

Sistema de Arquivos

Soluções de armazenamento baseadas em um sistema de arquivos distribuídos estão crescendo ainda mais rápido, em números e capacidade de implementações. O motivo: crescimento acentuado e contínuo em dados não estruturados em data centers empresariais. Dobrando os volumes de dados não estruturados a cada ano, as organizações estão ligando os sistemas de arquivos distribuídos para gerenciar volumes de dados com maior desempenho, facilidade operacional aprimorada e custos menores.

Para obter os benefícios desse fluxo de dados valiosos, em vez de se afogar nele, os CIOs precisam de soluções de armazenamento escaláveis, flexíveis e automatizadas que sejam capazes de aproveitar o fluxo de trabalho da nuvem híbrida. O sistema de arquivos distribuído certo não deve apenas atender a esses critérios, mas ajudar a oferecer a energia que as plataformas de lógica analítica que necessitam de desempenho exigem para extrair a percepção e a inteligência do capital de dados.

O Dell EMC Isilon oferece um sistema de arquivos distribuídos que foi reconhecido pelo Magic Quadrant do Gartner por três anos seguidos como líder em abrangência de visão e capacidade de executar dentro da categoria de armazenamento em objeto e arquivo distribuído.

Em um mercado em que a capacidade de maximizar o valor dos dados é vital para o sucesso dos negócios, as organizações precisam de soluções de armazenamento de dados que possam melhorar o acesso e a análise do capital de dados.

No passado, as organizações só esperavam que seu sistema de arquivos distribuídos fosse capaz de dimensionar — ele tinha de ser grande o suficiente para gerenciar os dados hoje e acomodar o crescimento de dados amanhã. O alto desempenho não era necessariamente uma prioridade, uma vez que a maioria das empresas assumiu que a grande parte dos dados de arquivo seria "frio" e não precisaria de velocidades super-rápidas e da latência mínima do armazenamento de dados "quentes".

As cargas de trabalho modernas mudaram essa equação, e hoje nenhuma empresa quer ser deixada para trás por um sistema de arquivos distribuídos que não consegue acompanhar os requisitos de uma empresa digital. Grande não é mais o suficiente — as soluções de armazenamento de hoje devem ser grandes, rápidas e eficientes. Além disso, um sistema de arquivos distribuídos superior deve oferecer:

Custos menores por meio de redução de consumo de energia e custo operacional (OPEX) reduzido.

Utilização do recurso aprimorada, permitindo que as organizações obtenham maior ROI de seus investimentos em armazenamento.

Reduzir o custo total de propriedade.

Acordos de Nível de Serviço (SLAs) aprimorados para atender as rígidas demandas das necessidades dos negócios e dos requisitos de conformidade.

Não podemos iniciar nenhuma discussão sobre sistemas de arquivos distribuídos sem definir alguns conceitos fundamentais.

Um serviço é um conjunto de facilidades oferecidas aos nós de uma rede por um software que opera em uma ou mais máquinas.

Um servidor é o software que opera em uma determinada máquina e que trata de oferecer o serviço. Chamaremos, também, de servidor a máquina que executa esse software.

O cliente é o software que utiliza o serviço do servidor. Também chamaremos de cliente a máquina que executa o software cliente.

Um sistema de Arquivos é uma parte de um sistema Operacional que trata de oferecer um repositório de dados de longa duração.

Um Sistema de Arquivos Distribuído é um Sistema de Arquivos onde vários servidores são responsáveis por oferecer o serviço de arquivos para vários clientes instalados em diferentes máquinas.

Transparência, consiste em promover acesso a recursos distribuídos de forma oculta, como se fosse um único sistema para o usuário. Considerando os SAD's, de acordo com Coulouris, Dollimore e Kindberg (2005) e Tanenbaum e Steen (2007) ela pode ser dividida em transparecias:

De acesso: não necessita fornecer a localização dos recursos, ou seja, os programas devem executar os processos de leitura e escrita de arquivos remotos da mesma maneira que operam sobre os arquivos locais, sem qualquer modificação no programa. O usuário não deve perceber se o recurso acessado é local ou remoto.

De localização: os programas clientes devem ver um espaço de nomes de arquivos uniforme, sem a necessidade de fornecer a localização física dos arquivos para encontrá-los, mesmo que esses arquivos se desloquem entre os servidores.

De mobilidade: independente dos arquivos se moverem entre servidores, os programas clientes não precisam ser alterados para a nova localidade do grupo de arquivos. Essa característica permite flexibilidade em mover arquivos sem comprometer toda a estrutura, ou ter que refazer links entre programas clientes e o local do arquivo.

De desempenho: o desempenho da aplicação cliente não poderá ser comprometido enquanto ocorre uma variação dos processos sobre os recursos disponíveis pelos SAD's, isto é, mesmo que haja concorrência no acesso pelos arquivos isso não deve afetar os usuários.

De escalabilidade: os recursos computacionais podem sofrer alterações para abrigar maior poder computacional ou o ingresso de novos servidores sem prejudicar o serviço.

Contra falhas: garantir a disponibilidade dos arquivos ininterruptamente e se ocorrerem falhas o programa cliente não deverá saber como elas serão tratadas.

De replicação: várias cópias dos mesmos arquivos armazenados em locais diferentes para garantir a disponibilidade. A aplicação cliente deverá visualizar apenas uma cópia do mesmo, não necessitando saber a quantidade replicada e o local.

Como o nome fala por si mesmo, alterações feitas por um cliente não devem interferir nas operações de outros clientes, mesmo que esses clientes estejam manipulando o mesmo arquivo.

Quando diversos processos acessam os mesmos dados simultaneamente, é preciso tomar cuidado para que esses processos recebam informações corretas e também para que a consistência do sistema de arquivos não seja afetada.

Replicação De Arquivos

O SAD deve manter cópias atualizadas dos arquivos em diferentes locais.

Essa medida traz algumas benesses como balanceamento de carga, pois mais servidores podem atender a uma determinada requisição, melhora a escalabilidade e tolerância a falhas, pois se caso algum arquivo falhar, ele pode ser requisitado a algum outro servidor que o tenha, retornando a um estado consistente, sem que os clientes tenham conhecimento.

Heterogeneidade

O SAD deve ser concebido levando em conta a heterogeneidade de plataformas (sistemas operacionais, arquiteturas de processador, dentre outros fatores), que podem ser usadas, tanto do lado dos servidores como dos clientes.

Um rede Heterogênea, isto é, que possui nós de diferentes tipos, pode ser vantajosa na medida em que cada tipo de tarefa pode ser executada em uma máquina apropriada.

Tolerância A Falhas

O serviço deve continuar executando mesmo com falhas parciais. Tanto os clientes quanto os servidores de arquivos podem sofrer quedas e romperem a comunicação com os outros nós da rede por intervalos que podem variar entre segundos e até horas.

O sistema deve evitar, sempre que possível, que falhas como esta causem uma depreciação muito grande no tempo de resposta aos clientes ou que o serviço seja interrompido.

Consistência

Só deve haver uma versão disponível de cada arquivo no SAD, ou seja, se algum cliente ler um arquivo X em um momento, e outro cliente ler o mesmo arquivo X em outro momento e/ou outro lugar, se X não foi modificado, ambos os clientes devem visualizar exatamente o mesmo conteúdo.

Proteção

O SAD deve ter preocupação com permissão (a pessoa que está acessando o arquivo pode fazê-lo?), autenticação (essa pessoa é realmente ela?), privacidade (dados secretos podem ser vistos por outras pessoas?) e integridade (esse dado arquivo foi danificado no transporte?).

Eficiência

O SAD deve fazer todas as atividades listadas nos requisitos acima com um desempenho comparável a, ou melhor que, um sistema de arquivos centralizado, de modo que, mesmo se a carga nos servidores varie constantemente, o funcionamento de aplicações clientes do SAD não sejam muito penalizadas.

A provisão dos serviços oferecidos pelos sistemas de arquivos distribuídos é feita pelo serviço de arquivo e pelo serviço de diretório, que serão abordados a seguir.

Serviço De Arquivo

O serviço de arquivo é responsável por expor a interface de manipulação dos arquivos, em outras palavras, é a definição dos serviços providos pelo SAD no tocante a manipulação dos arquivos como leitura, escrita e alteração do conteúdo e alguns atributos do arquivo. Esses atributos do arquivo pode ser algo como o dono, tamanho, data de criação, permissões, dentre outros.

Quando o cliente está manipulando um determinado arquivo, o mesmo pode estar completamente copiado e disponível ou qualquer operação será feita via rede. A essas duas abordagens de construção do serviço de arquivo dá-se o nome de cópia remota e acesso remoto respectivamente.

Quando a manipulação é feita completamente remota, ou seja sem que nenhuma parte do arquivo seja copiada, o gerenciamento de versões do arquivo é menos complexo, pois toda alteração é feita diretamente no arquivo. A cópia remota será tratada em uma seção posterior, sob o nome de cache.

Serviço De Diretório

O serviço de diretório, por sua vez, é responsável por informar quais as operações para manipulação de diretórios, ou seja, é a definição dos serviços providos pelo SAD no que concerne a criação e remoção de diretórios, movimentação de arquivos de um diretório para outro, nomear e renomear arquivos.

Além disso este serviço também precisa manter uma lista com todos os diretórios ativos, seus respectivos arquivos e ainda dos diretórios que eventualmente forem apagados, remover os arquivos filhos deste que fora apagado, se o cliente tiver a devida permissão.

Além das operações naturais dos diretórios, o serviço de diretório auxilia o serviço de arquivos a desempenhar algumas de suas atividades como listar os arquivos de algum diretório, efetuar buscas por determinados arquivos.

Como conseguimos encontrar um arquivo no sistema de arquivos de um computador doméstico onde o sistema de arquivos é centralizado?

Cada arquivo possui um path, do inglês caminho, ou localização no referido sistema de arquivos. Como fazer com um sistema de arquivos distribuído se todos os arquivos de um diretório podem não estar no mesmo computador dele? Para esse problema existe o serviço de resolução de nomes.

Serviço de resolução de nomes

É uma parte bem complexa de um SAD pois precisa descobrir, em tempo útil, a localização dos arquivos solicitados. Cada arquivo deve ter seu nome mapeado para o servidor que o armazena, de modo que as aplicações consigam identificar unicamente os arquivos. Existem três formas básicas de implementar a nomeação de arquivos e de diretórios:

máquina + caminho_do_arquivo, ex. maquina/caminho/do/arquivo ou maquina:caminho/do/arquivo

montar pedaços do sistema de arquivos remoto na hierarquia local de arquivos;

um único espaço de nomes, ou seja, o mesmo espaço de nomes em todas as máquinas.

As técnicas 1 e 2 são especialmente usadas para distribuir um conjunto de sistemas de arquivos que inicialmente não foram concebidos para serem distribuídos. Já a terceira técnica é a desejada de um SAD, pois provê a transparência suficiente para que quem o use, sinta como se estivesse acessando um sistema de arquivos local.

Para não gerar tráfego demasiado na rede com muita informação de controle, ferindo o requisito de eficiência definido, pode-se usar o mecanismo de cache, tanto nos servidores do SAD quanto no cliente.

Cache

Como o sistema está em uma rede local, o tráfego excessivo é indesejado. Para diminuir essa quantidade de acessos, pode-se recorrer ao mecanismo de cache.

O cache aumenta o desempenho pois reduz a quantidade de acessos aos arquivos guardados no SAD, com isso são feitas menos requisições usando a rede.

É possível manter o arquivo inteiro em cache (mais simplicidade na implementação e menos eficiência) ou manter apenas um trecho que está sendo editado (mais eficiente, porém de difícil implementação). A literatura prevê alguns modelos de atualização de arquivos da cache para o SAD, são eles:

Write through

O cliente mantém um arquivo (ou trecho) sendo utilizado e acontecendo qualquer mudança nele, a mudança é refletida de imediato nos servidores.

Prós: leituras subseqüentes a uma escrita estarão sempre atualizadas.

Contras: Ocorre um problema nesse modelo, imagine:

```
* um cliente A altera um arquivo f e o tem em seu cache e o servidor mantém a versão atualizada de f.
* A fecha o arquivo f e ainda mantém f em cache.
* outro cliente, adote B, abre o arquivo f, o modifica e o fecha e o servidor está com a última versão editada por B.
* se A abrir f, ele verá que está em seu cache, portanto não o requisitará ao SAD e com isso terá uma versão não atualizada.
```

Uma melhoria ao modelo anterior. Melhoria no sentido de as escritas não serem a cada modificação, mas sim havendo alguma modificação e considerando algum intervalo de tempo, as alterações são enviadas ao SAD.

E melhoria também no momento que um cliente abre um dado arquivo que está em cache.

É verificado se esse arquivo está atualizado com o arquivo no SAD, se não estiver, o mesmo é atualizado e enfim o mesmo é aberto.

Prós: versões de arquivos em cache são verificadas de modo a não haver inconsistência.

Contras: leituras feitas depois de uma escrita, não estarão garantidamente atualizadas, pois o tempo de atualização pode ainda não ter passado.

Write on close

Todas as alterações são submetidas sempre quando o arquivo é fechado.

Prós: reduz bastante a quantidade de atualizações no SAD.

Contras: problemas de edição simultânea, pois quem fechar o arquivo por último é que terá sua versão persistida.

Centralize control

Centraliza o controle de acesso aos arquivos no servidor, de modo que ele enfileira as requisições de leitura e escrita de modo a: - Se algum acesso ao arquivo for feito apenas para leitura, nenhuma precaução precisa ser tomada - Se um arquivo é aberto para escrita, o servidor não permite que outro cliente abra o mesmo arquivo para escrita.

Prós: facilidade em bloquear uma escrita indevida.

Contras: não é elegante, pois o servidor que é proativo e não o cliente e ainda necessita de verificação de cache no momento de um novo acesso.

Após a exposição desses conceitos e detalhes acerca dos SAD, a seguir serão expostos alguns exemplos mais famosos de SAD.

O SUN Network File System é um sistema de arquivos distribuído no qual clientes acessam os arquivos armazenados em apenas um servidor.

Criado em 1984, sofreu algumas modificações e no ano seguinte a SUN Microsystems o tornou público, assim sendo possível encontrar atualmente implementações do NFS para quase todos os sistemas operacionais.

Como o NFS é stateless, ou seja, não mantém nenhum estado de qualquer requisição feita a eles, não corre risco de haver perda de informações em caso de queda do servidor, mas por esse mesmo motivo, não consegue lidar com acesso concorrente, entregando a cada cliente uma cópia do mesmo arquivo.

A equipe de desenvolvimento do NFS tinha em mente os seguintes objetivos para seu sistema de arquivos:

Funcionar em diversas plataformas de hardware e de software;

Tanto o cliente quanto o servidor serem capazes de se recuperar rapidamente de crashes;

Acesso transparente, de modo que as aplicações-cliente não necessitem ser modificadas para acessar tais arquivos;

Manter a mesma semântica do sistema de arquivos do UNIX, de modo que a transparência seja maior ainda em máquinas UNIX;

Cumprir todos esses objetivos e ainda manter um desempenho aceitável, comparável a um sistema de arquivos SCSI local.

Algumas decisões foram tomadas para a implementação do NFS. Dentre elas veremos algumas a diante.

Sistema De Nomes

No NFS se pode montar um pedaço do sistema de arquivos distribuído em algum diretório do sistema de arquivos local do cliente, de modo que a nomenclatura dos diretórios provenientes do NFS segue o mesmo padrão do sistema de arquivos local.

Portanto o NFS provê transparência no acesso, pois se o servidor onde os dados se encontram mudar de lugar, o cliente não precisa saber dessa mudança para continuar a usar o NFS.

Segurança

O NFS usa o mesmo esquema de conceder e verificar permissões do UNIX.

O NFS não dá suporte a acesso concorrente e lock de arquivos, portanto a alteração do mesmo arquivo poderá implicar em resultados indesejáveis.

Semântica

Foi implementada a semântica UNIX para o tratamento dos arquivos. Nela, por exemplo, é permitido que um arquivo possa ser excluído enquanto alguma aplicação o utiliza. Outra particularidade dessa semântica é de que a permissão de um arquivo pode ser alterada enquanto o mesmo está aberto. Se essa alteração for feita no arquivo remoto, ela sempre será checada, caso contrário não.

Com todas essas funcionalidades, o NFS ainda funciona com eficiência, e como indício disso, continua sendo largamente usando ainda hoje.

Andrew File System

O Andrew File System ou AFS surgiu na Universidade Carnegie Mellon com o intuito de prover compartilhamento de arquivos entre os, pelo menos, 7 mil estudantes, professores e funcionários da referida universidade¹, além de tentar avançar o estado da arte em computação distribuída.

Antes de tudo, como todos na universidade de Carnegie Melo usavam UNIX, a primeira decisão de projeto sobre o AFS foi que ele deveria ser completamente compatível com o UNIX, no nível das chamadas ao sistema operacional.

Outra decisão foi de que a unidade básica de tráfego do sistema foi o arquivo inteiro; isso sem dúvida é o aspecto mais controverso e interessante do AFS.

A integridade do cache é implementado usando o princípio de callback. Quando um arquivo é aberto, é posto em cache no sistema de arquivos local e o servidor guarda essa informação.

Sempre que algum outro cliente alterar o mesmo arquivo, ele é avisado via callback de que o arquivo foi alterado. Toda alteração é checada através desse callback.

Dentre as funcionalidades do AFS, podemos citar:

Autenticação

O cliente se autentica usando suas informações de conta de usuário UNIX.

Access Control List

As ACESS Control List ou ACL são implementadas de modo a restringir o acesso indevido aos arquivos armazenados no AFS. Elas são:

Leitura de arquivos

Escrita (Atualização) em arquivos

Inserção (Criação) de arquivos

Deleção de arquivos

Pesquisa nos arquivos

Bloquear arquivos

Gerenciar diretórios (mudar ACL)

O AFS é atualmente¹ distribuído com o Linux, mas ainda em versão experimental.