



Universidade Tecnológica Federal do Paraná  
Departamento Acadêmico de Computação  
Bacharelado em Ciência da Computação

# Sistemas Distribuídos

## **Sistemas de Arquivos Distribuídos**

**Prof. Rodrigo Campiolo**

27/09/20

# Tópicos

---

- Introdução
- Serviço de arquivos
- NFS
- Semânticas de compartilhamento
- Política de atualização de cache
- Considerações finais
- Atividades

# Introdução

---

- Sistema de Arquivos Distribuídos (SAD)
  - Armazenar e acessar arquivos remotos como se estivessem locais.
  - Permite os usuários acessar arquivos de qualquer computador na intranet.
- Exemplos:
  - NFS – Sun Network File System
  - AFS – Andrew File System
  - CODA

# Requisitos SAD

---

- **Transparência**

- **Acesso:** acesso a arquivos locais e remotos com um único conjunto de operações.
- **Localização:** espaço de nomes de arquivos uniformes, independente do deslocamento entre servidores.
- **Mobilidade:** clientes e servidores não precisam ser alterados quando arquivos são movidos.
- **Desempenho:** clientes devem operar satisfatoriamente mesmo quando a carga varia.
- **Escala:** serviço pode ser expandido para lidar com acessos e carga.
- **Segurança e eficiência:** controle de acesso e desempenho similar a sistemas de arquivos locais.

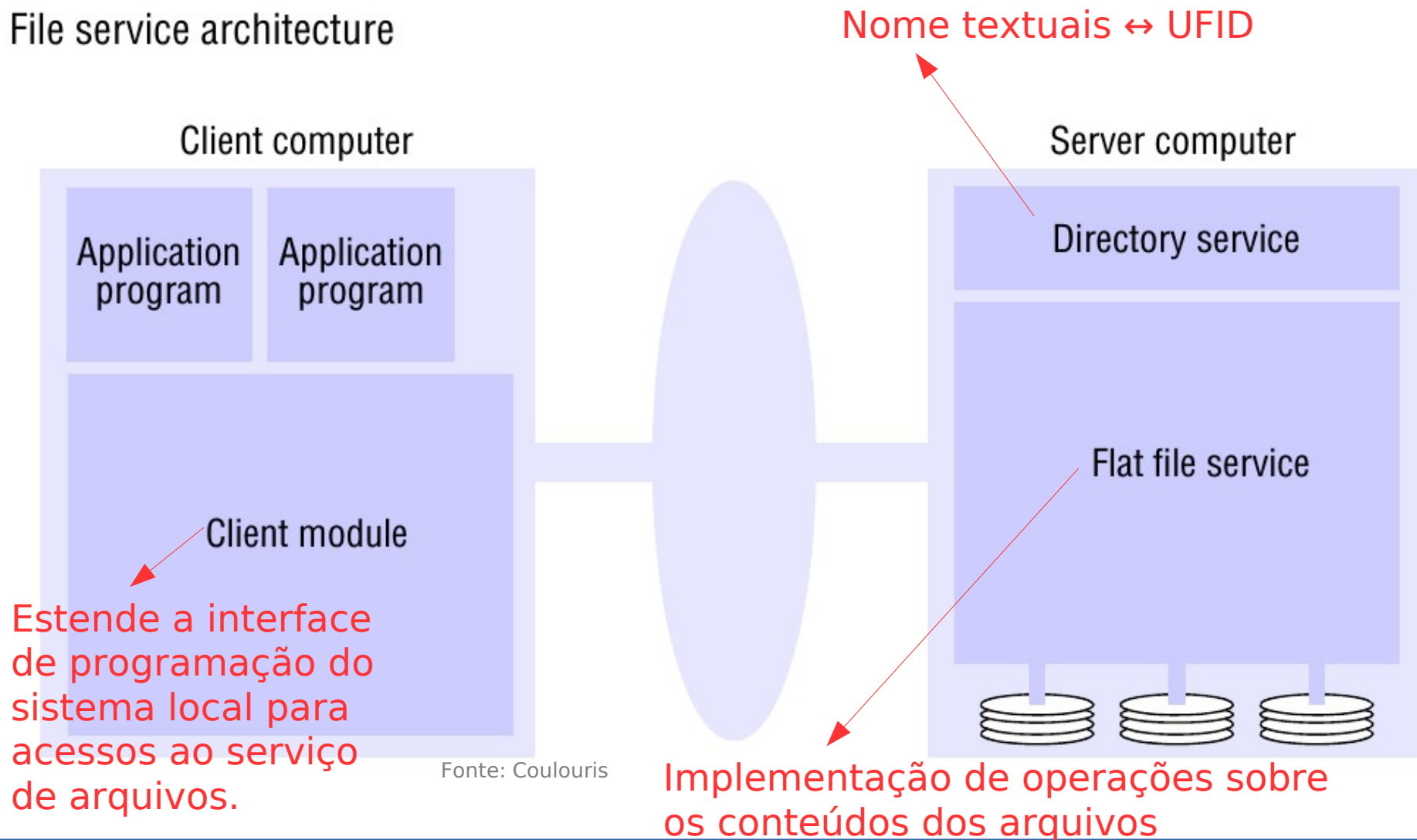
# Requisitos SAD

---

- **Atualizações concorrentes de arquivos:** alterações realizadas por um cliente não devem interferir nas operações de outros que usam o mesmo arquivo simultaneamente.
- **Replicação de arquivos:** cópias de um arquivo em várias localizações, que implica em desempenho e disponibilidade.
- **Heterogeneidade:** interfaces para suportar diferentes plataformas.
- **Tolerância a falhas:** continuar funcionar no caso de falhas de clientes e servidores. Lidar com falhas na comunicação.
- **Consistência:** estratégia para manter o arquivo consistente quando há acessos concorrentes.

# Serviço de arquivos

## File service architecture



# Serviço de arquivos

---

---

*Read(FileId, i, n) → Data*  
— gera *BadPosition*

Se  $1 \leq i \leq \text{Length}(\text{File})$ : lê uma sequência de até  $n$  elementos de um arquivo, começando no elemento  $i$ , e a retorna em *Data*. Gera uma exceção se o valor  $i$  é inválido.

*Write(FileId, i, Data)*  
— gera *BadPosition*

Se  $1 \leq i \leq \text{Length}(\text{File})+1$ : grava uma sequência de *Data* em um arquivo, começando no elemento  $i$ , ampliando o arquivo, se necessário. Gera uma exceção se o valor  $i$  é inválido.

*create() → FileId*

Cria um novo arquivo de tamanho zero e gera um UFID para ele.

*Delete(FileId)*

Remove o arquivo.

*GetAttributes(FileId) → Attr*

Retorna os atributos do arquivo.

*SetAttributes(FileId, Attr)*

Configura os atributos do arquivo (somente os atributos que não estão sombreados na Figura 12.3).

---

**Figura 12.6** Operações do serviço de arquivos plano.

Fonte: Coulouris

# Serviço de diretórios

---

---

*Lookup(Dir, Name) → FileId*  
— gera *NotFound*

Localiza o nome textual no diretório e retorna o UFID relevante.  
Se *Name* não estiver no diretório, gera uma exceção.

*AddName(Dir, Name, FileId)*  
— gera *NameDuplicate*

Se *Name* não estiver no diretório, adiciona (*Name, File*) no diretório e atualiza o registro de atributos do arquivo.  
Se *Name* já estiver no diretório, gera uma exceção.

*UnName(Dir, Name)*  
— gera *NotFound*

Se *Name* estiver no diretório, a entrada contendo *Name* é removida do diretório.  
Se *Name* não estiver no diretório, gera uma exceção.

*GetNames(Dir, Pattern) → NameSeq*

Retorna todos os nomes textuais presentes no diretório que correspondam à expressão regular *Pattern*.

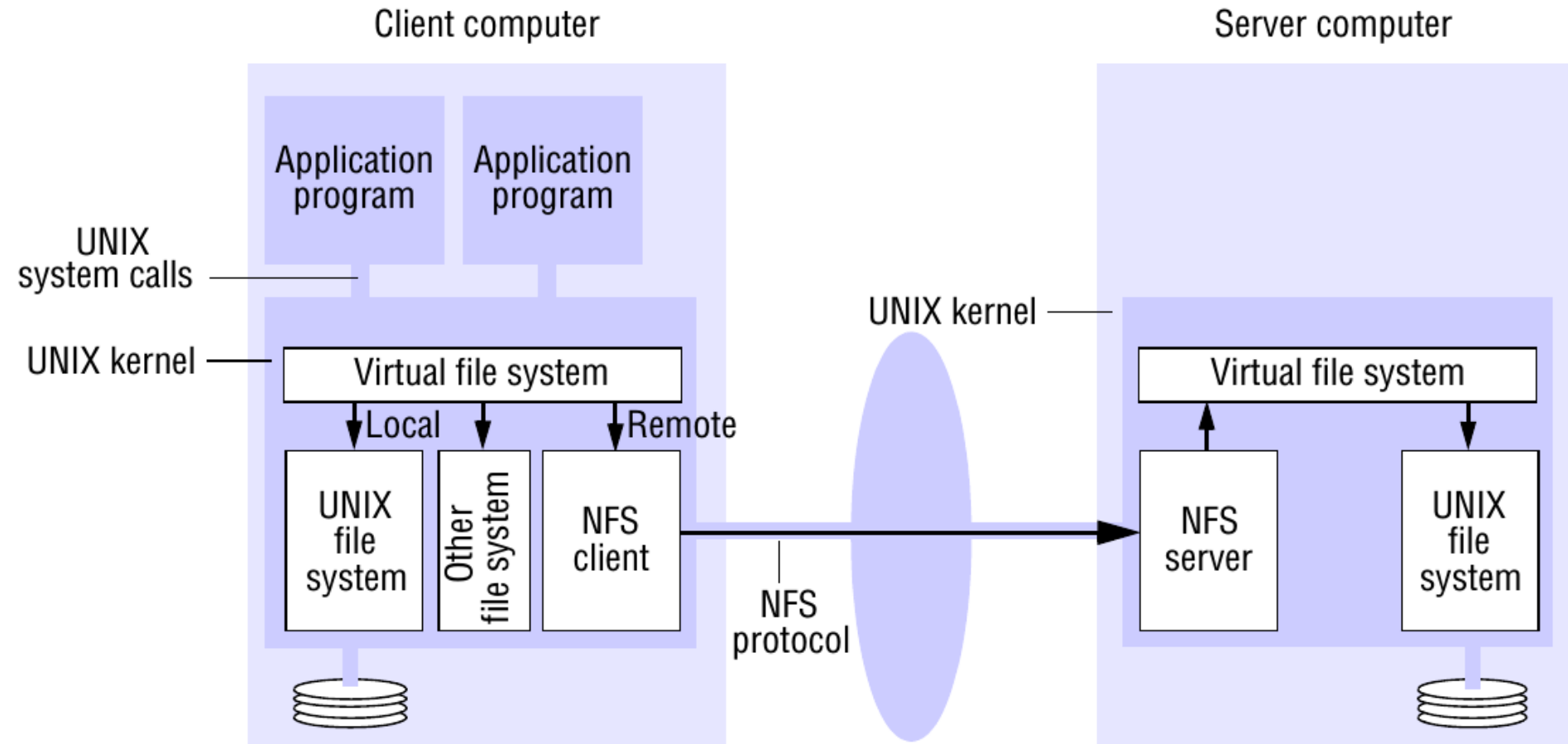
---

**Figura 12.7** Operações do serviço de diretório.

Fonte: Coulouris



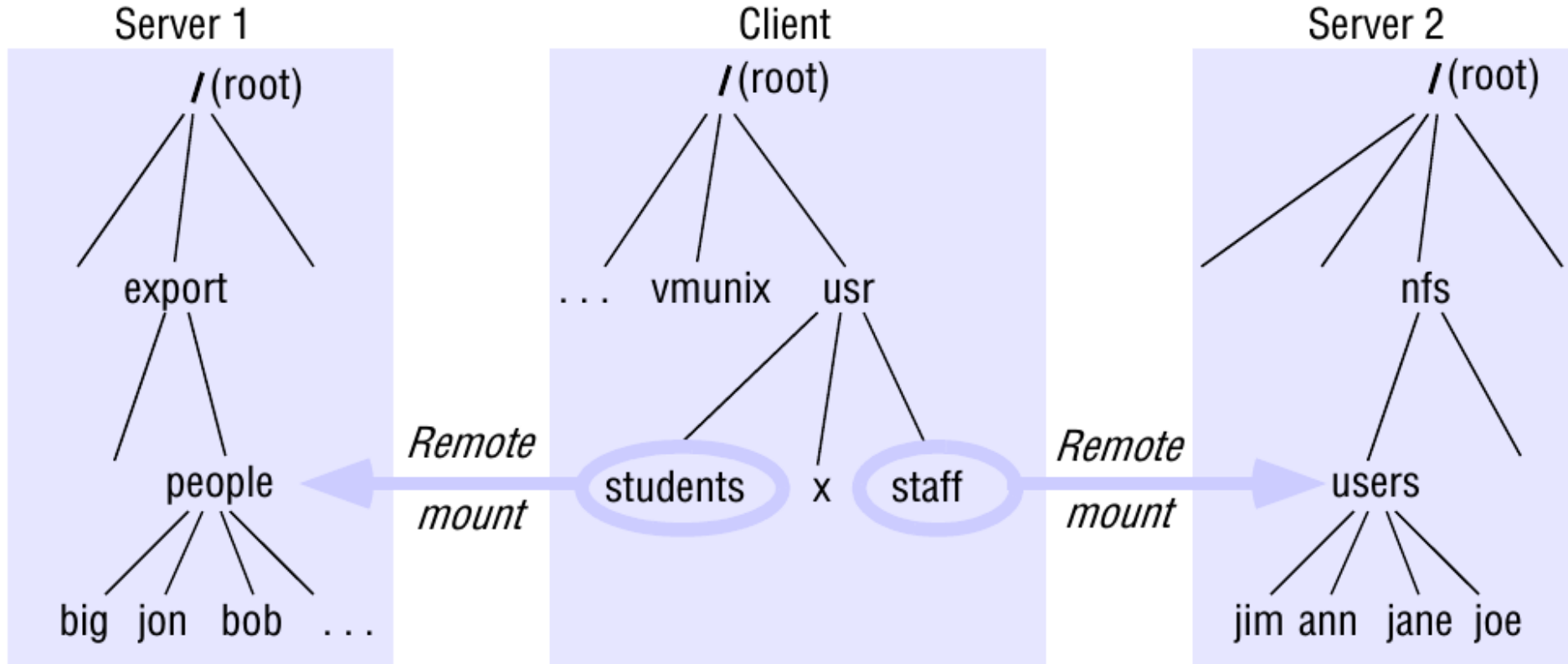
# NFS



**Figura:** Arquitetura do NFS.

Fonte: Coulouris

# NFS



**Figura:** Sistemas de arquivo local e remoto.

Fonte: Coulouris

# Semânticas de Compartilhamento

---

- One copy (UNIX)
  - Atualizações são escritas para uma única cópia e disponibilizadas imediatamente.
- Transações
  - Trava de arquivos (lock): compartilha para leitura e a escrita é exclusiva.
- Sessão
  - Cópia do arquivo ao abrir, trabalha localmente e escreve ao fechar.
- Imutável
  - Arquivo especificado como compartilhado, não pode ser modificado, ou seja, compartilhamento somente leitura.

# Política de Atualização de Cache

---

- Write-through
  - Escreve para o disco tão logo que o dado é armazenado na cache.
- Delayed-write
  - Escreve modificações na cache e só mais tarde no servidor.
  - Variações:
    - Write on close
    - Update em intervalos regulares

# Considerações Finais

---

- A escolha de um SAD depende das características e uso dos arquivos.
- Disponibilidade e desempenho podem ser obtidos com replicação e *caching*.
- Deve-se considerar desempenho, tolerância a falhas e segurança.
- Arquiteturas com servidores independentes ou aglomerados.
- Comunicação geralmente via RPC com TCP.

# Atividades

---

- Realize uma pesquisa sobre GFS (Google File System) e faça um resumo que contenha a definição, arquitetura e principais características do GFS.

# Referências

---

COULOURIS, George F; DOLLIMORE, Jean; KINDBERG, Tim; BLAIR, Gordon. **Sistemas distribuídos: conceitos e projeto**. 5. ed. Porto Alegre: Bookman, 2013.

THANH, Tran; et al. **A Taxonomy and Survey on Distributed File Systems**. Fourth International Conference on Networked Computing and Advanced Information Management, IEEE, 2008.

LEVY, Eliezer; SILBERSCHATZ, Abraham. **Distributed File Systems: Concepts and Examples**. ACM Computing Surveys, Vol. 22, No. 4, 1990.