

Sistemas de Arquivos Distribuídos

Sistemas Operacionais

Prof. Pedro Ramos
pramos.costar@gmail.com

Última aula: Sistemas distribuídos e RPCs

- Os servidores exportam rotinas para que algum conjunto de clientes chame
- Para usar o servidor, o cliente faz uma chamada de rotina
- O sistema operacional gerencia a comunicação

Hoje: Sistemas de arquivos distribuídos

- Um dos usos mais comuns dos sistemas distribuídos
- Ideia básica:
 - Dado um conjunto de discos conectados diferentes nós.
 - Discos compartilhados entre nós como se todos os discos estivessem conectados a cada nó.
- Exemplos:
 - **CRC**: Um ou mais nós de servidor com todos os discos e várias estações de trabalho sem disco (ou com um disco de tamanho mínimo) em uma LAN.
 - Apple **iCloud**
 - **Google** Drive
 - **Kubernetes**

Sistemas de arquivos distribuídos: Questões

- *Nomenclatura* e *transparência*
- Acesso *remoto* a arquivos
- Armazenamento em *cache*
- Servidor com *estado* ou sem estado
- *Replicação*

Nomenclatura e transparência

- *Problemas*

- Como os arquivos são nomeados?
- Os nomes dos arquivos revelam sua localização?
- Os nomes de arquivos mudam se o arquivo for movido?
- Os nomes dos arquivos mudam se o usuário se mudar?

- *Transparência do local: o nome do arquivo não revela o local de armazenamento físico.*

- *Independência de localidade: O nome do arquivo não precisa mudar se o local de armazenamento do arquivo mudar.*

- *A maioria dos esquemas de nomenclatura usados na prática não tem independência de localidade, mas muitos têm transparência do local.*

Estratégias de nomenclatura: Nomes absolutos

- Nomes absolutos: `<nome da máquina: nome do caminho>`
- Vantagens:
 - Encontrar um nome de arquivo totalmente especificado é **simples**.
 - É **fácil adicionar** e **excluir** novos nomes.
 - **Nenhum estado** global.
 - **Escala** facilmente.
- Desvantagens:
 - O **usuário** deve saber o **nome completo** e está ciente de quais arquivos são locais e quais são remotos.
 - O **arquivo depende** do **local** e, portanto, não pode ser movido.
 - Torna o **compartilhamento** mais **difícil**.
 - **Não** é **tolerante** a **falhas**.

Estratégias de nomenclatura: Pontos de montagem

- Pontos de montagem
 - Cada host tem um **conjunto** de **nomes locais** para **locais remotos**.
 - Cada **host** tem uma **tabela** de **montagem** (**/etc/fstab**) que especifica <nome do caminho remoto @ nome da máquina> e um <nome do caminho local>.
 - No momento da inicialização, o **nome local é vinculado ao nome remoto**.
 - Os usuários então se referem ao <nome do caminho local> como se fosse local, e o **NFS (Network File System)** cuida do mapeamento
- Vantagens: localização transparente, **o nome remoto pode mudar** entre reinicializações
- Desvantagens: estratégia única e unificada difícil de manter, o **mesmo arquivo pode ter nomes diferentes**

NFS: Exemplo

Conteúdo parcial de /etc/fstab para máquinas CRC:

```
/usr1/mail@sga.pucminas.br:/var/spool/mail  
/users/users3@elsrv2:/users/users3  
/users/users4@elsrv2:/users/users4  
/rcf/mipsel/4.2/share@elsrv1:/exp/rcf/share  
/rcf/common@elsrv1:/exp/rcf/common
```

Teste na sua!

Estratégias de nomeação: Espaço global de nomes

- Espaço de nome único: Google File System (GFS) e Hadoop Distributed File System (HDFS)

- Independentemente do nó em que você esteja, os nomes dos arquivos são os mesmos.

- Conjunto de **clientes** de estação de trabalho e um conjunto de **máquinas** de servidor de arquivos dedicadas.

- Quando um **cliente** é **iniciado**, ele **obtem** sua **estrutura** de **nome** de **arquivo** de um **servidor**.

- À medida que os usuários acessam os arquivos, **o servidor envia cópias para a estação de trabalho** e a estação de trabalho armazena os arquivos em cache

Espaço global de nomes

• Vantagens:

- A nomenclatura é consistente e fácil de manter.
- O espaço de nome global garante que todos os arquivos sejam os mesmos, independentemente de onde você faça login.
- Como os nomes são vinculados tardiamente, é mais fácil movê-los.

• Desvantagens:

- É difícil para o sistema operacional manter o conteúdo do arquivo consistente devido ao armazenamento em cache.
- O espaço global de nomes pode limitar a ~~flexibilidade~~.
- Problemas de **desempenho** (podem ocorrer devido a **concorrência por recursos**).

Acesso remoto a arquivos e armazenamento em cache

Quando o usuário especifica um arquivo remoto, o sistema operacional pode fazer o acesso:

- 1. remotamente, no computador do servidor e, em seguida, retornar os resultados usando RPC (chamado de serviço remoto), ou*
- 2. pode transferir o arquivo (ou parte do arquivo) para o host solicitante e realizar acessos locais (chamadas à cache)*

Problemas de cache:

- Onde e quando os blocos de arquivos são armazenados em cache?
- Quando as modificações são propagadas de volta para o arquivo remoto?
- O que acontece se vários clientes armazenarem em cache o mesmo arquivo?

Acesso remoto a arquivos e armazenamento em cache

Localização

- *Disco local*

– Vantagens:

- **Tempo** de acesso reduzido.
- Mais **seguro** se o nó falhar.

– Desvantagens:

- É **difícil** manter a **cópia local consistente** com a cópia remota.
- Mais **lento** do que apenas mantê-lo na memória local.
- Requer que o **cliente tenha** um **disco**.

Acesso remoto a arquivos e armazenamento em cache

Localização

- *Memória local*

– Vantagens: Tempo de acesso rápido.

– Desvantagens:

- É difícil manter a cópia local **consistente** com a cópia remota.
- Não **tolera** bem a **falha** do nó.
- **Tamanho limitado** do **cache**.
- Funciona com estações de trabalho **sem disco**.

Políticas de atualização da cache

O **momento** de **gravar** as **alterações locais** no **servidor** tem uma função central na determinação do desempenho do sistema de arquivos distribuídos.

- **Write through**: produz os resultados mais confiáveis, pois cada gravação atinge o disco remoto antes que o processo continue, mas tem o desempenho mais fraco.

- O armazenamento em cache com gravação é equivalente ao uso do serviço remoto para todas as gravações e explora o armazenamento em cache somente para leituras.

- **Write back**: produz o tempo de resposta mais rápido, pois a gravação só precisa atingir o nível mais próximo da cache antes que o processo continue.

- Reduz o tráfego de rede e o número de gravações no disco para gravações repetidas no mesmo bloco de disco, já que apenas uma das gravações passará pela rede.

- Se o computador do usuário falhar, os dados não gravados serão perdidos.

Quando fazer? Write-back quando o arquivo é fechado, quando um bloco é removido do cache ou a cada 30 segundos.

Consistência do cache

Write-back pode gerar inconsistência. Como?

- Consistência iniciada pelo cliente: O **cliente** entra em contato com o servidor e **pergunta** se **sua cópia** é **consistente** com a cópia do servidor.
 - Pode verificar a cada acesso.
 - Pode verificar em um determinado intervalo.
 - Pode verificar somente ao abrir um arquivo.
- Consistência iniciada pelo servidor: O servidor detecta possíveis conflitos e invalida os caches
 - O servidor precisa saber:
 - quais clientes armazenaram em cache quais partes de quais arquivos.
 - quais clientes são leitores e quais são escritores.

Estado do servidor e replicação

- Servidor com estado (stateful) vs servidor sem estado (stateless)

- Com estado: Guarda estado entre requisições
- Sem estado: Não guarda nenhuma informação dos clientes e do estado dos arquivos entre uma requisição e outra.

– Trade-off entre desempenho e tolerância a falhas de colisão

- Replicação

- Os dados do servidor são replicados entre as máquinas
- Necessidade de garantir a consistência dos arquivos quando um arquivo é atualizado em um servidor

Resumo

- *Nomenclatura*
 - *Desejamos a completa independência do nome, mas é difícil alcançá-la 100%*
 - *Os nomes dependentes do local são os mais predominantes*
- *Aceleramos o acesso remoto a arquivos com o armazenamento em cache.*
- *Necessidade de gravar as alterações de volta no disco*

PERGUNTAS?

REFERÊNCIAS

- **TANENBAUM, Andrew.** Sistemas operacionais modernos.
- **SILBERSCHATZ, Abraham et al.** Fundamentos de sistemas operacionais: princípios básicos.
- **MACHADO, Francis; MAIA, Luiz Paulo.** Arquitetura de Sistemas Operacionais.
- **CARISSIMI, Alexandre et al.** Sistemas operacionais.