determinação do escopo do assunto em questão reivindicado.

DESCRIÇÃO RESUMIDA DOS DESENHOS

Os recursos, aspectos e vantagens específicos do sistema de armazenamento com

codificação adaptativa serão mais bem entendidos em relação à descrição, reivindicações anexas e desenhos anexos seguintes, em que:

a figura 1 é um diagrama de sistema geral que representa um dispositivo de computação de uso geral que constitui um sistema exemplar que implementa um sistema e método de armazenamento com codificação adaptativa aqui descrito.

A figura 2 ilustra uma rede ponto a ponto (P2P) exemplar que pode ser usada com os sistema e método de armazenamento com codificação adaptativa aqui descritos.

A figura 3 fornece um gráfico que mostra o número de pares de armazenamento de informação para alcançar uma confiabilidade desejada de 10^{-6}.

A figura 4 fornece um gráfico que mostra a confiabilidade do par e a taxa de replicação desejadas.

A figura 5 fornece um gráfico que mostra o número de pares de armazenamento de informação necessário para alcançar confiabilidade desejada de 10^{-6} usando codificação resiliente de apagamento.

A figura 6 fornece um gráfico do número de fragmentos ERC e o tamanho de arquivo adequado associado para armazenamento de informação em uma rede P2P.

A figura 7 fornece um gráfico que representa uso de largura de banda entre pares em uma configuração P2P com ERC adaptativa e ERC fixa (em uma confiabilidade de par = 50 %). A figura 8 fornece um gráfico que representa uso de largura de banda entre pares em uma configuração P2P com ERC adaptativa e ERC fixa (em uma confiabilidade de par = 99 %).

A figura 9 representa uma modalidade do processo de armazenamento com codificação adaptativa.

A figura 10 representa um fluxograma operacional exemplar que mostra como a técnica de armazenamento com codificação adaptativa é empregada em uma rede P2P.

A figura 11 representa uma modalidade dos sistema e processo de armazenamento com codificação adaptativa que implementam um procedimento para otimizar a eficiência de armazenamento de uma rede P2P.

A figura 12 representa uma outra modalidade dos sistema e método de armazenamento com codificação adaptativa que implementam procedimentos para otimizar a eficiência de armazenamento de um sistema P2P.

A figura 13 representa uma modalidade dos sistema e método de armazenamento com codificação adaptativa que empregam cópia de segurança P2P com suporte ao servidor.

DESCRIÇÃO DETALHADA

Na seguinte descrição das modalidades preferidas do presente sistema de armazenamento com codificação adaptativa, a referência é feita aos desenhos anexos que formam

uma parte deste, e que são mostrados a título de ilustração das modalidades específicas nas quais o sistema de armazenamento com codificação adaptativa pode ser praticado. Entende-se que outras modalidades podem ser utilizadas e que mudanças estruturais podem ser feitas sem fugir do escopo do presente sistema de armazenamento com codificação adaptativa.

1.0 AMBIENTE OPERACIONAL EXEMPLAR

A figura 1 ilustra um exemplo de um ambiente do sistema de computação adequado 100 no qual a invenção pode ser implementada. O ambiente do sistema de computação 100 é somente um exemplo de um ambiente de computação adequado e não pretende-se que sugira nenhuma limitação ao escopo do uso ou de funcionalidade da invenção. Nem o ambiente de computação 110 deve ser interpretado sem nenhuma dependência ou exigência relacionada a qualquer componente ou combinação de componentes ilustrados no ambiente operacional exemplar 100.

A invenção é operacional com inúmeros outros ambientes ou configurações de sistema de computação de uso geral ou de uso especial. Exemplos de sistemas, ambientes e/ou configurações de computação bem conhecidos que podem ser adequados para uso com a invenção incluem, mas sem limitações, computadores pessoais, computadores servidores, dispositivos de computação ou de comunicações de mão, portáteis ou móveis, tais como telefones celulares e PDAs, sistemas multiprocessadores, sistemas com base em multiprocessadores, conversores de sinal de freqüência, dispositivos eletrônicos programáveis pelo cliente, PCs em rede, minicomputadores, computadores de grande porte, ambientes de computação distribuída que incluem qualquer um dos sistemas ou dispositivos expostos, e congêneres.

A invenção pode ser descrita no contexto geral das instruções executáveis por computador, tais como módulos de programa que são executados por um computador em conjunto com módulos de hardware, incluindo componentes de um arranjo de microfone 198. No geral, módulos de programa incluem rotinas, programas, objetos, componentes, estruturas de dados, etc., que realizam tarefas em particular ou implementam tipos de dados abstratos em particular. A invenção também pode ser praticada em ambientes de computação distribuída em que tarefas são realizadas por dispositivos de processamento remoto que são ligados por meio de uma rede de comunicações. Em um ambiente de computação distribuída, os módulos de programa podem ficar localizados em mídia de armazenamento tanto local quanto remota, incluindo dispositivos de armazenamento em memória. Em relação à figura 1, um sistema exemplar para implementar a invenção inclui um dispositivo de computação de uso geral na forma de um computador 110.

Componentes do computador 110 podem incluir, mas sem limitações, uma unidade de processamento 120, uma memória do sistema 130 e um barramento do sistema 121 que

acopla vários componentes do sistema, incluindo a memória do sistema, na unidade de processamento 120. O barramento do sistema 121 pode ser qualquer um de diversos tipos de estruturas de barramento, incluindo um barramento de memória ou controlador de memória, um barramento periférico e um barramento local que usa qualquer uma de uma variedade de arquiteturas de barramento. A título de exemplo, e sem limitações, tais arquiteturas incluem barramento Arquitetura Padrão da Indústria (ISA), barramento Arquitetura Micro Canal (MCA), barramento ISA Melhorado (EISA), barramento Associação dos Padrões Eletrônicos de Vídeo (VESA) local, e barramento Interconexão de Componente Periférico (PCI), também conhecido como barramento *Mezzanine*.

Tipicamente, o computador 110 inclui uma variedade de mídias legíveis por computador. A mídia legível por computador pode ser qualquer mídia disponível que pode ser acessada pelo computador 110 e inclui tanto mídia volátil quanto mídia não volátil, tanto mídia removível quanto mídia não removível. A título de exemplo, e sem limitações, a mídia legível por computador pode compreender mídia de armazenamento em computador e mídia de comunicação. A mídia de armazenamento em computador inclui mídia volátil e não volátil, removível e não removível implementada em qualquer método ou tecnologia para armazenamento da informação, tais como instruções legíveis por computador, estrutura de dados, módulos de programa ou outros dados.

Mídia de armazenamento no computador inclui, mas sem limitações, RAM, ROM, PROM, EPROM, EEPROM, memória flash, ou outra tecnologia de memória, CD-ROM, discos versáteis digitais (DVD), ou outro armazenamento em disco ótico, cassetes magnéticos, fita magnética, armazenamento em disco magnético ou outros dispositivos de armazenamento magnético, ou qualquer outra mídia que possa ser usada para armazenar a informação desejada e que pode ser acessada pelo computador 110. Tipicamente, mídia de comunicação incorpora instruções legíveis por computador, estrutura de dados, módulos de programa ou outros dados em um sinal de dados modulado, tais como uma onda portadora ou outro mecanismo de transporte, e inclui qualquer mídia de distribuição de informação. O termo "sinal de dados modulado" significa um sinal que tem uma ou mais de suas características ajustadas ou modificadas de uma maneira tal que seja para codificar informação no sinal. A título de exemplo, e sem limitações, mídia de comunicação inclui mídia com fios, tais como rede com fios ou conexão direta com fios, e mídia sem fios, tais como mídia acústica, RF, infravermelho, e outras mídias sem fios. Combinações de qualquer um dos expostos também devem ser incluídas no escopo da mídia legível por computador.

A memória do sistema 130 inclui mídia de armazenamento no computador na forma de memória volátil e/ou não volátil, tais como memória exclusiva de leitura (ROM) 131 e memória de acesso aleatório (RAM) 132. Um sistema básico de entrada / saída (BIOS) 133 que contém as rotinas básicas que ajudam a transferir informação entre elementos no com-

putador 110, tal como durante a inicialização, é armazenado, tipicamente, na ROM 131. Tipicamente, a RAM 132 contém dados e/ou módulos de programa que são imediatamente acessíveis pela unidade de processamento 120 e/ou que estão sendo atualmente operados por ela. A título de exemplo, e sem limitações, a figura 1 ilustra o sistema operacional 134, programas de aplicação 135, outros módulos de programa 136 e dados do programa 137.

O computador 110 também pode incluir outras mídias de armazenamento no computador removíveis / não removíveis, voláteis / não voláteis. A título de exemplo somente, a figura 1 ilustra uma unidade de disco rígido 141 que lê e grava em uma mídia magnética não removível e não volátil, uma unidade de disco magnético 151 que lê e grava em um disco magnético removível e não volátil 152, e uma unidade de disco ótico 155 que lê e grava em um disco ótico removível e não volátil 156, tais como um CD ROM ou outra mídia ótica. Outras mídias de armazenamento no computador removíveis / não removíveis, voláteis / não voláteis que podem ser usadas no ambiente operacional exemplar incluem, mas sem limitações, cassetes de fita magnética, cartões de memória flash, discos versáteis digitais, fita de vídeo digital, RAM em estado sólido, ROM em estado sólido e congêneres. Tipicamente, a unidade de disco rígido 141 é conectada no barramento do sistema 121 por meio de uma interface de memória não removível, tal como a interface 140, e, tipicamente, a unidade de disco magnético 151 e a unidade de disco ótico 155 são conectadas no barramento do sistema 121 por uma interface de memória removível, tal como a interface 150.

As unidades e suas mídias de armazenamento no computador discutidas anteriormente e ilustradas na figura 1 fornecem armazenamento de instruções legíveis por computador, de estruturas de dados, de módulos de programa e de outros dados para o computador 110. Por exemplo, na figura 1, a unidade de disco rígido 141 é ilustrada armazenando o sistema operacional 144, os programas de aplicação 145, outros módulos de programa 146 e dados de programa 147. Note que estes componentes tanto podem ser os mesmos quanto podem ser diferentes do sistema operacional 134, dos programas de aplicação 135, de outros módulos de programa 136 e dos dados de programa 137. Aqui, são dados ao sistema operacional 144, aos programas de aplicação 145, aos outros módulos de programa 146 e aos dados de programa 147 diferentes números para ilustrar que, no mínimo, eles são cópias diferentes. Um usuário pode inserir comandos e informação no computador 110 por meio de dispositivo de entrada, tais como um teclado 162 e um dispositivo de apontamento 161, comumente chamado de mouse, dispositivo de apontamento com esfera superior ou plataforma sensível ao toque.

Outros dispositivos de entrada (não mostrados) podem incluir uma manete, um controlador de jogos, antena parabólica, digitalizador, receptor de rádio e uma televisão ou receptor de vídeo por difusão ou congêneres. Estes e outros dispositivos de entrada são freqüentemente conectados na unidade de processamento 120 por meio de uma interface de

entrada de usuário com fios ou sem fios 160 que é acoplada no barramento do sistema 121, mas podem ser conectados por outras interfaces e estruturas de barramento convencionais, tais como, por exemplo, uma porta paralela, uma porta de jogos, um barramento serial universal (USB), uma interface IEEE 1394, uma interface sem fios Bluetooth™, uma interface sem fios IEEE 802.11, etc. Adicionalmente, o computador 110 também pode incluir um dispositivo de fala ou de entrada de áudio, tais como um microfone ou um arranjo de microfone 198, bem como um alto-falante 197 ou outro dispositivo de saída de som conectado por meio de uma interface de áudio 199, novamente incluindo interfaces com fios ou sem fios convencionais, tais como, por exemplo, paralela, serial, USB, IEEE 1394, Bluetooth™, etc.

Um monitor 191 ou outro tipo de dispositivo de exibição também é conectado no barramento do sistema 121 por meio de uma interface, tal como uma interface de vídeo 190. Além do monitor, computadores também podem incluir outros dispositivos periféricos de saída, tal como uma impressora 196, que pode ser conectada por meio de uma interface periférica de saída 195.

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

Quando usado em um ambiente de rede LAN, o computador 110 é conectado na LAN 171 por meio de uma interface ou adaptador de rede 170. Quando usado em um ambiente de rede WAN, o computador 110 inclui, tipicamente, um modem 172 ou outro dispositivo para estabelecer comunicações na WAN 173, tal como a Internet. O modem 172, que pode ser interno ou externo, pode ser conectado no barramento do sistema 121 por meio da interface de entrada do usuário 160 ou de outro mecanismo apropriado. Em um ambiente de rede, os módulos de programa representados em relação ao computador 110, ou a partes dele, podem ser armazenados no dispositivo de armazenamento em memória remoto. A título de exemplo, e sem limitações, a figura 1 ilustra programas de aplicação remotos 185 residentes no dispositivo de memória 181. Percebe-se que as conexões de rede mostradas são exemplares e que podem ser usados outros dispositivos para estabelecer uma ligação de comunicações entre os computadores.

No geral, o sistema de armazenamento com codificação adaptativa opera em uma rede P2P, tal como a rede ilustrada pela figura 2. Para uma sessão de transmissão de da-

dos em fluxo contínuo em particular, um "servidor" 200 é definido como um nó na rede P2P que organiza inicialmente os dados ou mídia transmitida em fluxo contínuo, um "cliente" (ou receptor) 210 é definido como um nó que solicita atualmente os dados, e um "par de serviço" 220 é definido como um nó que serve o cliente com uma cópia completa ou parcial dos dados.

No geral, o servidor 200, o cliente 210 e os pares de serviço 220 são todos nós de usuário final conectados em uma rede tal como a Internet. Em virtude de o servidor 200 sempre poder servir os dados, o nó servidor também age como um par de serviço 220. O nó servidor 200 também pode desempenhar funcionalidades administrativas que não podem ser desempenhadas por um par de serviço 220, por exemplo, manter uma lista de pares de serviço disponíveis, desempenhar funcionalidade de gerenciamento de direitos digitais (DRM), e assim por diante. Além do mais, como com os esquemas P2P convencionais, o sistema de armazenamento com codificação adaptativa aqui descrito se beneficia da maior eficiência à medida que mais e mais nós pares 220 são implementados. Em particular, à medida que o número de nós pares 220 aumenta, a carga no servidor de dados 200 diminuirá, desse modo, ficando menos oneroso operar, enquanto que cada nó cliente 210 poderá receber dados com muito mais qualidade durante uma sessão de transferência de dados em particular.

Além do mais, deve ficar claro que o papel dos nós em particular pode mudar. Por exemplo, um nó em particular pode agir como o cliente 210 em uma transferência de dados em particular, agindo como um par de serviço 220 em uma outra sessão. Adicionalmente, nós em particular podem agir simultaneamente tanto como nós clientes 210 quanto como servidores 200 quanto como pares de serviço 220 para transmitir simultaneamente um ou mais arquivos de dados, ou partes destes arquivos, recebendo outros dados de um ou mais outros pares de serviço.

Durante uma transmissão de dados, primeiro, o cliente 200 localiza inúmeros pares próximos 220 que detêm parte dos dados desejados ou todos eles e, então, recebe os dados dos múltiplos pares (que podem incluir o servidor 200). Conseqüentemente, cada par de serviço 220 age para auxiliar o servidor 200 pela redução do encargo total de carregamento servindo uma parte da solicitação de transferência do cliente 210. Em decorrência disto, freqüentemente, o cliente 210, especialmente no caso em que há muitos clientes, pode receber dados com muito mais qualidade, já que há uma largura de banda de serviço significativamente maior disponível quando há muitos pares de serviço 220 para auxiliar o servidor 200.

O ambiente operacional exemplar tendo sido agora discutido, as partes restantes desta seção de descrição serão dedicadas a uma descrição dos módulos de programa que incorporam o sistema e processo de armazenamento com codificação adaptativa.

2.0 ARMAZENAMENTO PONTO A PONTO CONFIÁVEL E EFICIENTE

O sistema de armazenamento com codificação adaptativa fornece um esquema de codificação resiliente de apagamento (ERC) que determina adaptativamente se usa-se codificação ERC ou não e emprega o número ideal de fragmentos a ser usados para codificação ERC para um dado tamanho de arquivo para confiabilidade e eficiência ideais. O número de fragmentos usado para codificação ERC de um arquivo será denominado "número de fragmentos ERC" com os propósitos desta discussão. Os seguintes parágrafos fornecem uma discussão da eficiência e da confiabilidade do armazenamento ponto a ponto (P2P) e do uso do ERC em redes P2P, bem como uma discussão do número de fragmentos ERC usado. Então, várias modalidades do sistema e processo de armazenamento com codificação adaptativa são discutidas.

2.1 Confiabilidade em Armazenamento P2P; Redundância de Dados

A solução *ad hoc* para ocasionar confiabilidade a um sistema com partes não confiáveis é o uso da redundância. Se cada par individual na rede tem uma confiabilidade de p, para alcançar uma confiabilidade desejada de p_0, pode-se simplesmente replicar a invenção a n pares:

$$n = log(1 - p_0) / log(1 - p), \quad (1)$$

em que n é um número de pares que detêm a informação. No momento da recuperação, o cliente pode entrar em contato com os pares que armazenam a informação um a um. Desde que um dos pares que armazenam a informação esteja em linha, a informação pode ser confiavelmente recuperada.

Embora alcance a confiabilidade, a estratégia de simples replicação não é eficiente. A figura 3 esboça o número de pares que armazenam informação necessários para alcançar confiabilidade "seis nove". Com a confiabilidade do par em 50 %, é necessário replicar e armazenar a informação em 20 pares. Isto leva a 20 vezes mais largura de banda e espaço de armazenamento para distribuir e armazenar informação. Obviamente, a eficiência foi sacrificada em troca da confiabilidade de informação.

2.2 Codificação Resiliente de Apagamento em P2P

Para melhorar a eficiência, ainda mantendo a mesma confiabilidade, ERC pode ser uma ferramenta usada. ERC divide o arquivo original em k fragmentos originais $\{x_i\}$, $i = 0, ..., k\text{-}1$, cada um dos quais é um vetor do Campo Galois GF(q), em que q é a ordem do campo. Dito que alguém está codificando um arquivo que tem 64 KB de tamanho, se alguém usar $q = 2^{16}$ e $k = 16$, cada fragmento terá 4 KB e consistirá em uma palavra de 2 K, com cada palavra sendo um elemento de GF(2^{15}). Então, ERC gera fragmentos codificados provenientes dos fragmentos originais. Um fragmento ERC codificado é formado pela operação:

$$c_j = G_i \square [x_0\ x_1\ \ldots\ x_{k-1}]^i, \quad (2)$$

em que c_j é um fragmento codificado, G_i é um vetor gerador k-dimensional, e a e-

quação (2) é uma multiplicação de matriz, todas em GF(q). No momento da decodificação, o par coleta m fragmentos codificados, em que m é um número igual ou ligeiramente maior que k, e tenta decodificar os k fragmentos originais. Isto é equivalente a resolver a equação:

(copiar fórmula pg 12)(3)

Se a matriz formada pelos vetores do gerador tiver uma classificação completa k, as mensagens originais podem ser recuperadas.

Há muitos ERCs disponíveis. Um particularmente interessante é o código Reed-Solomon (RS). O código RS usa vetores de gerador estruturados, e é máxima distância separável (MDS). Em decorrência disto, quaisquer k fragmentos codificados distintivos poderão decodificar os fragmentos originais. Uma outra vantagem do código RS é que o fragmento codificado pode ser facilmente identificado e gerenciado pelo índice i do vetor gerador, assim, facilitando a detecção dos códigos RS duplicados. Na seguinte discussão de ERC, considera-se que o código RS é usado. Entretanto, o sistema de armazenamento com codificação adaptativa pode ser implementado com qualquer número de ERCs convencionais.

2.3 ERC: Número de fragmentos

Pelo uso de ERC em armazenamento P2P, um arquivo de dados é distribuído a mais pares, mas cada par precisa somente armazenar um fragmento codificado que é $1/k$ em tamanho do arquivo original, levando a uma redução geral na largura de banda e no espaço de armazenamento exigido para alcançar o mesmo nível de confiabilidade e, assim, a uma melhoria na eficiência. Deixe que n_1 seja o número de pares que os fragmentos codificados precisa que sejam distribuídos para alcançar um certo nível de confiabilidade desejado. Desde que o código RS seja código MDS, k pares que detêm k fragmentos codificados distintivos serão suficientes para recuperar o arquivo original. A probabilidade de que haja exatamente m pares disponíveis pode ser calculada por meio da distribuição binomial:

(copiar fórmula pg 13)(4)

Assim, pode-se calcular n_1 a partir de p, p_0 e k como:

(copiar fórmula pg 13)(5)

A taxa de replicação r é definida como:

$r = n_1 / k$(6)

A taxa de replicação r é um bom indicador de eficiência, já que r cópias de arquivos precisam ser distribuídas e armazenadas na nuvem P2P.

É mostrada na figura 4 a taxa de replicação desejada alcançando confiabilidade "seis nove" para diferentes números de fragmentos ERC k. Observa-se que o uso de ERC reduz enormemente a taxa de replicação exigida. Comparando o número de fragmentos não ERC (k = 1) e ERC k = 256, a taxa de replicação desejada diminui de r = 132 para r = 13,1 para confiabilidade de par de 10 %, de r = 20 para r = 2,5 para confiabilidade de par de 50

%, e de r = 3 para r = 1,05 para confiabilidade de par de 99 %. ERC pode melhorar a eficiência sem sacrificar a confiabilidade.

Observa-se que um maior número de fragmentos ERC reduz adicionalmente a taxa de replicação. Com uma confiabilidade de par de 50 %, ir de k = 8 para 16, 32, 64, 128 e 256 leva a uma redução da taxa de replicação de r = 5,75 para 4,375, 3,53, 3,02, 2,68 e 2,48. A melhoria de eficiência correspondente é de 24 %, 19 %, 15 %, 11 % e 8 %, respectivamente. Isto parece sugerir que deve-se usar grande número de fragmentos ERC para mais eficiência.

Entretanto, um maior número de fragmentos ERC implica em que mais pares são necessários para armazenar e recuperar os fragmentos codificados. Da forma mostrada na figura 5, o número de pares que precisam deter os fragmentos codificados para alcançar a confiabilidade "seis nove" é esboçado. Novamente, com 50 % de confiabilidade de par, ir de k = 8 para 16, 32, 64, 128 e 256 aumenta o número de pares que armazenam informação de n_i = 46 para 70, 113, 193, 343 e 630. Cada duplicação de k resulta em 52 %, 61 %, 71 %, 78 %, 84 % mais pares necessários para armazenar a informação. A duplicação de k também exige pelo menos o dobro de número de pares a ser contatados durante a recuperação de informação.

Na maior parte das redes P2P práticas, estabelecer uma conexão entre os pares exige uma quantidade não trivial de sobreprocessamento. Uma parte do sobreprocessamento pode ser atribuída à recuperação da identidade do par apropriado e a encontrar a trajetória de roteamento apropriada (por exemplo, por meio da Tabela de Dispersão Distribuída (DHT)). Uma outra parte do sobreprocessamento é em função da necessidade de invocar certos algoritmos de tradução de endereço de rede (NAT), por exemplo, STUN (travessia simples de UDP por meio de NAT) se um ou ambos os pares estiverem atrás da NAT. Considerando que o sobreprocessamento médio para estabelecer a conexão entre dois pares é sobre*processamento* (ajustado em 16 KB neste exemplo), pode-se calcular a largura de banda geral da rede necessária para armazenar um arquivo de tamanho s por:

$$store\_bandwidth = s * r + n_1 * sobreprocessamento \quad (7)$$

Com a equação (7), percebe-se que um maior número de fragmentos ERC nem sempre leva à maior eficiência. Em vez disto, para um pequeno arquivo, um pequeno número de fragmentos ERC ou mesmo não ERC deve ser usado. Computa-se a largura de banda geral exigida na equação (7) para diferentes tamanhos de arquivos e número de fragmentos ERC, e esboça-se as curvas mostradas na figura 6. O limite entre diferentes números de fragmentos ERC é a faixa do tamanho de arquivo ideal adequada para um número de fragmentos ERC em particular. Por exemplo, a curva de base da figura 6 mostra o limite do tamanho de arquivo abaixo do qual não ERC deve ser usado, e acima do qual ERC com número de fragmentos k = 2 deve ser usado. Uma observação interessante é que o limite de

tamanho do arquivo é relativamente insensível à disponibilidade de par, o que simplifica enormemente a escolha do parâmetro de fragmento ERC ideal. No geral, para um arquivo menor do que aproximadamente 10 KB, ERC não deve ser usado. Para ERC com um número de fragmentos k = 2, 4, 8, 16, 32, 128 e 256 a faixa de tamanho do arquivo mais adequada é aproximadamente 10 – 33 KB, 33 – 100 KB, 100 – 310 KB, 310 – 950 KB, 950 KB – 2,9 MB, 2,9 MB – 8,9 MB, 8,9 MB – 26 MB, > 26MB, respectivamente.

2.4 Esquema ERC Adaptativo

Os sistema e método de armazenamento com codificação adaptativa escolhem adaptativamente o número de fragmentos ERC apropriado para armazenar eficientemente conteúdo em uma rede P2P de forma confiável. Usando a curva de limite de arquivo estabelecida na figura 6, uma modalidade do sistema escolhe adaptativamente usar não ERC e ERC com um número de fragmentos k = 2, 4, 8, 16, 32, 64, 128, 256 para diferentes tamanhos de arquivo. A abordagem ERC adaptativa é comparada com o parâmetro ERC fixo, e a diferença no uso da largura de banda da rede é mostrada na figura 7 e na figura 8, em que a confiabilidade do par é de 50 % e de 99 %, respectivamente. Comparado com o uso de um número de fragmentos ERC fixo k = 1 (não ERC), 8, 32 e 256, o método ERC adaptativo pode melhorar a eficiência em uma média de 61 %, 26 %, 25 % e 50 % para confiabilidade de par de 50 %, e 50 %, 18 %, 29 % e 57 % para confiabilidade de par de 99 %. A melhoria na eficiência é significativa.

No sentido mais geral, uma modalidade do processo de armazenamento com codificação adaptativa é mostrada na figura 9. Da forma mostrada na ação do processo 902, o sistema de armazenamento com codificação adaptativa calcula limites de tamanho de arquivo ideais para um diferente número de fragmentos. Um arquivo de um dado tamanho de arquivo a ser codificado é inserido (ação de processo 904). É feita uma verificação se o tamanho de arquivo inserido corresponde à codificação não apagamento (k = 1), como mostrado na ação do processo 906. Se o tamanho do arquivo inserido não corresponder a ERC, o arquivo é codificado sem usar ERC (ação de processo 908). Se o tamanho de arquivo corresponder a uma faixa de tamanho de arquivo ERC, o arquivo é codificado usando codificação ERC e o número de fragmentos correspondente ao tamanho de arquivo do arquivo inserido, que é o número de fragmentos ideal para aquele tamanho de arquivo (ação de processo 910).

Uma aplicação principal do processo ERC adaptativo aqui descrito é uma cópia de segurança ou restauração P2P. Um par pode fazer cópia de segurança de arquivos em outros pares em uma rede e, então, restaurar estes arquivos pela recuperação deles a partir dos pares na rede no caso de eles se perderem (por exemplo, no caso em que eles são perdidos em um travamento do computador). No geral, a figura 10 ilustra um fluxograma operacional exemplar que mostra como a técnica de armazenamento com codificação adap-

tativa pode ser empregada em um sistema P2P. Percebe-se que todas as caixas e interconexões entre as caixas que são representadas por linhas rompidas ou tracejadas na figura 10 representam modalidades alternativas do sistema de armazenamento com codificação adaptativa aqui descrito, e que qualquer uma ou todas estas modalidades alternativas, descritas a seguir, podem ser usadas em conjunto com outras modalidades alternativas que são descritas por todo este documento.

Em particular, como ilustrado pela figura 10, antes das operações de transferência de dados, tal como quando for desejado fazer cópia de segurança de dados em pares em uma rede, o servidor 200 ou o par 220 codifica 1000 os dados a ser transferidos a outros pares para armazenamento. O sistema de armazenamento com codificação adaptativa pode operar com qualquer um de inúmeros codecs convencionais, tais como, por exemplo, MPEG 1/2/4, WMA, WMV, etc. Além do mais, durante o processo de codificação 1000, o servidor 200 ou o par 220 também gera tanto um cabeçalho de dados quanto um arquivo parceiro que contém a estrutura de dados.

Como exposto, em uma modalidade, uma vez que os dados são codificados 1000, os pacotes de dados codificados são divididos 1005 em inúmeras unidades de dados de tamanho fixo. Adicionalmente, como com os dados codificados, o cabeçalho de dados e a estrutura de dados também são divididos 1005 em inúmeras unidades de dados do mesmo tamanho fixo, usadas para dividir os pacotes de dados codificados. Dividir 1005 esta informação em unidades de dados de tamanho fixo permite que os pares pré-aloquem blocos de memória antes das operações de transferência de dados, desse modo, evitando operações de alocação de memória computacionalmente onerosas durante o processo de transferência de dados. Adicionalmente, o uso de unidades de dados menores permite controle mais fino pelo cliente ou par que armazena os dados sobre a quantidade exata de largura de banda gasta por cada par para satisfazer as solicitações de unidade de dados do cliente durante as operações de transferência de dados.

Além de dividir 1005 os dados codificados, o cabeçalho de dados e a estrutura de dados em unidades de dados menores, se codificação resiliente de apagamento for empregada, uma camada adicional de codificação é usada para fornecer maior redundância em um ambiente P2P típico, em que servir pares é inerentemente não confiável. Em particular, como exposto, em uma modalidade, se codificação resiliente de apagamento for determinada como apropriada para o arquivo de dados, as unidades de dados são adicionalmente divididas em inúmeros blocos de dados e um processo de codificação resiliente de apagamento 1010 é usado para codificar o arquivo.

O uso de tal codificação 1010 garante que um ou mais dos pares terá os blocos de dados necessários para reconstruir as unidades de dados em particular, simplificando a demanda do cliente para identificar qual dos pares contém os dados necessários. Adicional-

mente, em uma modalidade, as chaves de codificação resiliente de apagamento usadas por cada par de serviço 220 são automaticamente atribuídas a cada par pelo servidor 200. Entretanto, em uma outra modalidade, cada par de serviço 220 simplesmente escolhe uma chave de codificação resiliente de apagamento aleatoriamente. Então, estas chaves são recuperadas pelo cliente 210 quando cada par 220 é inicialmente contatado pelo cliente.

Uma vez que o arquivo de dados foi inicialmente codificado 1000, dividido em unidades de dados 1005 e, possivelmente, adicionalmente codificado com apagamento 1010, então, as unidades de dados ou blocos de dados resultantes são distribuídos 1015 aos vários pares 220. Esta distribuição 1015 pode ser deliberada no sentido de que os blocos ou pacotes dos dados codificados são simplesmente fornecidos, no todo ou em parte, a inúmeros pares em que, então, eles são ocultados ou armazenados para futura transferência de dados quando chamados por um cliente que deseja recuperar os dados.

Uma vez que os dados foram distribuídos 1015 aos pares de serviço 220, então, o cliente 210 está pronto para começar as solicitações de dados para aqueles pares no caso em que o cliente deseja recuperar estes dados a partir do armazenamento. Adicionalmente, como exposto, o servidor 200 também pode agir como um par 220 com os propósitos de transferir dados para o cliente 210.

Neste ponto, o cliente 210 começa uma sessão de transferência de dados, primeiro, pela recuperação de uma lista de pares de serviço 220 disponíveis. Esta lista é diretamente recuperada do servidor 200, de um dos pares 220 ou pelo uso de um método convencional de tabela de dispersão distribuída (DHT) para identificar pares de serviço em potencial. Uma vez que o cliente 1010 recuperou a lista de pares, então, o cliente se conecta em cada par de serviço 220 e recupera 1025 uma lista de arquivos disponíveis de cada par. Uma vez que o cliente 210 recuperou a lista de arquivos disponíveis de cada par 220, então, o cliente recupera 1035 o cabeçalho de dados e a estrutura de dados dos dados a ser transferidos a partir de um ou mais dos pares pela solicitação de unidades de dados correspondentes àquela informação de um ou mais dos pares por meio de uma conexão de rede entre o cliente e aqueles pares.

No geral, o cabeçalho de dados contém informação global que descreve os dados, por exemplo, o número de canais nos dados, as propriedades e características (taxa de amostragem de áudio, taxa de resolução / quadro de vídeo) de cada canal, codecs usados, autor / titular de direito autoral da mídia, e assim por diante. Conseqüentemente, a recuperação do cabeçalho de dados no início da sessão de transferência de dados permite que o cliente 220 ajuste ou inicialize 1040 as ferramentas necessárias para decodificar 1065 os pacotes subseqüentemente recebidos antes da recepção destes pacotes durante a sessão de transferência de dados.

Adicionalmente, depois de recuperar 1035 a estrutura de dados dos dados em par-

ticular, o cliente analisa a estrutura de dados e calcula IDs da unidade de dados 1045 das unidades de dados dos dados transferidos que precisarão ser solicitados durante o processo de transferência de dados. Então, o cliente 210 solicita estas unidades de dados 1050, uma a uma, a partir de um ou mais dos pares 220.

Finalmente, uma vez que todas as unidades de dados que constituem um pacote de dados em particular foram recuperadas de acordo com a solicitação 1050 do cliente 210, estes pacotes de dados são remontados 1055 no pacote de dados original. Então, pacotes de dados remontados são decodificados 1060 e podem ser restaurados 1065 no cliente 210.

3.0 Armazenamento P2P: Políticas e Estratégias de Projeto

Além de ajustar o número de fragmentos ERC com base no tamanho do arquivo a ser armazenado na rede P2P, a eficiência também pode ser melhorada. Várias modalidades do sistema de armazenamento com codificação adaptativa aqui descrito são projetadas para melhorar a eficiência de armazenamento pelo emprego de certas estratégias como descrito a seguir. Estas estratégias podem ser empregadas em conjunto com o sistema de armazenamento com codificação adaptativa ou podem ser empregadas em qualquer rede P2P.

3.1 Custo de Armazenamento P2P

Nesta seção, armazenar um arquivo em uma rede P2P é comparado com armazenar o arquivo diretamente em um servidor confiável "seis nove". Observa-se que a solução P2P reduz a largura de banda e os custos do servidor, mas exige que o par gaste mais largura de banda para distribuir o arquivo no armazenamento P2P. O uso geral da largura de banda da rede aumenta na solução P2P. O aumento na largura de banda do carregamento do cliente pode ser considerado um custo do sistema de armazenamento P2P. Este custo para diferentes confiabilidades e tamanhos de arquivo de par é tabulado na Tabela 1.

Tabela 1 – Custo do maior uso de largura de banda em P2P

	Tamanho do Arquivo				
Confiabilidade	10 KB	100 KB	1 MB	10 MB	100 MB
10 %	332,9	79,1	29,5	16,5	12,5
50 %	51,0	12,11	4,34	2,23	1,56
99 %	9,4	1,87	0,55	0,22	0,09

Observa-se que o custo do uso de armazenamento P2P é pequeno se a confiabilidade do par for alta e o arquivo for grande. Por exemplo, armazenar 100 MB de arquivo em pares com confiabilidade de 99 % incorre em somente 9 % de custo. Entretanto, quando a confiabilidade do par for baixa e o arquivo for pequeno, o custo pode ser significativo.

3.2 Políticas de Armazenamento P2P

A partir da Tabela 1, pode-se derivar a seguinte política de uso da nuvem de armazenamento P2P:

a) Deve-se usar pares não confiáveis para armazenar grandes arquivos, e usar pa-

res confiáveis para armazenar pequenos arquivos. O custo do sistema P2P será menor se grandes arquivos forem alocados em pares não confiáveis, e se forem atribuídos arquivos menores a pares confiáveis.

b) Deve-se usar pares não confiáveis para armazenar arquivos estáticos, e deve-se usar pares confiáveis para armazenar arquivos dinâmicos. Chama-se aqueles arquivos que não mudam de estáticos e chama-se aqueles arquivos que mudam constantemente de dinâmicos. Múltiplos pequenos arquivos estáticos podem ser empacotados em um grande arquivo estático e armazenados na nuvem de armazenamento P2P. A mesma estratégia não é efetiva para arquivos dinâmicos, já que a mudança de um único arquivo exige que todo o arquivo combinado seja atualizado.

Uma conseqüência desta política é que se a rede P2P for usada para armazenar o estado de uma aplicação, informação de estado do par, e assim por diante, deve-se desviar a informação até os pares mais confiáveis da rede. Se for restrito que o arquivo que contém o estado da aplicação seja colocado somente em pares altamente confiáveis (em essência, os pares altamente confiáveis formarão uma sub-rede que constitui os núcleos da rede P2P estendida), pode-se reduzir enormemente esta taxa de replicação e o custo de atualizar o arquivo de estado, e melhorar a eficiência.

c) Deve ser permitido que pares não confiáveis distribuam menos, e deve ser permitido que pares confiáveis distribuam mais.

d) Deve ser atribuído a arquivos menores um maior custo de distribuição, e deve ser atribuído a arquivos maiores um menor custo de distribuição.

As políticas c) e d) são para aplicações de cópia de segurança e de recuperação P2P, em que um par pode distribuir conteúdo para a nuvem de armazenamento P2P e armazenar conteúdo para outros pares. Uma rede de armazenamento P2P equilibrada deve deixar cada par equilibrar sua contribuição e benefício. Em trabalhos anteriores, apontou-se que a largura de banda é recurso primário na aplicação de armazenamento P2P. Deixe a contribuição do par ser a quantidade de fragmentos codificados que ele recebe e armazena para os outros pares. Deixe o benefício do par ser a quantidade de conteúdo que ele distribui para a nuvem P2P. Levando em consideração que baixa confiabilidade leva ao armazenamento de dados mais redundante, deve-se punir pares não confiáveis para permitir que eles distribuam menos, e deve-se recompensar pares confiáveis para permitir que eles distribuam mais. Tal política pode ter um benefício positivo na economia P2P, já que ela encoraja o usuário a manter a aplicação P2P em linha, assim, melhorando a confiabilidade geral da rede P2P e reduzindo a taxa de replicação exigida.

Também pode-se punir a distribuição de um pequeno arquivo pela sua atribuição com um alto custo de distribuição, exigindo que o par contribua proporcionalmente mais, e pode-se recompensar a distribuição de um grande arquivo pela sua atribuição com um baixo

custo de distribuição, deixando que o par contribua proporcionalmente menos. Como uma conseqüência, aplicações de cópia de segurança P2P devem ser projetadas para minimizar a freqüência de cópia de segurança. Em vez de atualizar imediatamente o arquivo exatamente depois de sua mudança, pode-se considerar empacotar múltiplas mudanças em um grande arquivo e atualizá-lo somente uma vez, dito toda meia-noite, na nuvem de armazenamento P2P.

Uma modalidade do sistema e método de armazenamento com codificação adaptativa que são projetados ao redor das políticas expostas é mostrada na figura 11. Da forma mostrada na ação de processo 1102, a confiabilidade de cada par na rede distribuída ou P2P é determinada. Um arquivo a ser distribuído ou armazenado é inserido (ação de processo 1104). O tamanho do arquivo é avaliado (ação de processo 1106) e um custo de distribuição é atribuído ao arquivo com base na largura de banda de armazenamento esperada na equação (7) (ação de processo 1108). Se o arquivo for um grande arquivo, um custo de distribuição mais alto pode ser atribuído. Se o arquivo for pequeno, pode ser atribuído ao arquivo um menor custo de distribuição. Com base no tamanho do arquivo, o sistema de armazenamento com codificação adaptativa escolherá pares com confiabilidade apropriada para armazenar o arquivo (ação de processo 1110). Isto é, os pares cuja confiabilidade está abaixo de um dado limite são usados para armazenar e distribuir o grande arquivo, e pares cuja confiabilidade está abaixo de um dado limite são usados para armazenar e distribuir o pequeno arquivo.

Uma outra modalidade dos sistema e método de armazenamento com codificação adaptativa que são projetados ao redor das políticas expostas é mostrada na figura 12. Da forma mostrada na ação de processo 1202, a confiabilidade de cada par na rede distribuída ou P2P é determinada. Um arquivo a ser distribuído ou armazenado é inserido (ação de processo 1204). O arquivo é comparado com o mesmo arquivo que foi previamente armazenado para determinar se o arquivo é estático ou dinâmico (ação de processo 1206). Na primeira vez que o arquivo for depositado, considera-se que o arquivo é dinâmico. Se forem observadas mudanças freqüentes no arquivo, o arquivo permanece designado como dinâmico. Se for observado que o arquivo não muda por um período de tempo prolongado, o arquivo é designado como estático. Os arquivos dinâmicos são armazenados em pares altamente confiáveis (ação de processo 1210) (Assim, em primeiro lugar, arquivos serão armazenados em servidores ou pares altamente confiáveis.). Uma vez que observa-se que os arquivos não mudam, e eles ficam estáticos, estes arquivos estáticos serão redistribuídos e armazenados em pares com confiabilidade mais baixa.

Percebe-se que as modalidades mostradas nas figuras 11 e 12 podem ser usadas sozinhas ou em conjunto a fim de aumentar a eficiência e confiabilidade gerais de uma rede distribuída ou ponto a ponto.

3.3 Armazenamento P2P com Suporte ao Componente de Servidor

Se um componente de servidor for usado em conjunto com a rede P2P, pode-se usar o armazenamento P2P para grandes arquivos estáticos, e pode-se usar o servidor para pequenos arquivos dinâmicos. Uma vez que são os grandes arquivos que consomem a maior parte dos recursos do servidor, o armazenamento P2P complementa bem o servidor.

Da forma mostrada na figura 13, uma modalidade dos sistema e processo de armazenamento com codificação adaptativa emprega cópia de segurança P2P com suporte ao servidor. Da forma mostrada na figura 13, é feita cópia de segurança dos arquivos dinâmicos na rede no servidor (ação de processo 1302). Então, o cliente e/ou o servidor podem detectar automaticamente aqueles arquivos dinâmicos que não mudaram mais e estão se transformando em arquivos estáticos (ações de processo 1304, 1306). Então, estes arquivos estáticos detectados podem ser empacotados juntamente em um grande arquivo, da forma mostrada na ação de processo 1308, e podem ser distribuídos com ERC na nuvem de armazenamento P2P (ação de processo 1310). Isto aumenta efetivamente o tamanho do arquivo armazenado na nuvem P2P. Combinado com ERC de um grande número de fragmentos, isto pode melhorar a eficiência.

A modalidade mostrada na figura 13 pode ser usada sozinha ou em conjunto com as modalidades mostradas nas figuras 11 e 12 para aumentar a eficiência e confiabilidade gerais de uma rede distribuída ou ponto a ponto. Também percebe-se que esta modalidade pode ser usada tanto com codificação resiliente de apagamento quanto sem ela.

Percebe-se que qualquer uma ou todas as modalidades alternativas supramencionadas podem ser usadas em qualquer combinação desejada para formar modalidades híbridas adicionais. Embora o assunto em questão tenha sido descrito em linguagem específica para recursos estruturais e/ou atos metodológicos, entende-se que o assunto em questão definido nas reivindicações anexas não é necessariamente limitado aos recursos ou atos específicos supradescritos. Em vez disto, os recursos e atos específicos supradescritos são divulgados como formas de exemplos da implementação das reivindicações.

REIVINDICAÇÕES

1. Processo implementado em computador para codificar arquivos a ser armazenados em uma rede distribuída, **CARACTERIZADO** pelo fato de que compreende as ações de processo de:

calcular faixas de tamanho de arquivo ideais correspondentes a diferentes números de fragmentos de codificação resiliente de apagamento (ERC), em que cada número de fragmentos é o número de fragmentos ideal para uma faixa de tamanhos de arquivo correspondente (902);

inserir um arquivo de um dado tamanho de arquivo (904);

se o tamanho do arquivo for menor do que a faixa de tamanhos de arquivo para o menor número de fragmentos ERC de dois, codificar o arquivo sem usar codificação resiliente de apagamento (906, 908);

se o tamanho do arquivo do arquivo inserido corresponder a uma faixa de tamanhos de arquivo, codificar o arquivo usando codificação resiliente de apagamento e o número de fragmentos ideal correspondente à faixa de tamanho de arquivo do arquivo inserido (906, 910).

2. Processo implementado em computador, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que compreende adicionalmente a ação de processo de transmitir o arquivo codificado a um ou mais pares em uma rede distribuída.

3. Processo implementado em computador, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que compreende adicionalmente computar o número de pares em que o arquivo codificado será armazenado de acordo com a confiabilidade do par e com a confiabilidade desejada do conteúdo do arquivo.

4. Processo implementado em computador, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que calcular as faixas de tamanho de arquivo ideais correspondentes a diferentes números de fragmentos de codificação resiliente de apagamento (ERC) compreende as ações de processo de:

determinar um limite entre diferentes números de fragmentos como o tamanho de arquivo ideal adequado para um número de fragmentos ERC em particular.

5. Processo implementado em computador, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que compreende adicionalmente as ações de processo de:

obter um conjunto de fragmentos de arquivo do arquivo codificado igual ou maior do que um número de fragmentos em que o arquivo é dividido para codificação; e

decodificar os fragmentos de arquivo codificados com decodificação resiliente de apagamento se o arquivo foi codificado por codificação resiliente de apagamento para obter uma versão decodificada do arquivo codificado; e

decodificar os fragmentos codificados sem decodificação resiliente de apagamento

se o arquivo não foi codificado por codificação resiliente de apagamento para obter uma versão decodificada do arquivo codificado.

6. Processo implementado em computador, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que a codificação resiliente de apagamento usada na codificação do arquivo é codificação Reed Solomon.

7. Processo implementado em computador, de acordo com a reivindicação 6, **CARACTERIZADO** pelo fato de que:

se o tamanho do arquivo for menor de que aproximadamente 10 KB, codificação resiliente de apagamento não é usada;

se o tamanho do arquivo for aproximadamente 10 KB até 33 KB, o número de fragmentos ideal é dois;

se o tamanho do arquivo for aproximadamente 33 KB até 100 KB, o número de fragmentos ideal é quatro;

se o tamanho do arquivo for aproximadamente 100 KB até 310 KB, o número de fragmentos ideal é oito;

se o tamanho do arquivo for aproximadamente 310 KB até 950 KB, o número de fragmentos ideal é dezesseis;

se o tamanho do arquivo for aproximadamente 950 KB até 2,9 MB, o número de fragmentos ideal é trinta e dois;

se o tamanho do arquivo for aproximadamente 2,9 MB até 8,9 MB, o número de fragmentos ideal é sessenta e quatro;

se o tamanho do arquivo for aproximadamente 8,9 MB até 26 MB, o número de fragmentos ideal é cento e vinte e oito; e

se o tamanho do arquivo for maior do que aproximadamente 26 MB, o número de fragmentos ideal é duzentos e cinqüenta e seis.

8. Mídia legível por computador com instruções executáveis por computador, **CARACTERIZADA** pelo fato de que é para desempenhar o processo citado na reivindicação 1.

9. Sistema para melhorar a confiabilidade e a eficiência de armazenamento de uma rede ponto a ponto, **CARACTERIZADO** pelo fato de que compreende:

um dispositivo de computação de uso geral;

um programa de computador que compreende módulos de programa executáveis pelo dispositivo de computador de uso geral, em que o dispositivo de computação é direcionado pelos módulos de programa do programa de computador para:

determinar o número de fragmentos ideal para codificar um arquivo de dado tamanho com codificação resiliente de apagamento (1010);

se o número de fragmentos ideal para codificar o arquivo com codificação resiliente

de apagamento for um, não codificar o arquivo com codificação resiliente de apagamento (1010); e

se o número de fragmentos ideal for maior do que um, codificar o arquivo rompendo o arquivo no número de fragmentos ideal e codificar o arquivo com codificação resiliente de apagamento (1010).

10. Sistema, de acordo com a reivindicação 9, **CARACTERIZADO** pelo fato de que compreende adicionalmente um módulo de programa para computar o número de pares em que os fragmentos de arquivo codificados serão armazenados de acordo com a confiabilidade do par e com a confiabilidade desejada do conteúdo.

11. Sistema, de acordo com a reivindicação 10, **CARACTERIZADO** pelo fato de que compreende adicionalmente um módulo de programa para distribuir o arquivo codificado com codificação resiliente de apagamento a um ou mais pares em uma rede.

12. Sistema, de acordo com a reivindicação 11, **CARACTERIZADO** pelo fato de que o módulo de programa para distribuir o arquivo compreende submódulos para:

determinar a confiabilidade de cada par na rede distribuída;

determinar o tamanho do arquivo;

e usar um ou mais pares com confiabilidade apropriada determinada pelo tamanho do arquivo para distribuir o arquivo.

13. Sistema, de acordo com a reivindicação 12, **CARACTERIZADO** pelo fato de que o módulo de programa para distribuir o arquivo compreende submódulos para:

se o arquivo for grande, usar pares cuja confiabilidade está abaixo de um dado limite para distribuir o grande arquivo; e

se o arquivo não for grande, usar pares cuja confiabilidade está acima de um dado limite para distribuir o arquivo.

14. Sistema, de acordo com a reivindicação 11, **CARACTERIZADO** pelo fato de que o módulo de programa para distribuir o arquivo compreende submódulos para:

determinar a confiabilidade de cada par na rede distribuída;

determinar se o arquivo é estático;

se o arquivo for estático, usar pares cuja confiabilidade está abaixo de um dado limite para distribuir o arquivo; e

se o arquivo não for estático, usar pares cuja confiabilidade está acima de um dado limite para distribuir o arquivo.

15. Sistema, de acordo com a reivindicação 11, **CARACTERIZADO** pelo fato de que o módulo de programa para distribuir o arquivo compreende submódulos para:

determinar a confiabilidade de cada par na rede distribuída;

monitorar mudanças no arquivo;

primeiro, distribuir o arquivo para pares mais confiáveis;

se não for observada mudança no arquivo, redistribuir o arquivo para pares monos confiáveis.

16. Sistema, de acordo com a reivindicação 9, **CARACTERIZADO** pelo fato de que compreende adicionalmente um módulo de programa para melhorar a eficiência da rede distribuída usando um servidor, compreendendo adicionalmente submódulos para:

fazer cópia de segurança de todos os arquivos dinâmicos na rede distribuída para o servidor;

verificar periodicamente pares e o servidor na rede distribuída para ver se os arquivos dinâmicos com cópia de segurança no servidor mudaram;

se os arquivos dinâmicos não mudaram, designar estes arquivos como estáticos e empacotá-los em um grande arquivo; e

distribuir o grande arquivo com codificação resiliente de apagamento para a rede distribuída.

17. Sistema, de acordo com a reivindicação 9, **CARACTERIZADO** pelo fato de que o módulo de programa para determinar o número de fragmentos ideal para codificar um arquivo de um dado tamanho com codificação resiliente de apagamento compreende submódulos para:

determinar uma faixa de tamanho de arquivo ideal para cada número de fragmentos de codificação resiliente de apagamento, em que cada número de fragmentos é o número de fragmentos ideal para a faixa correspondente;

determinar em qual faixa de tamanho de arquivo o tamanho de um arquivo inserido cai; e

usar como o número de fragmentos ideal correspondente à faixa de tamanho de arquivo ideal na qual o tamanho do arquivo inserido cai.

18. Processo implementado em computador para decodificar um arquivo codificado armazenado em uma rede distribuída, **CARACTERIZADO** pelo fato de que compreende as ações de processo de:

recuperar um conjunto de fragmentos de um arquivo codificado igual ou maior do que o número de fragmentos que foi usado para codificar o arquivo, em que o arquivo foi codificado por codificação resiliente de apagamento com um número de fragmentos ideal para um dado tamanho; e

decodificar os fragmentos codificados com decodificação resiliente de apagamento para obter uma versão decodificada do arquivo codificado.

19. Processo implementado em computador, de acordo com a reivindicação 18, **CARACTERIZADO** pelo fato de que pelo menos alguns dos fragmentos codificados são recuperados a partir de uma mídia de armazenamento de um ou mais pares na rede distribuída.

20. Processo implementado em computador, de acordo com a reivindicação 18, **CARACTERIZADO** pelo fato de que pelo menos alguns dos fragmentos codificados são recuperados a partir de uma mídia de armazenamento de um servidor na rede distribuída.

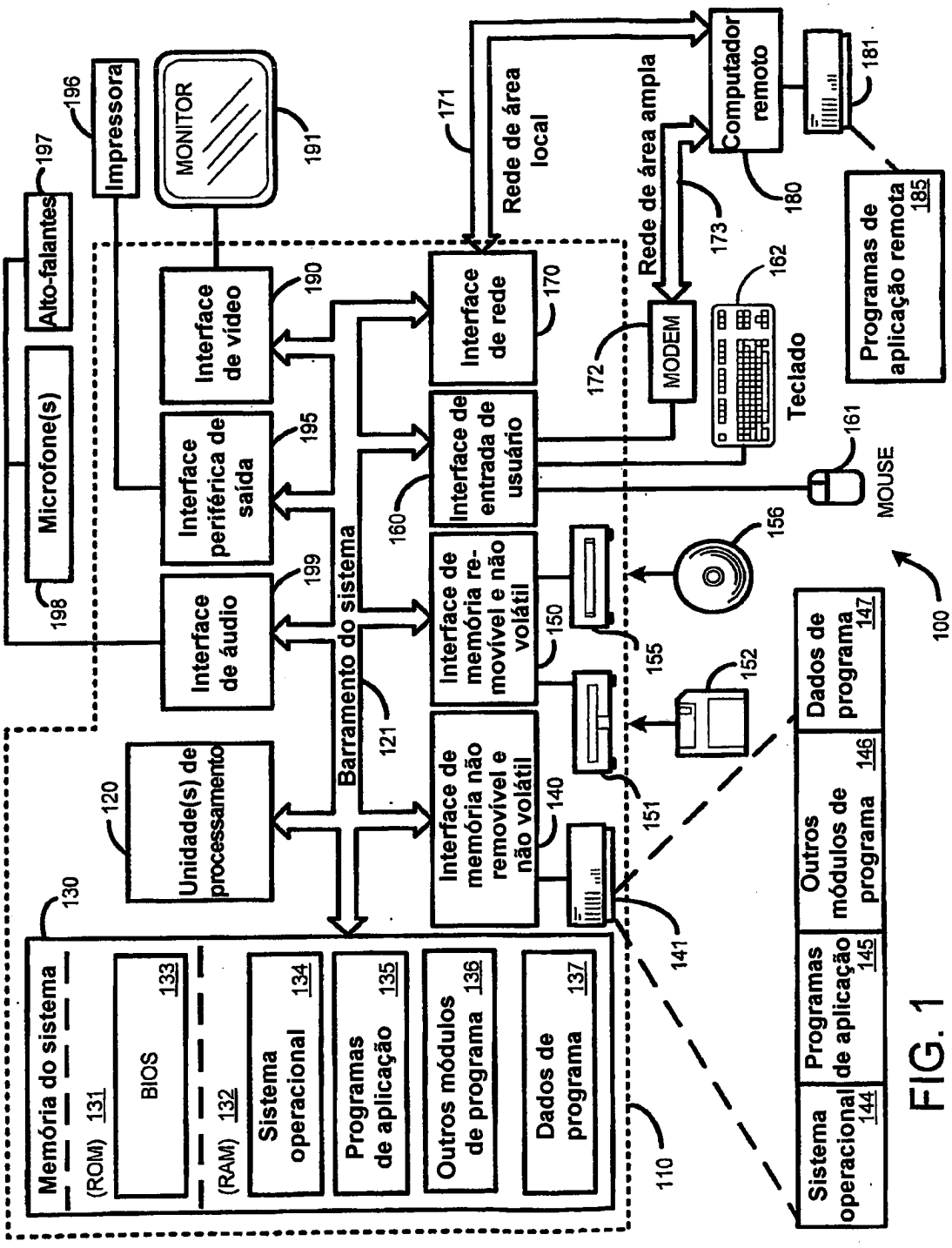

Memória do sistema 130
(ROM) 131
BIOS 133
(RAM) 132
Sistema operacional 134
Programas de aplicação 135
Outros módulos de programa 136
Dados de programa 137
Unidade(s) de processamento 120
Barramento do sistema 121
Interface de memória não removível e não volátil 140
Interface de memória removível e não volátil 150
Interface de entrada de usuário 160
Interface periférica de saída 195
Interface de áudio 199
Interface de vídeo 190
Interface de rede 170
Rede de área local 171
Rede de área ampla 173
MODEM 172
Computador remoto 180
Programas de aplicação remota 185
Teclado 162
MOUSE 161
Microfone(s) 198
Alto-falantes 197
Impressora 196
MONITOR 191
Sistema operacional 144
Programas de aplicação 145
Outros módulos de programa 146
Dados de programa 147
110
141
151
152
155
156
100

FIG. 1

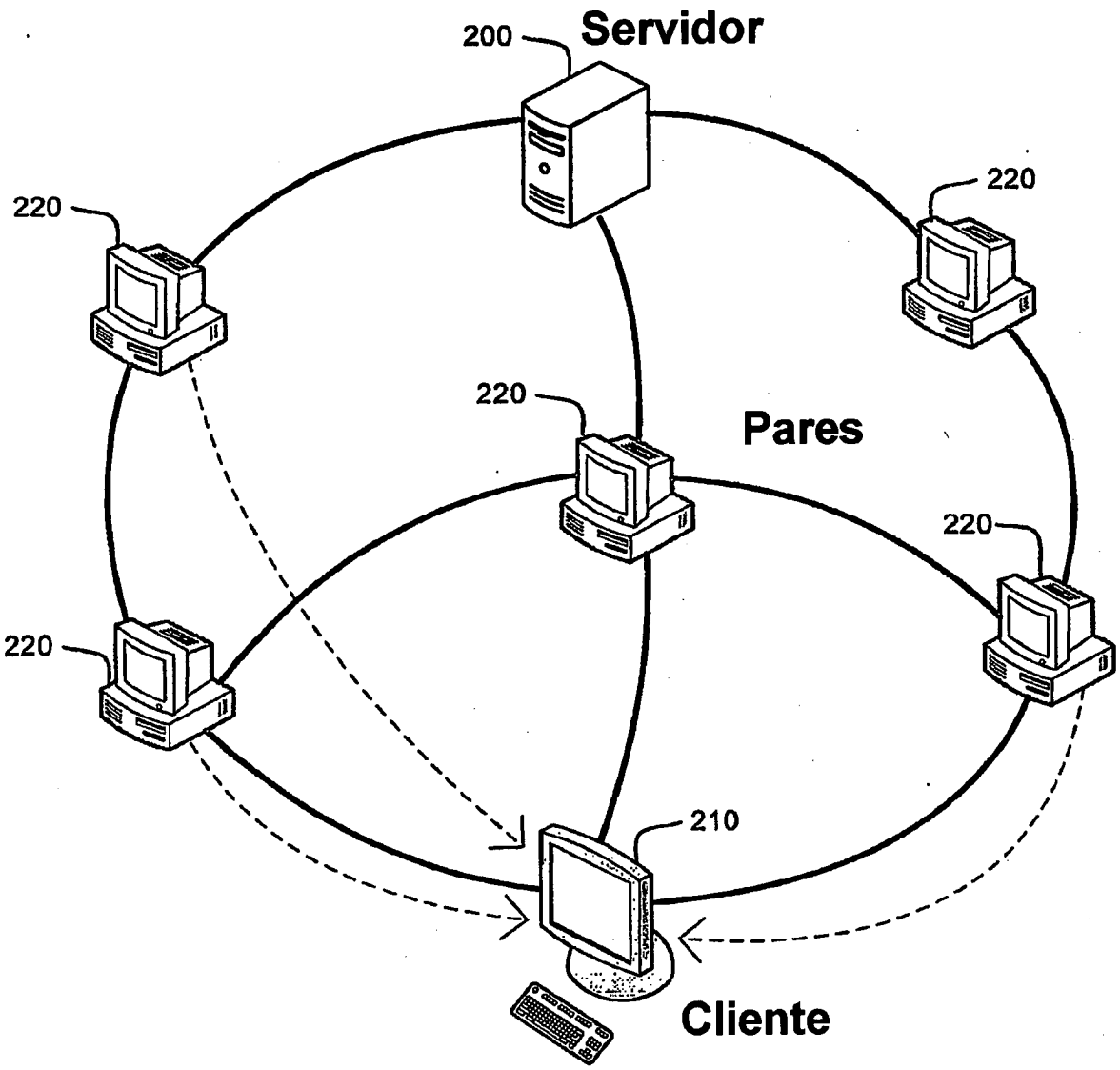


FIG. 2

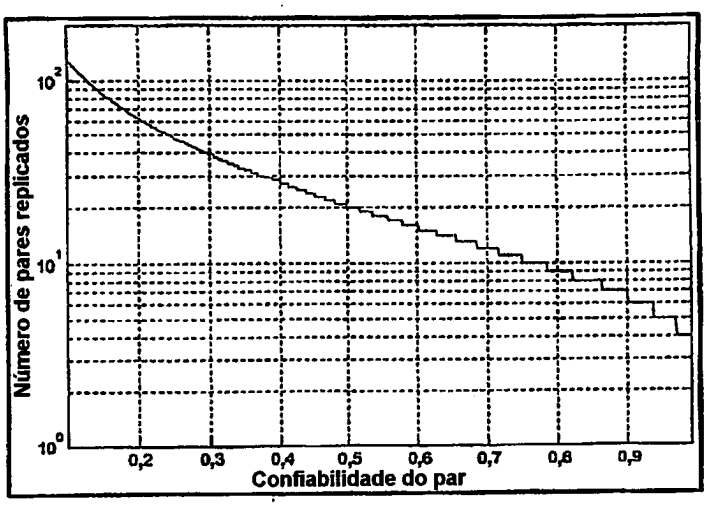


FIG. 3

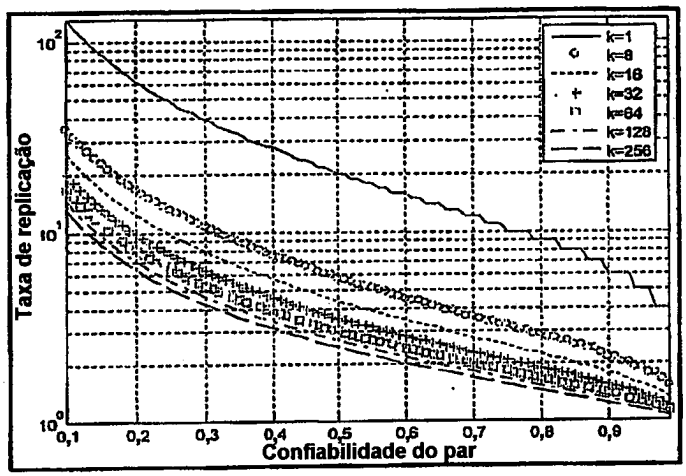


FIG. 4

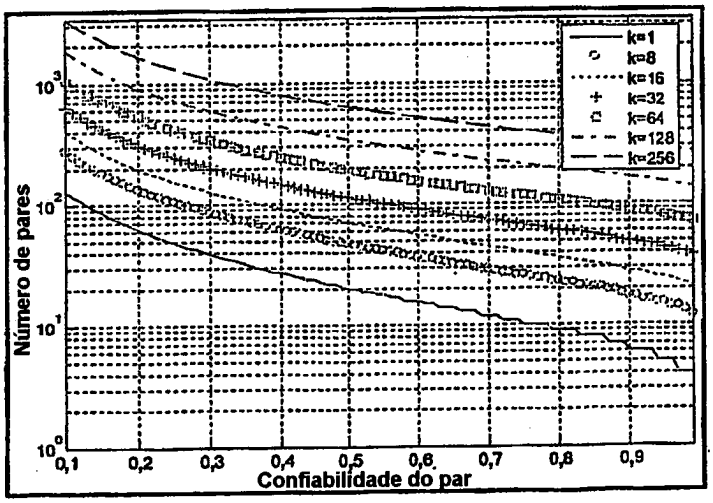


FIG. 5

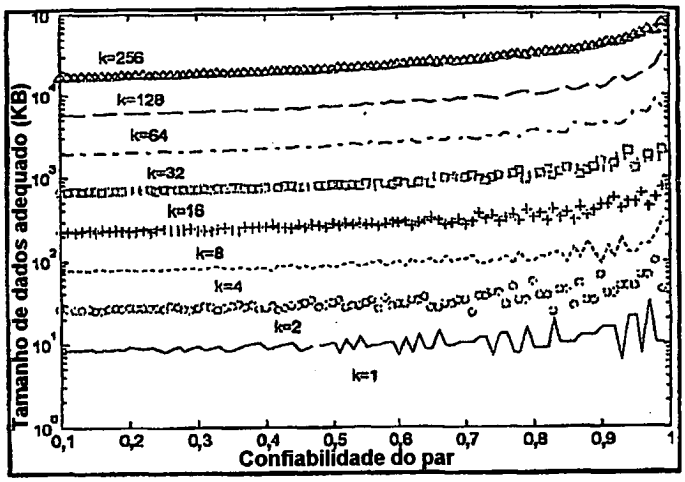


FIG. 6

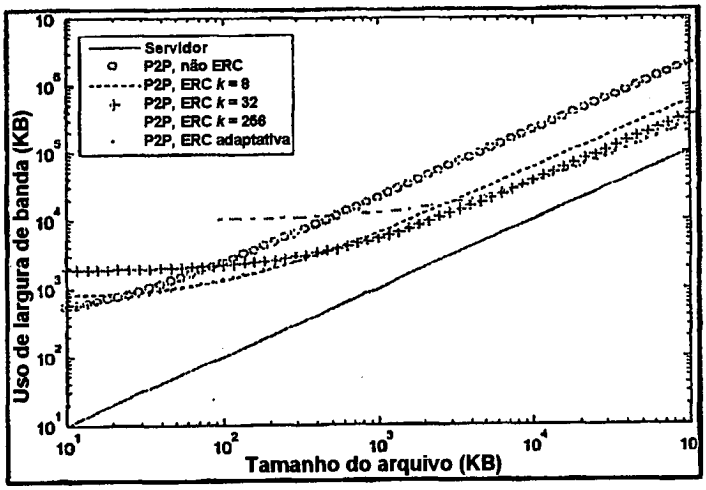


FIG. 7

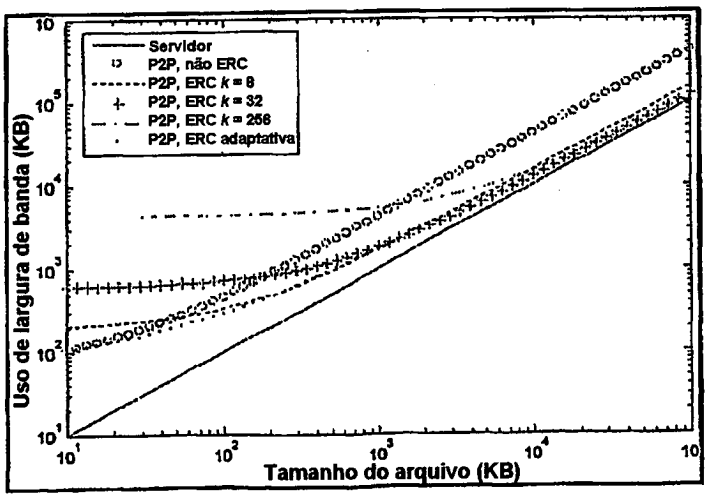
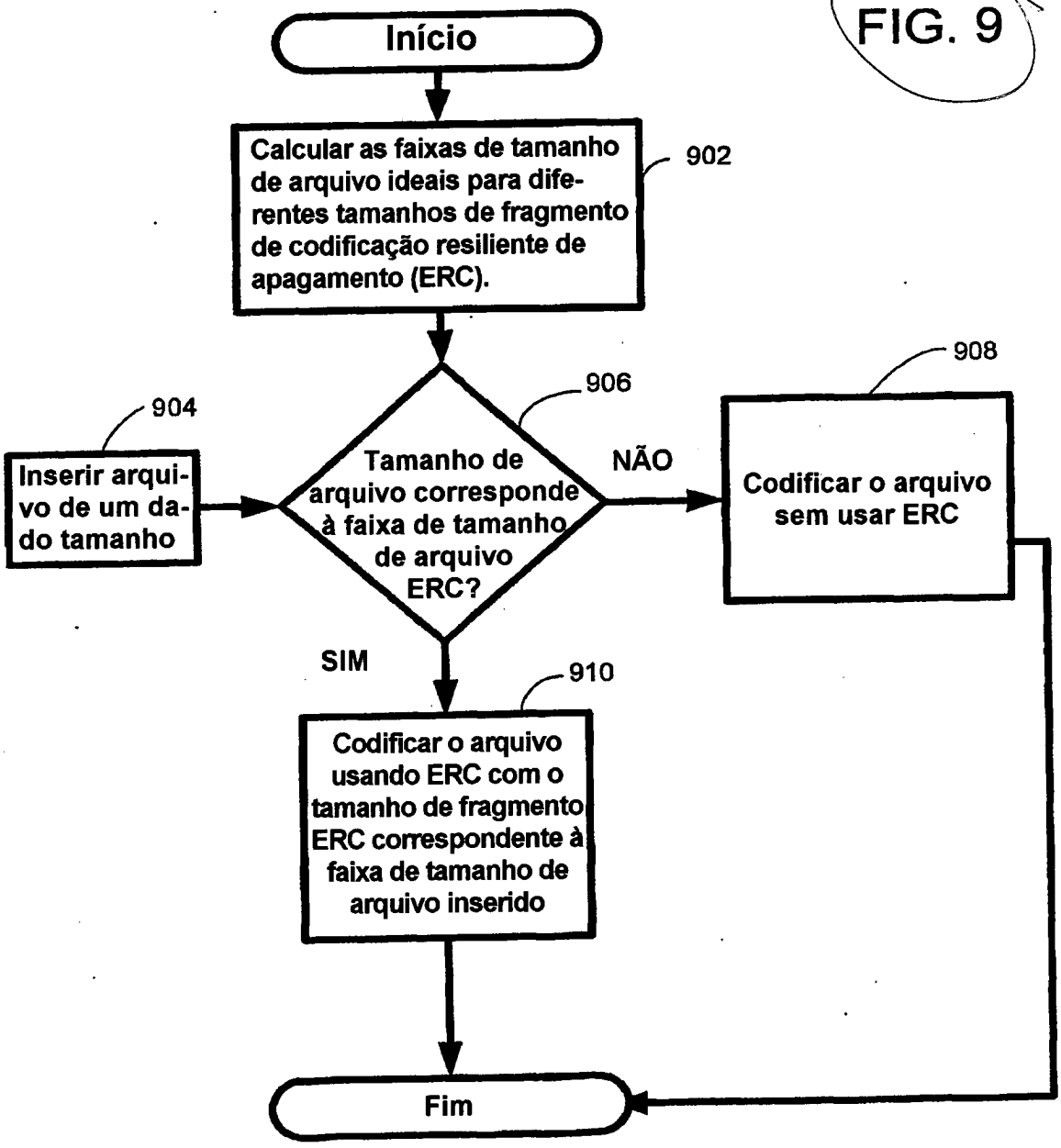


FIG. 8

FIG. 9


Início
Calcular as faixas de tamanho de arquivo ideais para diferentes tamanhos de fragmento de codificação resiliente de apagamento (ERC).
902
Inserir arquivo de um dado tamanho
904
Tamanho de arquivo corresponde à faixa de tamanho de arquivo ERC?
906
NÃO
Codificar o arquivo sem usar ERC
908
SIM
Codificar o arquivo usando ERC com o tamanho de fragmento ERC correspondente à faixa de tamanho de arquivo inserido
910
Fim

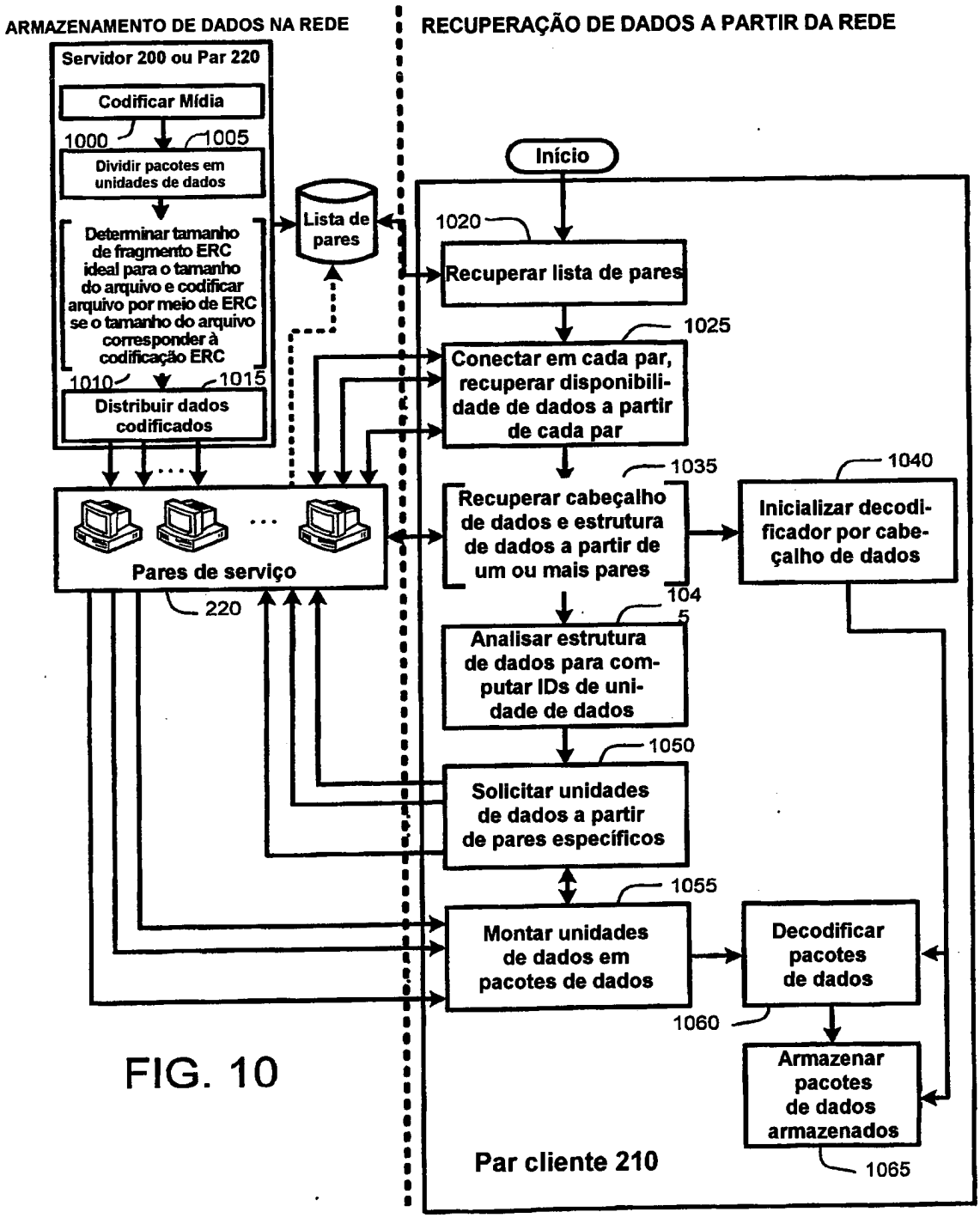

ARMAZENAMENTO DE DADOS NA REDE
RECUPERAÇÃO DE DADOS A PARTIR DA REDE
Servidor 200 ou Par 220
Codificar Mídia
1000
1005
Dividir pacotes em unidades de dados
Determinar tamanho de fragmento ERC ideal para o tamanho do arquivo e codificar arquivo por meio de ERC se o tamanho do arquivo corresponder à codificação ERC
1010
1015
Distribuir dados codificados
Lista de pares
Pares de serviço
220
Início
1020
Recuperar lista de pares
1025
Conectar em cada par, recuperar disponibilidade de dados a partir de cada par
1035
Recuperar cabeçalho de dados e estrutura de dados a partir de um ou mais pares
1040
Inicializar decodificador por cabeçalho de dados
104
5
Analisar estrutura de dados para computar IDs de unidade de dados
1050
Solicitar unidades de dados a partir de pares específicos
1055
Montar unidades de dados em pacotes de dados
Decodificar pacotes de dados
1060
Armazenar pacotes de dados armazenados
1065
Par cliente 210

FIG. 10

FIG. 11

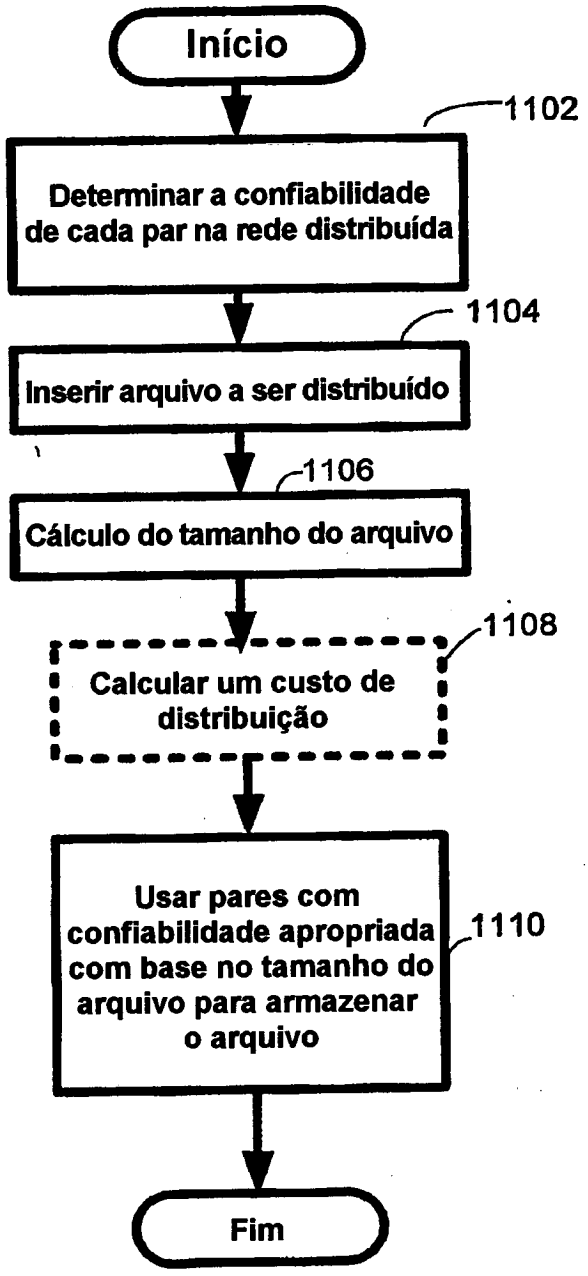
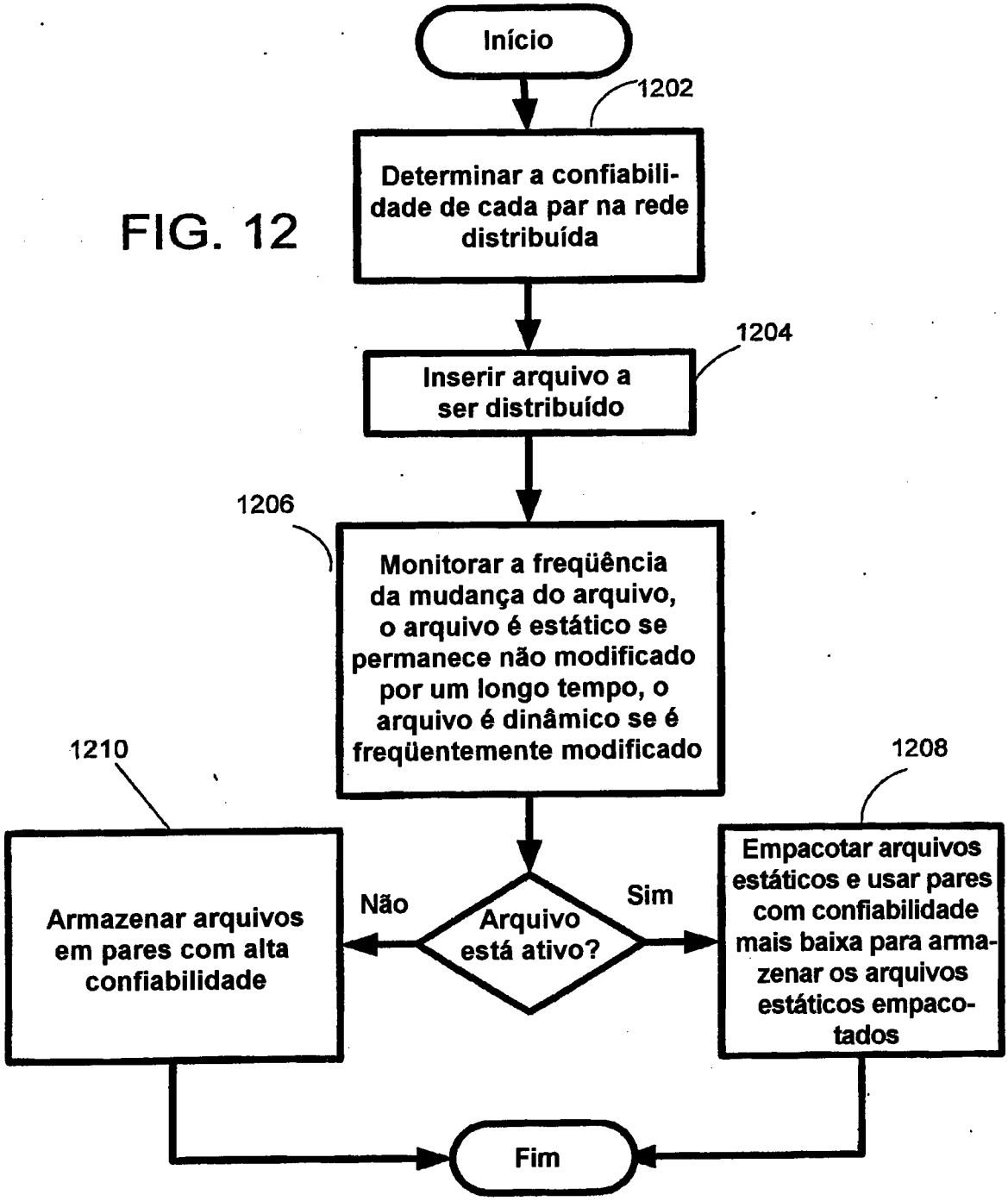

Início
1102
Determinar a confiabilidade de cada par na rede distribuída
1104
Inserir arquivo a ser distribuído
1106
Cálculo do tamanho do arquivo
1108
Calcular um custo de distribuição
Usar pares com confiabilidade apropriada com base no tamanho do arquivo para armazenar o arquivo
1110
Fim

FIG. 12


Início
1202
Determinar a confiabilidade de cada par na rede distribuída
1204
Inserir arquivo a ser distribuído
1206
Monitorar a freqüência da mudança do arquivo, o arquivo é estático se permanece não modificado por um longo tempo, o arquivo é dinâmico se é freqüentemente modificado
1210
1208
Armazenar arquivos em pares com alta confiabilidade
Não
Arquivo está ativo?
Sim
Empacotar arquivos estáticos e usar pares com confiabilidade mais baixa para armazenar os arquivos estáticos empacotados
Fim

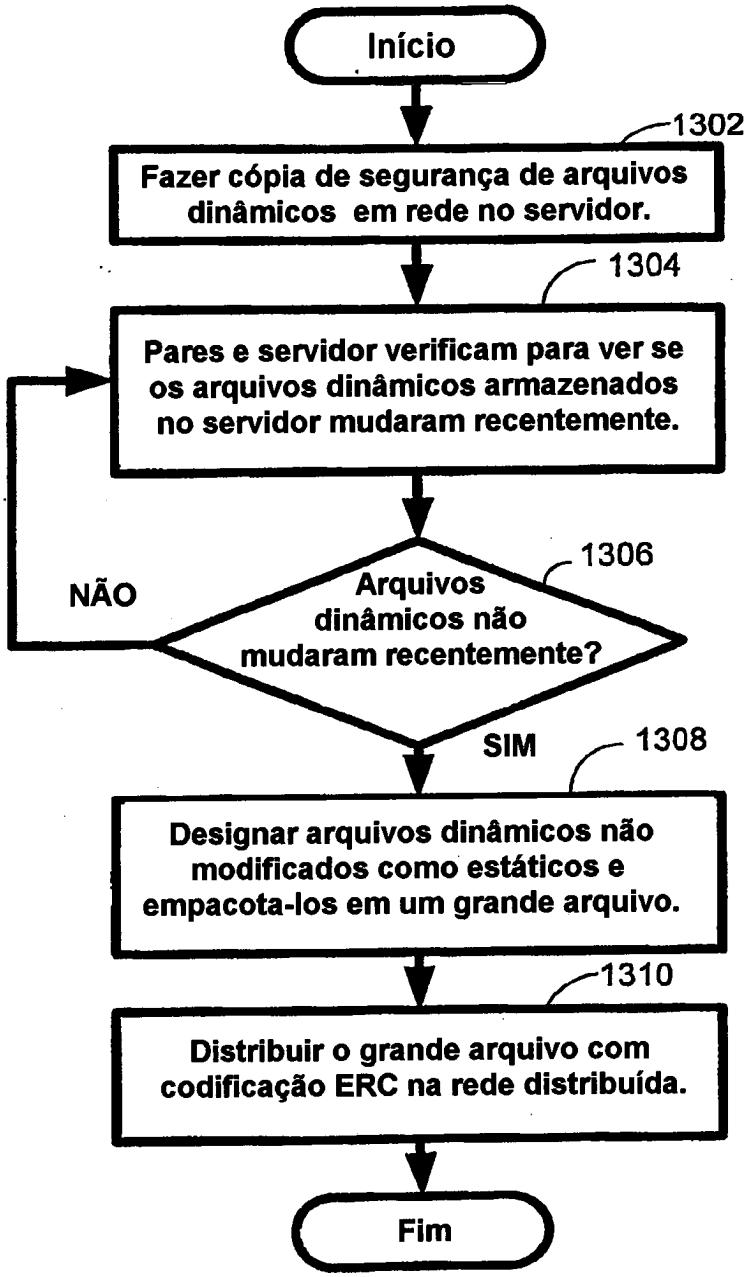

Início
1302
Fazer cópia de segurança de arquivos dinâmicos em rede no servidor.
1304
Pares e servidor verificam para ver se os arquivos dinâmicos armazenados no servidor mudaram recentemente.
1306
Arquivos dinâmicos não mudaram recentemente?
NÃO
SIM
1308
Designar arquivos dinâmicos não modificados como estáticos e empacota-los em um grande arquivo.
1310
Distribuir o grande arquivo com codificação ERC na rede distribuída.
Fim

FIG. 13

PI0708199-5

RESUMO

"ARMAZENAMENTO PONTO A PONTO CONFIÁVEL E EFICIENTE"

É divulgado um sistema de armazenamento com codificação adaptativa que usa codificação resiliente de apagamento (ERC) adaptativa que muda o número de fragmentos usados para codificação de acordo com o tamanho do arquivo distribuído. ERC adaptativa pode melhorar enormemente a eficiência e a confiabilidade do armazenamento P2P. Inúmeros procedimentos para aplicações de armazenamento P2P também podem ser implementados. Em uma modalidade, pequenos arquivos de dados dinâmicos são desviados para os pares mais confiáveis ou mesmo para um servidor, enquanto que grandes arquivos estáticos são armazenados utilizando-se a capacidade de armazenamento dos pares não confiáveis. Também, para contribuição e benefício equilibrados, um par deve hospedar a mesma quantidade de conteúdo armazenada na rede P2P. Em decorrência disto, pares não confiáveis podem distribuir menos dados, e pares confiáveis podem distribuir mais. Também, arquivos menores são atribuídos com um custo de distribuição mais alto, e os arquivos maiores são atribuídos com um custo de distribuição mais baixo.