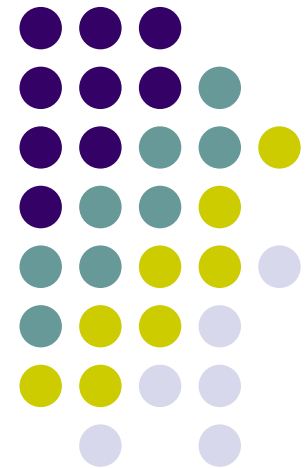


Sistemas de Arquivos Distribuídos

Universidade Federal do ABC
Prof. Dr. Francisco **Isidro** Massetto





Conceitos

- Dois tipos
 - Stateless
 - Statefull
- Statefull
 - Mantém informações de estado
 - Nome do arquivo
 - Ponteiro para arquivo
 - Posição atual

Conceitos



- Stateless
 - Cliente deve especificar todas as informações a cada acesso
 - Não mantém informações para acessos subsequentes

Benefícios



- Tolerância a falhas
 - Mais fácil de implementar em sistemas Stateless
- Desempenho
 - Sistemas statefull reduzem o número e tamanho da mensagem, otimizando o tráfego na rede



Características

- **Transparência**
 - Usuário vê o sistema de arquivos como um sistema único
 - Localização física dos arquivos fica oculta ao usuário
 - Garantia de transparência
 - Sistemas de nomes robusto



Características

- Escalabilidade
 - Novos computadores podem ser adicionados com facilidade
 - Duplicação de dados
 - Aumentando a escalabilidade → aumenta a concorrência
 - Largura de banda na rede de comunicação
 - Único servidor?? → gargalo!!!



Características

- Segurança
 - Acesso aos recursos
 - Permissão de acesso aos arquivos
 - Hierarquias de acesso
 - Proprietário – grupo – outros
 - Identidade dos usuários
 - Transmissão de dados
 - Criptografia das informações



Características

- Tolerância a falhas
 - Muito mais complexo
 - Falha no servidor
 - Cliente não pode ficar indefinidamente bloqueado
 - Timeout
 - Idempotência
 - Várias chamadas para executar a mesma operação
 - Em alguns casos é válida (consulta de saldos)
 - Outros casos podem gerar inconsistências
 - Soluções??
 - Cache
 - Identificação da mensagem



Características

- Consistência
 - Caching de cliente
 - Cópia local para evitar sobrecarga no servidor
 - Manter consistência
 - Invalidar versão do cliente
 - Timestamps
 - Se $T_{local} < T_{servidor} \rightarrow$ solicite versão do servidor
 - Travamentos (locking)
 - Acesso exclusivo a um arquivo por parte do cliente
 - Travas podem expirar
 - Clientes devem “renovar” travamentos
 - Replicações
 - Várias cópias que devem ser atualizadas

Em Resumo...



Transparência

Concorrência

Replicação

Heterogeneidade

Tolerância a Falhas

Consistência

Segurança

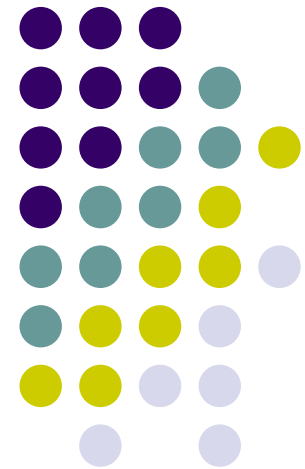
Eficiência

Eficiência

Objetivo de um Sistema de Arquivos Distribuídos é ter um desempenho comparado a um Sistema de Arquivos Local

Estudo de Caso

SUN NFS
(Network File System)



Visão Geral

- Características Gerais
- Arquitetura
- Protocolos
- Implementação
- Proteção de arquivos
- Transparência
- Performance
- Escalabilidade
- Conclusões



NFS: Características Gerais



- NFS é um sistema de arquivos distribuído que permite aos usuários acessar arquivos e diretórios localizados em computadores remotos e tratá-los como se fossem locais
- É possível usar comandos convencionais para criar, remover, ler (...) para manipular arquivos e diretórios remotos

NFS: Características Gerais



- Criado pela Sun em 1985
- Primeiro sistema de arquivos desenvolvido como produto
- Interfaces disponibilizadas para domínio público
- Tornou-se popular e amplamente utilizado na indústria e na academia

NFS: Características Gerais



- NFS pode ser utilizado em uma grande variedade de tipos de máquinas, sistemas operacionais e arquiteturas de rede. Essa independência é alcançada através da utilização de *Remote Procedure Call* (RPC)
- Entrega de mensagens: protocolo *at-least-once*
- Nível de transporte: UDP/IP (1987)

NFS: Características Gerais



- Máquinas possuem **diferentes** representações para um mesmo tipo de dados
- **XDR**: provê um método independente para a representação de dados a serem transmitidos na rede. É uma especificação para a representação padrão de vários tipos de dados

NFS: Arquitetura



- Na maioria dos casos: clientes e servidores na mesma rede local;
- Cada servidor **exporta** um ou mais de seus diretórios;
- Lista de diretórios que um servidor exporta: mantida no arquivo */etc/exports*;
- Clientes **montam** em seus sistemas de arquivos os diretórios exportados;

NFS: Arquitetura




E se a estação de trabalho não tem disco?

Um cliente sem disco pode montar um sistema de arquivo remoto no seu diretório raiz

Estações que possuem disco podem montar diretórios em qualquer lugar que desejarem...



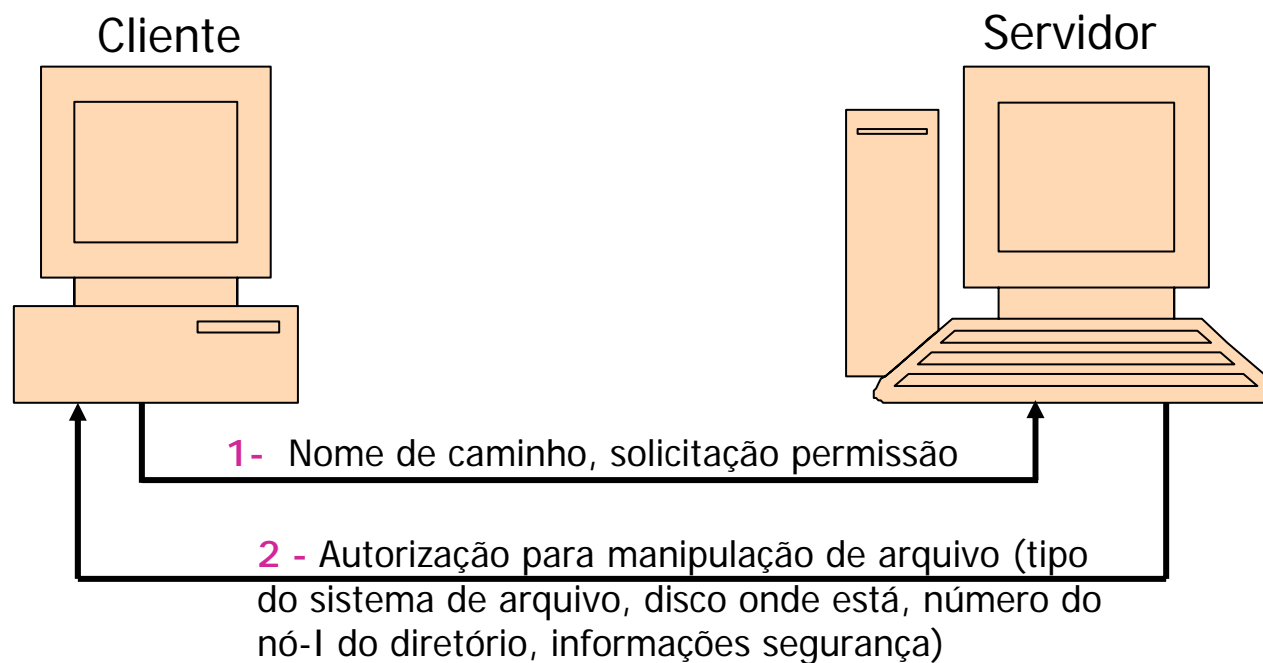
NFS: Protocolos

- Heterogeneidade  interface entre clientes e servidores bem definida
- Dois Protocolos:
 - Protocolo de Montagem dos arquivos
 - Protocolo para Acesso a Diretórios e a Arquivos



NFS: Protocolos

- Protocolo de Montagem dos arquivos





NFS: Protocolos

- Protocolo de Montagem dos arquivos
 - Montagem automática durante o *boot* da máquina cliente
 - Automontagem
 - Nenhum diretório remoto é montado durante o *boot* nem servidores são contactados
 - Montagem dinâmica de um sistema de arquivos
 - Tolerância a falhas
 - Melhoria na performance

Implementação do protocolo



- No servidor
 - Configurar diretórios que serão exportados
 - /etc/exports

```
/home 192.168.0.1 (rw)
/home 192.168.0.2 (rw)
/share 192.168.0.* (ro)
```

- Exportar diretórios
 - \$exportfs -a

Implementação do Protocolo



- No cliente
 - Montar sistema de arquivo
 - `$mount server:/dir /local/dir -t nfs`
 - **Mount** → aplicação que mapeia um dispositivo (neste caso, um sistema de arquivos em outro computador)
 - **server:/dir** → endereço do servidor que está exportando o sistema de arquivo e qual diretório será mapeado
 - **/local/dir** → “ponto de montagem”, diretório local que tornar-se-á acessível
 - **-t nfs** → especifica que o tipo de sistema de arquivo que será montado é remoto via protocolo NFS.

Implementação do Protocolo



- Configuração automática
 - Arquivo /etc/fstab
 - Configuração de montagem no momento do boot

```
server:/dir /local/dir nfs rw,noauto 0 0
```

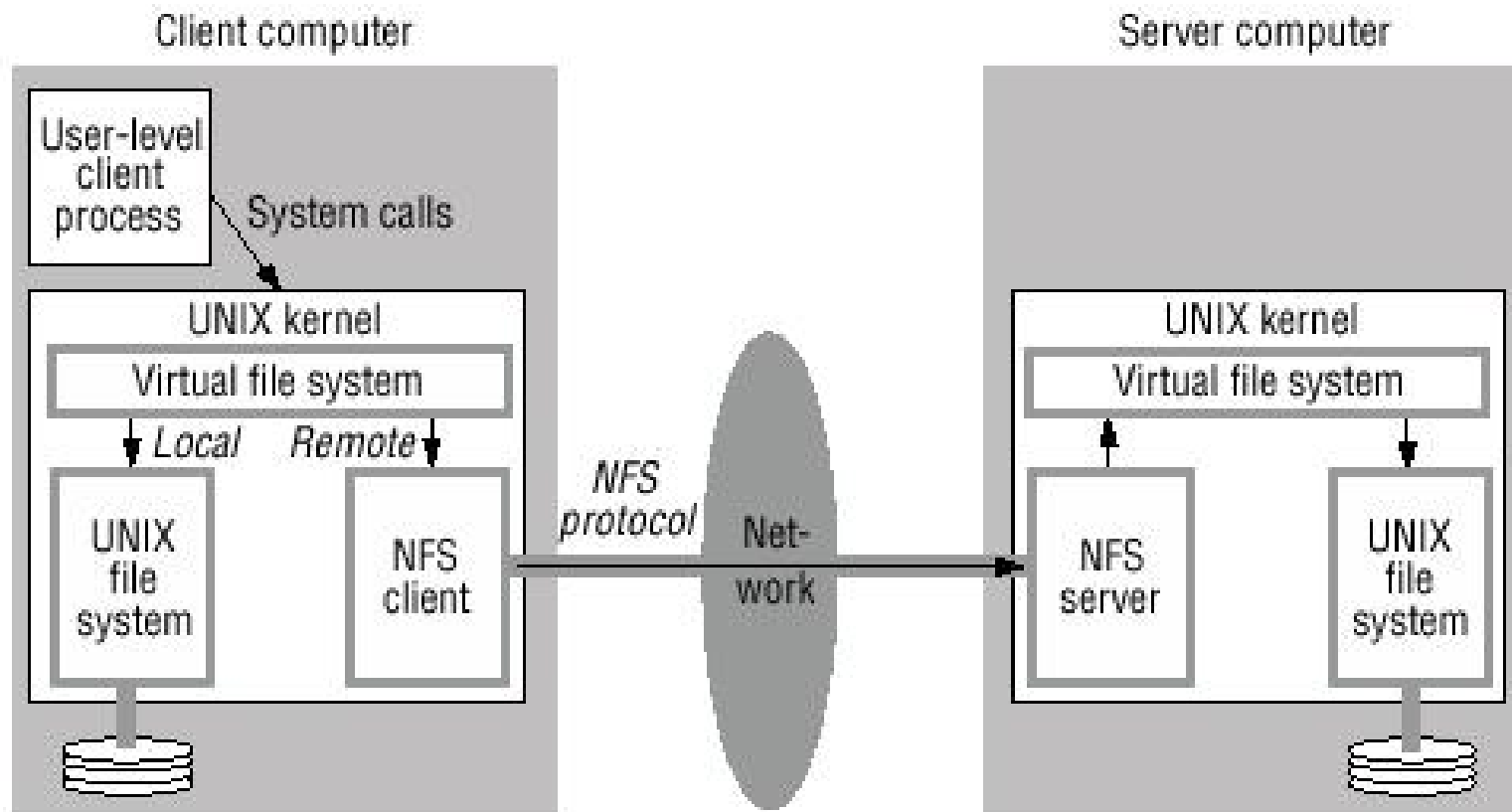
- Leitura do arquivo e montagem
- \$mount -a

NFS: Protocolos



- Protocolo para Acesso a Diretórios e a Arquivos
 - Clientes enviam mensagens aos servidores solicitando autorização para ler ou escrever em arquivos
 - Chamadas *open* e *close* não são aceitas
 - Não guarda informações sobre o estado dos arquivos abertos (*stateless*)
 - Problemas no tratamento a bloqueios de arquivos

NFS: Implementação





Proteção de Arquivos

- Utiliza o mecanismo de proteção do UNIX: bits *rwX* (read-write-execute)
- 3 níveis de proteção
 - Proprietário – grupo - outros
- Utilização de sistema de criptografia: validação do cliente e do servidor a cada pedido e resposta
- Chaves usadas na autenticação: mantidas pelo NIS (*Network Information Service*)



Proteção de Arquivos

- NIS (Network Information Service)
 - Armazena pares (chave, valor)
 - Cuida da criptografia das chaves
 - Mapeamento nome-usuário → senhas criptografadas
 - Mapeamento nome máquina → endereço da rede



NFS: Transparência

- **Transparência de Acesso:**

- Em clientes UNIX, o acesso a arquivos remotos é feito através dos comandos do UNIX

- **Transparência a Falhas:**

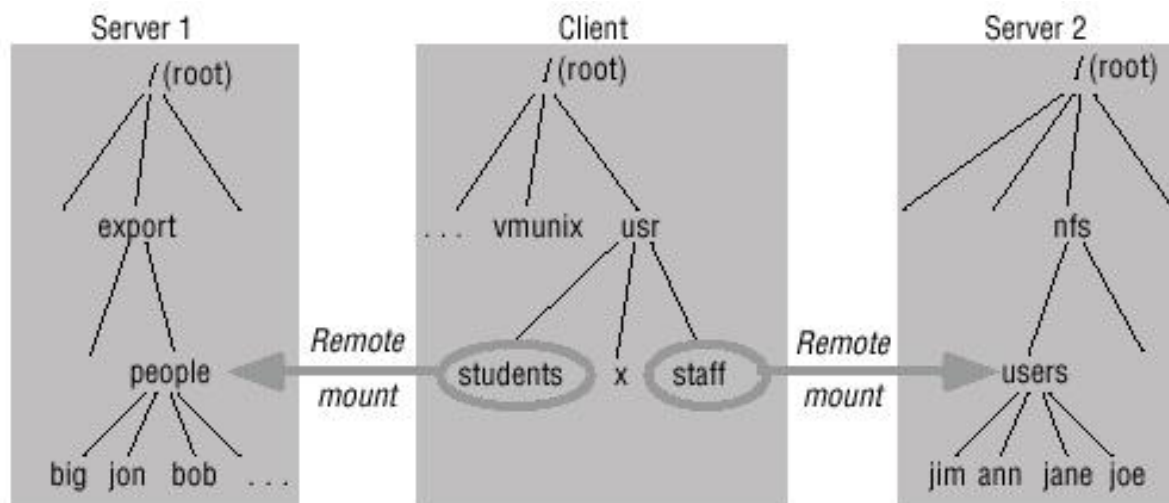
- Características importantes: serviço *stateless* e operações são *idempotentes*
- Se servidor falha → serviço é suspenso

Quando servidor retorna → processamento continua do ponto que havia sido interrompido.



NFS: Transparência

- **Transparência de Localização:**
 - Arquivos remotos podem estar associados a diferentes *pathnames* em diferentes clientes



NFS: Transparência



- **Transparência de Performance:**
 - Uso de *cache* no cliente e no servidor
 - Difícil manter coerência de *cache* dos módulos cliente



NFS: Transparência

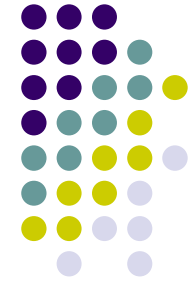
- **Transparência de Migração:**
 - Não é totalmente suportada pelo NFS
 - Quando um sistema de arquivos é movido de um servidor para outro, as tabelas remotas dos clientes precisam ser atualizadas
- NFS não suporta: Transparência de replicação, transparência de concorrência

NFS: Performance



- **Cache no servidor**
 - Sistema UNIX: arquivos, diretórios e atributos lidos do disco ➡ retidos no *cache*
 - Quando uma página é alterada ➡ novo conteúdo será escrito no disco apenas quando o *buffer* for requerido para outra página
 - Porém, para evitar perda de dados, as páginas alteradas são enviadas para o disco a cada 30 segundos

NFS: Performance



- ***Cache no servidor (NFS)***

Acesso para escrita



Modificações são enviadas imediatamente para o disco



Falhas no servidor resultariam em perdas de dados não detectadas pelos clientes

NFS: Performance



- ***Cache no Cliente***
 - **Objetivo:** reduzir o número de pedidos transmitidos ao servidor
 - Diferentes versões de arquivos podem existir em diferentes nós cliente
 - Método baseado em *timestamp* → validação de blocos no cache

NFS: Performance



- ***Cache no Cliente***

- Clientes mantêm no *cache* um *timestamp* indicando a hora que o arquivo foi modificado pela última vez no servidor
- Validação: sempre que um arquivo é aberto e sempre que o servidor é contactado para buscar um novo bloco de um arquivo ⇒ cliente requisita ao servidor o horário da última modificação e compara com seu *timestamp*

NFS: Performance



- ***Cache no Cliente***

Operação de Escrita



Páginas modificadas são marcadas



São enviadas para o servidor quando o arquivo é fechado ou de tempos em tempos.



NFS: Escalabilidade

- Limitada: projetado originalmente para permitir cada servidor suportar 5-10 clientes
- Número de clientes que podem acessar simultaneamente um arquivo compartilhado é limitado

Conclusões



- Objetivos:
 - Independência de Máquina e sistema operacional
 - Tolerância a Falhas
 - Transparência de Acesso
 - Performance
- Projeto do NFS:
 - Portabilidade
 - Heterogeneidade

Conclusões



- Outras implementações do NFS:
 - WebNFS
 - Solstice Network Client 3.2: MS Windows 3.x, Win95, Win98 e Windows NT



Bibliografia

- Coulouris et al. 1994 – *Distributed System: Concepts and Design, Second Edition*, Addison Wesley
- Tanenbaum, A. S. - *Distributed Operating Systems*, Prentice Hall Int. Ed., 1st Edition, 1995.
- Mullender, S. - *Distributed Systems*, Addison-Wesley Publishing Company, 2nd. Edition. 1993.
- *AIX Version 4.3 System Management Guide: Communications and Networks* (Chapter 10 – Network File System)
<http://nscp.upenn.edu/aix4.3html/aixbman/commadmn/toc.htm>
- Sandberg R. *The Sun Network Filesystem: Design, Implementation and Experience*. Proceedings of the Summer 1986 USENIX Technical Conference and Exhibition, 1986