

Serviços em Redes de Computadores

Tecnologia em Redes de Computadores

Aula 03

Prof. Me. Henrique Martins

Aula 03

- **Sistemas de Arquivos Distribuídos**
- **NFS: Compartilhamento de Arquivos com NFS**

Definição (Coulouris)

“Um sistema de arquivo distribuídos permite ao programas armazenarem e acessarem arquivos remotos exatamente como se fossem locais, possibilitando que os usuários acessem arquivos a partir de qualquer computador em uma rede.”

Definição (Tanenbaum)

“Considerando que compartilhar é fundamental para sistemas distribuídos, não é surpresa que sistemas de arquivos distribuídos sejam a base para muitas aplicações distribuídas. Sistemas de arquivos distribuídos permitem que vários processos compartilhem dados por longos períodos, de modo seguro e confiável.”

Introdução

- Um **Sistema de Arquivos Distribuído (SAD)** tem o objetivo de fornecer:
 - Os mesmos serviços e recursos de um sistema de arquivos convencional, (na visão dos clientes que o acessa), com a diferença que o SAD pode ser acessado de qualquer máquina dentro de uma rede (acesso remoto).
- Alguns SADs implementam certas **transparências** para os usuários, que não precisam saber que estão lidando com um sistema de arquivos de um determinado tipo ou natureza.

Transparências

- **Nome:** O nome do recurso a ser utilizado (como um arquivo) não deve indicar ou conter indícios de onde está localizado.
- **Localização:** O usuário não precisa fornecer a localização física do recurso (no caso um arquivo) para encontrá-lo.
- **Acesso:** O usuário não perceberá se o arquivo que está sendo usado é local ou remoto.

Transparências

- **Replicação:** Os arquivos do SAD podem ter cópias armazenadas em locais diferentes. O usuário não deve perceber que existem várias cópias do mesmo arquivo. Para ele só será apresentada uma, e quem a escolherá é o SAD.
- **Concorrência ou Paralelismo:** Vários usuários podem acessar o mesmo arquivo ao mesmo tempo, mas isso não deve afetar esses usuários nem outros.
- **Falha:** O SAD deve garantir que o acesso aos arquivos seja ininterrupto e sem falhas, sem que o usuário saiba como isso é tratado.

Motivação

- A velocidade de acesso a arquivos não cresce na mesma proporção que a velocidade dos processadores ou a velocidade de acesso a memória, o que torna o sistema de arquivos um gargalo para as aplicações.
- Esse tipo de pesquisa envolve atualmente várias empresas, como:
 - IBM (com o GPFS)
 - Sun (com o NFS)
 - Ericson (com o PVFS)
 - Institutos de pesquisa acadêmica
 - Universidade de Minnesota (com o GFS)
 - Universidade da Califórnia (com o RAMA)
 - Universidade Nacional de Chiao Tung, China, (com o Pasda)

Serviços Oferecidos pelos SADs

- Serviços mais importantes oferecidos pelos sistemas de arquivos distribuídos:
 - Serviço de Nomes Distribuído
 - Serviço de Arquivos Distribuído
 - Serviço de Diretórios Distribuído

Serviço de Nomes Distribuído

- O serviço de nomes cuida de indicar a localização de um determinado arquivo dado o seu nome ou caminho.
- Se a localização do arquivo estiver armazenada no nome dele, como por exemplo `\\servidor\diretorio\arquivo.txt`, então esse serviço de nomes não provê transparência de localização.
- Para prover essa transparência, o nome ou caminho de um arquivo não deve ter indícios de sua localização física, e caso esse arquivo mude de lugar, ou tenha várias cópias, o seu nome ou caminho não precisará ser alterado.
- O serviço de nomes precisa oferecer ou **resolução por nomes**, ou **resolução por localização**, ou ambos.

Serviço de Arquivos Distribuído

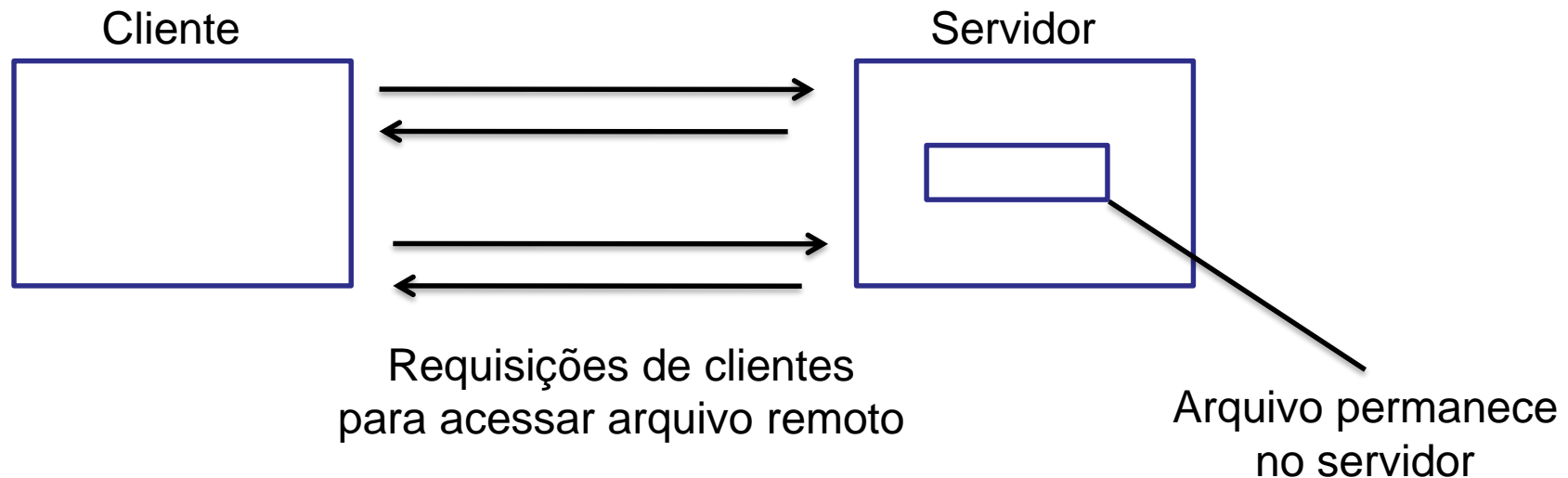
- Serviço de arquivos é responsável por fornecer operações sobre os arquivos que compõe o sistema.
- Os arquivos podem ser armazenados de diferentes formas, dependendo do seu tipo e uso. Por exemplo, arquivos que compõe um banco de dados podem ser armazenados em formato de registros.
- Esse serviço também cuida das propriedades dos arquivos, que não são armazenadas dentro dos mesmos, como data de criação, data de alteração, tamanho, dono do arquivo, permissões de leitura, escrita e execução, além de qualquer outra informação relevante.

Serviço de Arquivos Distribuído

- É de responsabilidade desse serviço também cuidar para manter a integridade das operações realizadas nos arquivos. Por exemplo, quando uma aplicação altera algum arquivo, todas as demais aplicações que estiverem acessando-o devem perceber essa alteração o mais rápido possível.
- Existem dois tipos de implementação para um serviço de arquivos:
 - **acesso remoto**
 - **cópia remota**

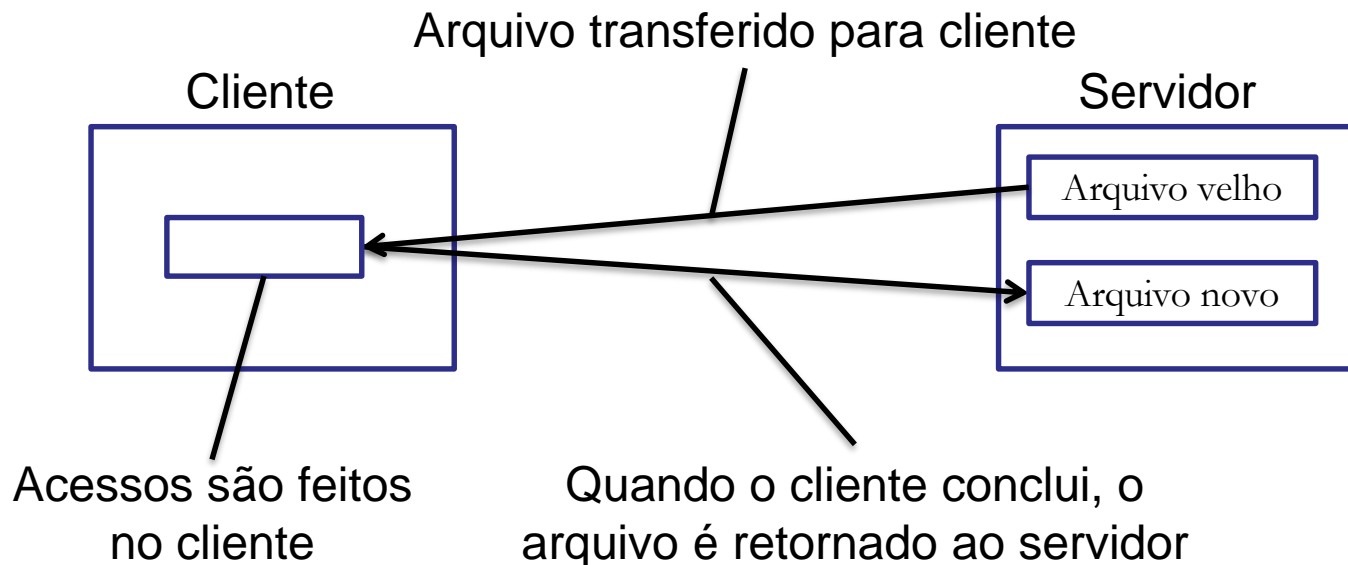
Serviço de Arquivos Distribuído

- No caso do **acesso remoto**, o cliente não possui um espaço para guardar os arquivos que estiver usando, e toda e qualquer operação realizada com os arquivos serão sempre através da rede. Isso pode deixar o sistema muito lento, já que depende da velocidade da rede.



Serviço de Arquivos Distribuído

- Já no caso da **cópia remota**, o cliente recebe uma cópia do arquivo para trabalhar, e depois quando tiver terminado ele o devolve com as alterações para o servidor. Isso só funciona se o cliente tiver espaço suficiente para armazenar o arquivo. A velocidade da rede só influenciará durante as transmissões do arquivo de um lado para outro.



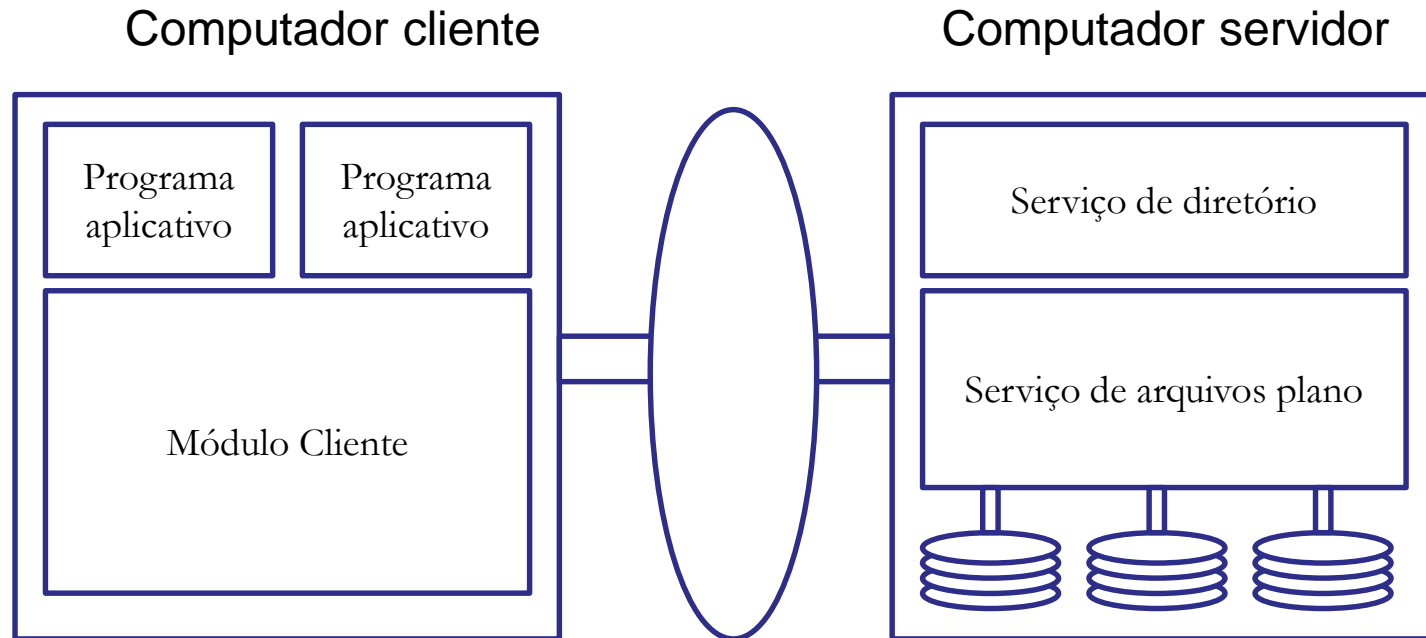
Serviço de Diretórios Distribuído

- Esse serviço é responsável por manter a organização dos arquivos armazenados no sistema. Ele fornece uma interface para que os usuários possam arranjar seus arquivos num formato hierárquico, que é estruturado em diretórios e subdiretórios. Um subdiretório só pode ter um único pai.
- Algumas das operações sobre diretórios oferecidas pelos serviços de diretórios são: criação, remoção, alteração, listagem, alteração de permissões, etc.

Arquitetura do serviço de arquivos

- Uma arquitetura que apresenta uma separação clara das principais preocupações no fornecimento de acesso aos arquivos é obtida por meio da estruturação do serviço de arquivo em três componentes:
 - Serviço de arquivos plano
 - Serviço de diretório
 - Módulo cliente

Arquitetura do serviço de arquivos



Arquitetura do serviço de arquivos

- **Serviço de arquivos plano:** se preocupa com a implementação de operações sobre o conteúdo dos arquivos.
- São usados *identificadores exclusivos de arquivos (UFIDs – Unique File IDentifires)* para fazer referência aos arquivos em todas as requisições de operações ao serviço de arquivo plano.

Arquitetura do serviço de arquivos

- **Serviço de diretório:** fornece um mapeamento entre nomes textuais de arquivos e seus UFIDs.
- Os clientes podem obter o UFIDs de um arquivo citando seu nome textual para o serviço de diretório.

Arquitetura do serviço de arquivos

- **Módulo cliente:** É executado em cada computador cliente, integrando e estendendo as operações do serviço de arquivos plano e do serviço de diretório sob uma interface de programação de aplicativo única, disponível para programas em nível de usuário nos computadores clientes.

Características dos SADs

- Qual sistema de arquivos usar em um sistema distribuído?
- Para resolver esse tipo de questão, deve-se saber qual será a aplicação que rodará nesse sistema, e a partir disso tentar descobrir o que é mais importante para o sistema, como:
 - Tolerância a Falhas
 - Acesso Concorrente
 - Replicação de Arquivos
 - Heterogeneidade
 - Segurança
 - Eficiência

Tolerância a Falhas

- Se um servidor cair ou ficar fora do ar ou da rede, o sistema de arquivos não pode perder informações e nem ficar indisponível total ou parcialmente.
- Além disso, os usuário não precisam saber como isso foi implementado e nem como funciona.
- Eles simplesmente requisitam um arquivo, e se algum servidor cair, o arquivo não pode ficar indisponível.
- Para isso é necessário implementar transparência a falhas.

Acesso Concorrente

- Vários usuários podem acessar vários arquivos, ou os mesmos arquivos, sem sofrer danos, perda de performance ou quaisquer outras restrições.
- Isso tudo deve ocorrer sem que o usuário precise saber como é realizado pelos servidores.
- Assim, é necessário haver transparência de concorrência e de paralelismo.

Replicação de Arquivos

- Se um sistema de arquivos suporta essa funcionalidade, a confiança e a eficiência do serviço de arquivos é generosamente aumentada. Eficiência em termos de tempo de resposta, carga do servidor e tráfego de rede. Adicionalmente, se um determinado servidor cair, o serviço de arquivos ainda pode realizar suas obrigações.
- Assim, replicação de arquivos provém tolerância a falhas, já que o usuário nem sequer percebe que o servidor que ele estava usando caiu e que outro (ou outros) entrou no lugar pra prover o arquivo que ele estava usando.

Heterogeneidade

- As interfaces de serviços devem ser definidas de modo que o software cliente e servidor possa ser implementado para diferentes sistemas operacionais e computadores.
- Esse requisito é um aspecto importante de sistemas abertos.

Segurança

- Praticamente todos os sistemas de arquivos fornecem mecanismos de controle de acesso baseado no uso de listas de controle de acesso.
- Nos SAD's, existe a necessidade de autenticar as requisições dos clientes para que o controle de acesso no servidor seja baseado nas identidades corretas do usuário e para proteger o conteúdo das mensagens de requisições e respostas.

Eficiência

- Um serviço de arquivo distribuído deve oferecer recursos que tenham pelo menos o mesmo poder e generalidade daqueles encontrados nos sistemas de arquivos convencionais, e deve obter um nível de desempenho comparável.

Eficiência

- Birrell e Needham (1980) expressaram seus objetivos de projeto para o CFS (Cambridge File Server) nos seguintes termos:

“Desejaríamos ter um servidor de arquivos simples, de baixo nível, para compartilhar um recurso dispendioso, a saber, o disco, que nos deixasse livre para projetar um sistema de arquivos mais apropriado para um cliente em particular, ao mesmo tempo em que disponibilizasse um sistema de alto nível compartilhado entre os clientes.”

Exemplos de SAD's

- NFS
- AFS
- CODA
- SPRITE
- PVFS
- GFS

NFS - *Networking File System*

- Sistema de arquivos distribuído desenvolvido inicialmente pela Sun e um dos mais usados em sistemas Unix.
- Em 1985 a Sun tornou publico o protocolo do NFS, o que permitiu que outras empresas e desenvolvedores pudessem criar clientes e servidores NFS.
- Hoje em dia já é possível encontrar implementações do NFS (tanto cliente como servidor) para quase todos os sistemas operacionais existentes, inclusive sistemas não-UNIX (como o Windows).

AFS - *ANDREW File System*

- Começou na Universidade de Carnegie Mellon em 1983, com suporte provido pela IBM.
- Seu objetivo era projetar e implementar um sistema de arquivos distribuído para o ambiente acadêmico que permitiria compartilhar uma estrutura comum de diretórios entre milhares de clientes.

CODA - *Constant Data Availability*

- SAD desenvolvido no início dos anos noventa, é um descendente do **AFS**.
- Seu principal objetivo é fornecer acesso a sistemas de arquivo distribuídos para computadores portáteis.
- Ele implementa alguns mecanismos de replicação não presentes no AFS.

SPRITE - File System

- O SPRITE é um sistema operacional distribuído desenvolvido pela Universidade da Califórnia, em Berkeley. Embora o seu design seja monolítico, ele teve muito sucesso no sistema de arquivos que emprega.
- O SPRITE File System é muito similar ao NFS, por manter o modelo UNIX de arquivos e suas operações.

PVFS - *Parallel Virtual File System*

- O *Parallel Virtual File System* é um sistema de arquivos distribuído desenvolvido para prover alta performance e escalabilidade paralela para clusters de PCs linux.

GFS - *Global File System*

- *Global File System* é um Sistema de arquivos implementado no sistema operacional Silicon Graphics IRIX, que pode ser acessado usando-se comandos e utilitários comuns do UNIX.
- A idéia principal é compartilhar fisicamente o sistema de armazenamento de dados.

Pontos Importantes

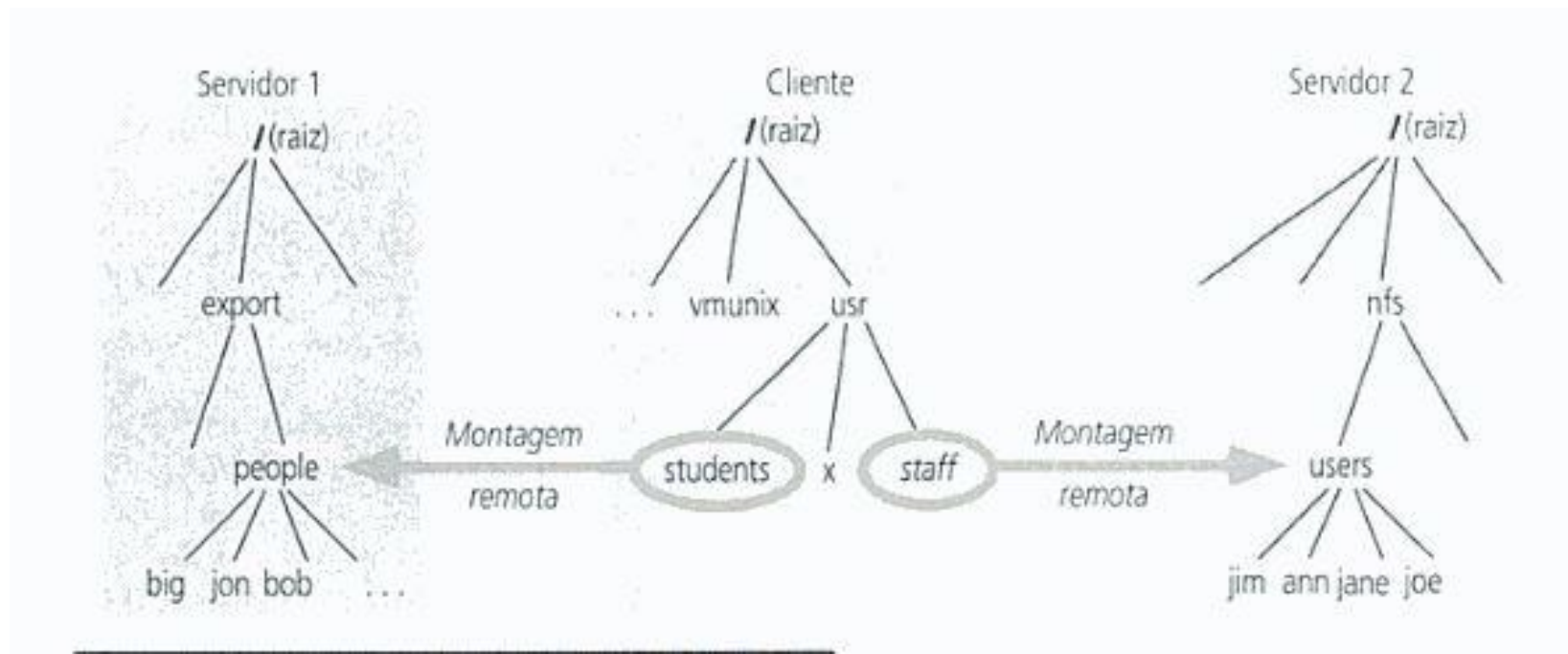
- Serviço de montagem
- Sincronização
- Consistência e Replicação
- Segurança

Serviço de montagem

- A montagem de subárvores de sistemas de arquivos (*filesystem*) remotos feita por clientes é suportada por um processo separado de serviço de montagem, executando em nível de usuário em cada computador servidor.
- Em cada servidor existe um arquivo com um nome conhecido, contendo os nomes dos sistemas de arquivos locais que estão disponíveis para montagem remota.

Serviço de montagem

- O sistema de arquivos montado em /usr/students no cliente é, na verdade, a subárvore localizada em /export/people no Servidor 1; o sistema de arquivos montado em /usr/staff no cliente é, na verdade, a subárvore localizada em /nfs/users no Servidor2.

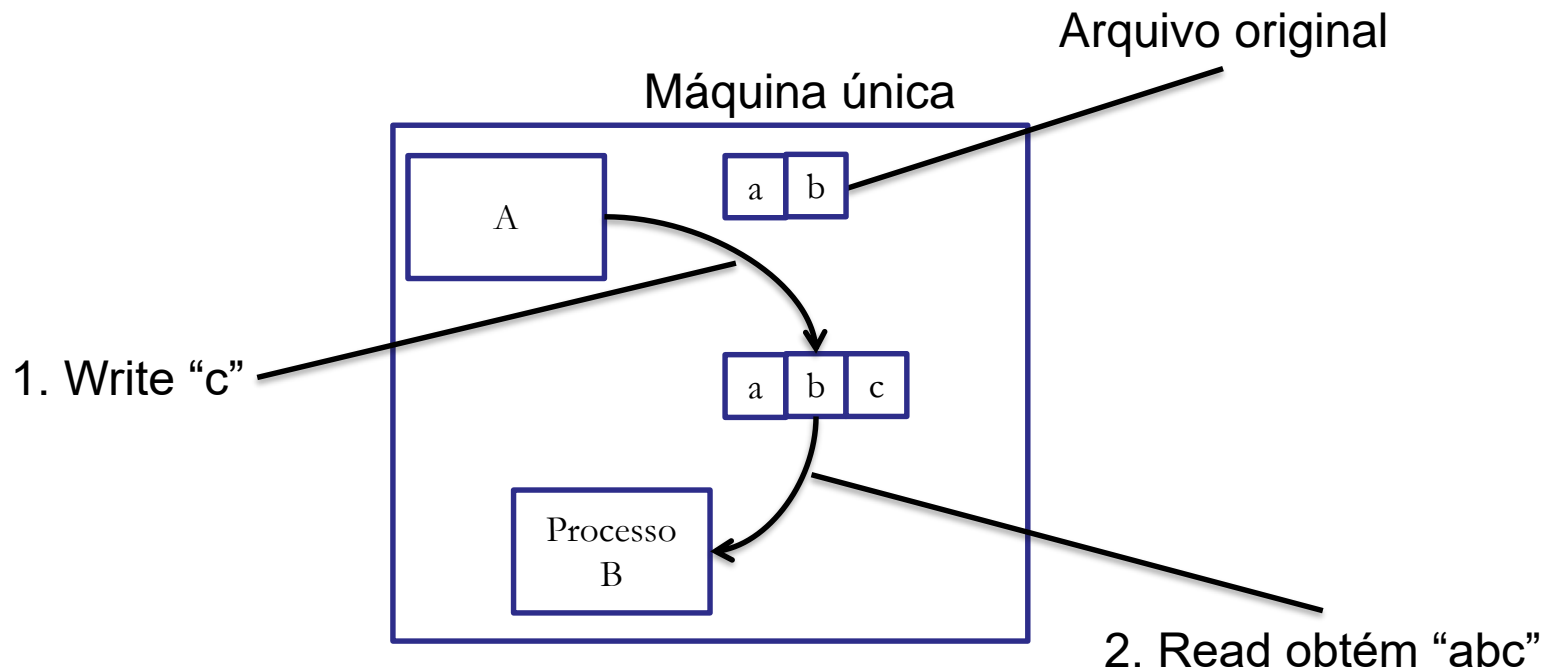


Sincronização

- A sincronização para sistemas de arquivos não seria um item a ser comentado se os arquivos não fossem compartilhados.
- Portanto quando dois ou mais usuários compartilham o mesmo arquivo ao mesmo tempo, é necessário definir com exatidão a semântica de leitura e de escrita para evitar problemas.

Sincronização

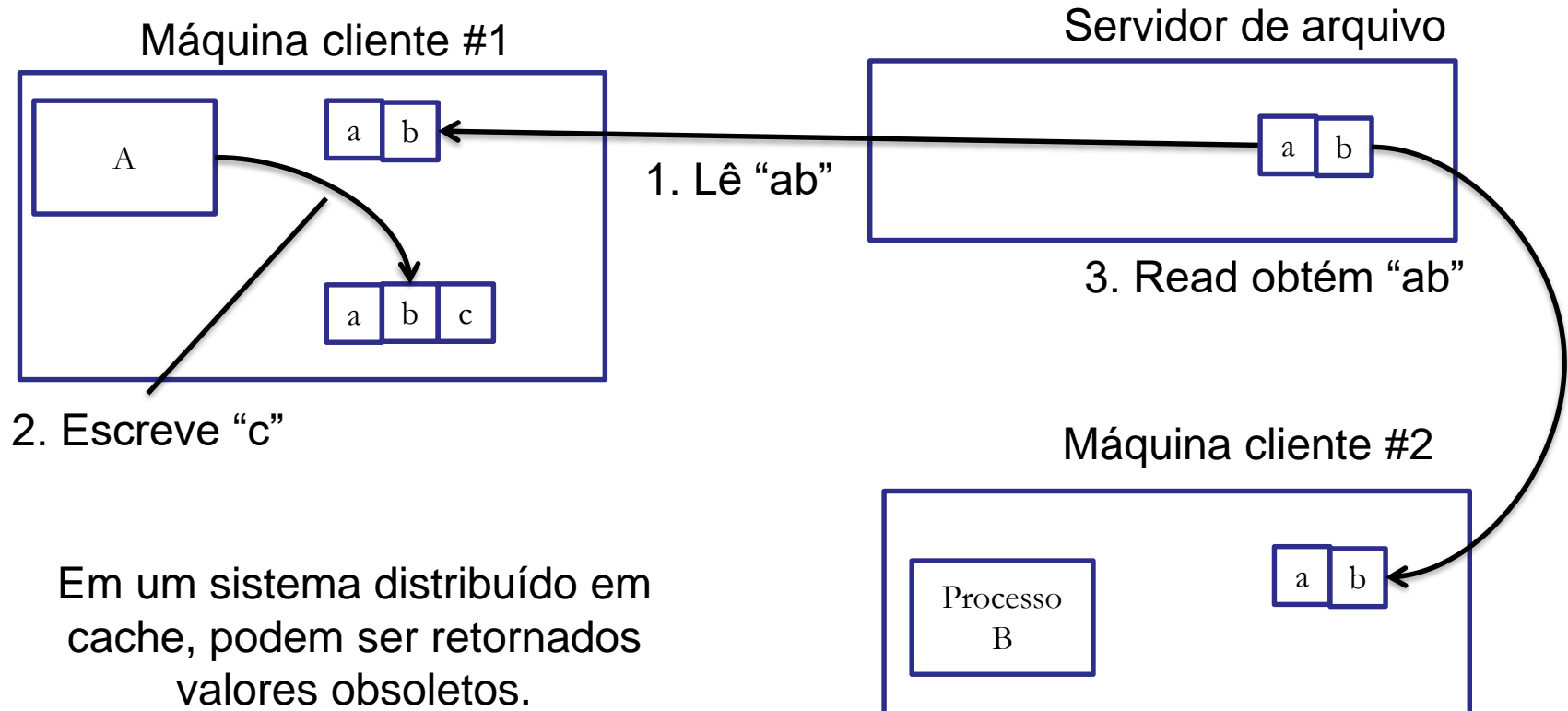
- Quando uma operação **read** vem depois de uma operação **write**, a **read** retorna valor que acabou de ser escrito.



Em um processador único, quando uma operação de leitura vem depois de uma operação de escrita, o valor retornado pela leitura é o valor que acabou de ser escrito.

Sincronização

- Quando duas operações **write** ocorrem em rápida sucessão, seguidas por um **read**, o valor lido é o valor armazenado pela última **write**.



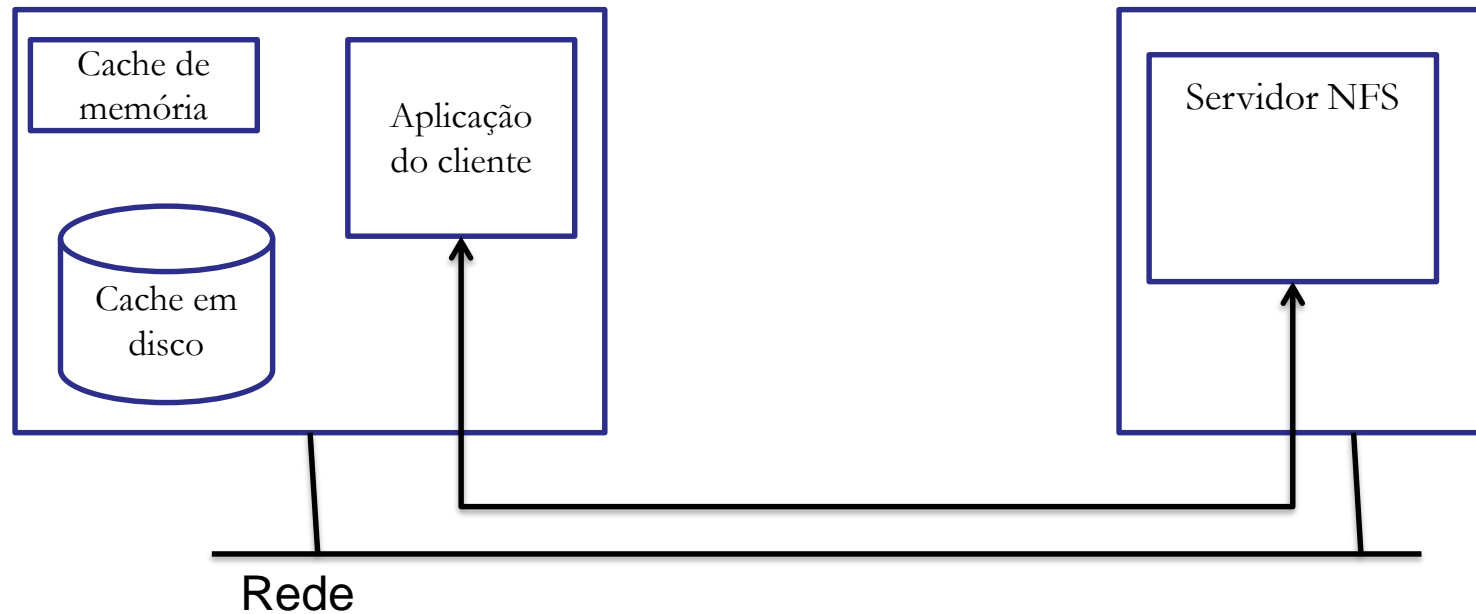
Consistência e Replicação

- Cache e replicação desempenham importante papel em sistemas de arquivos distribuídos, em particular quando são projetadas para funcionar em redes de longa distancia.
 - Cache do lado do cliente
 - Replicação do lado do servidor

Cache do lado do cliente

- Cache em NFS
 - Cada cliente pode ter uma cache de memória que contém dados lidos anteriormente de um servidor.
 - Além disso, também pode haver uma cache em disco que é adicionada como uma extensão à cache de memória, usando os mesmos parâmetros de consistência.

Cache do lado do cliente



Cache do lado do cliente

- Quando um cliente abre um arquivo e faz cache dos dados que obtém do servidor como resultado de várias operações **read**.
- Além disso, as operações **write** também podem ser executadas na cache.
- Quando o cliente fecha o arquivo, o NFS requer que, se ocorreram modificações, os dados sejam descarregados de volta para o servidor.

Replicação do lado do servidor

- Ao contrario do cache do lado do cliente, a replicação do lado do servidor em sistemas de arquivos distribuídos é menos comum.
- A replicação é aplicada quando a disponibilidade estiver em jogo; porém, da perspectiva de desempenho, faz mais sentido disponibilizar caches nas quais o arquivo inteiro (ou grande parte deles) é disponibilizado localmente para um cliente.

Replicação do lado do servidor

- Um outro problema da replicação do lado do servidor para conseguir desempenho é que, na verdade, a combinação do alto grau de replicação com uma baixa razão leitura/escrita pode degradar o desempenho.
- Como por exemplo: **toda operação de atualização precisa ser realizada em todas as réplicas.**
- Lembrando que as atualizações concorrentes precisam ser sincronizadas, o que resulta em mais comunicação e **maior redução de desempenho.**

Segurança

- Segurança em sistemas de arquivos distribuídos organizados segundo uma arquitetura cliente-servidor significa que todos os servidores devem manipular autenticação em controle de acesso.
- A idéia básica é que um sistema de arquivo remoto deve ser apresentado aos clientes como se fosse um sistema local de arquivos.

Segurança

- Quando se fala em segurança, falamos de comunicação segura onde deve ser estabelecido um canal seguro entre o cliente e o servidor.
- Sempre será necessário controlar acesso a arquivos que são manipulados por meio de atributos de arquivos de controle de acesso.
- Um servidor de arquivos é encarregado de verificar os direitos de acesso de seus cliente.

Exemplo Atual

- Cloud Computing



Referências

Sistemas Distribuídos: Conceitos e Projeto. COULOURIS, G; et al.

Sistemas Distribuídos: princípios e paradigmas. TANENBAUM, A. S;
STEEN, M. van

NFS (*network file system*) - <http://nfs.sourceforge.net/>

Material de Apoio

Sistemas de Arquivos Distribuídos -

<http://www.ime.usp.br/~carvalho/monografia-sad/monografia.html>

NFS

- O NFS é quase transparente para o usuário e é “sem estado”, o que significa que nenhuma informação é perdida quando um servidor NFS trava. Os clientes podem simplesmente aguardar o servidor retornar e depois continuar como se nada tivesse acontecido.

NFS: protocolos

- O NFS executa sobre o protocolo RPC (Remote Procedure Call) da Sun, que define uma maneira independente do sistema para os processos se comunicarem através de uma rede.
- Essa arquitetura permite que o NFS utilize o TCP ou UDP como protocolo de transporte.
- Apesar do NFS fazer a contagem de pacotes, não possui controle de congestionamento, nesse caso deve-se avaliar a possibilidade de utilizar o TCP. Essa opção é definida em **/etc/fstab**.

NFS: segurança

- O NFS implementa segurança através de um arquivo /etc/exports, onde são colocados os clientes confiáveis.
- É uma falha de segurança relativa, pois os clientes podem mentir sobre suas identidades.
- Mesmo controlando com os UIDs e GIDs de cada cliente, um usuário com acesso a root, pode alterar o UID intencionalmente para abrir um arquivo de outra pessoa.

NFS: segurança

- Caso tenha um firewall instalado, deve-se bloquear o acesso às portas 2049 TCP e UDP, que são utilizadas pelo NFS.
- Também é necessário o bloqueio da porta TCP e UDP 111 referente ao daemon **portmap**.
- Os sistemas NFS não devem ser exportados para máquinas não-locais.

NFS: daemons

Para que os clientes possam acessar o servidor NFS é necessário que os seguintes daemons estejam executando:

nfsd - daemon NFS, que atende requisições dos clientes NFS.

mountd - daemon de montagem NFS, que executa as solicitações que o nfsd lhe passa.

portmap - daemon portmapper, permite que clientes NFS descubram qual porta o servidor NFS está utilizando.

NFS: instalação

Instalação no servidor:

```
# apt-get install portmap nfs-kernel-server
```

Instalação no cliente:

```
# apt-get install nfs-common
```

NFS: configuração

Editar o arquivo `/etc/exports` para definir a pasta a ser compartilhada e as permissões de acesso.

# Path do diretório	IP do cliente	Permissões de acesso
/diretório/	192.168.0.0/24	(rw,no_root_squash)

Permissões de acesso:

ro: somente leitura.

rw: leitura e escrita.

no_root_squash: permite acesso normal por parte de root (não usar).

root_squash: é o padrão, mapeia UID 0 e GID 0 para os valores especificados por anonuid e anongid.

NFS: configuração

Editar o arquivo `/etc/hosts.deny` para definir a segurança do compartilhamento.

```
portmap: ALL  
lockd: ALL  
mountd: ALL  
rquotad: ALL
```

Editar o arquivo `/etc/hosts.allow` para definir o IP ou faixa de IPs que terão acesso a cada serviço.

```
portmap: 192.168.0.0/24  
lockd: 192.168.0.0/24  
rquotad 192.168.0.0/24  
mountd: 192.168.0.0/24  
statd: 192.168.0.0/24
```

NFS: iniciando o servidor

```
# /etc/init.d/nfs-kernel-server start/stop/restart
```

Carregar novamente o export após inserção de um novo membro:

```
# exportfs
```

NFS: ponto de montagem

Editar o arquivo /etc/fstab no cliente:

# Caminho do servidor	Ponto de montagem	FS	Opções	
192.168.0.1:/Compartilhado	/mnt/Compartilhado	nfs	rw	0 0

Aula prática

- Instalar o NFS no Linux
- Implementação de um DFS em um ambiente do Windows 2003 Server

Bibliografia

- KUROSE, James F., ROSS, Keith W. Redes de Computadores e a Internet: uma abordagem top down. São Paulo: Pearson, 2010.
- MORIMOTO, Carlos. Redes – Guia Prático. Porto Alegre: SUL Editores, 2009.
- NEMETH, Evi; SNYDER, Garth; TRENT, HEIN. Manual Completo do Linux – Guia do Administrador – 2ª ed. São Paulo: Pearson Prentice Hall, 2007.
- TANENBAUM, Andrew S. Redes de Computadores. Rio de Janeiro: Elsevier, 2003.