Sistemas Distribuídos

Sistemas de Arquivos Distribuídos

Roteiro

- Sistema de arquivos distribuídos
  - Requisitos
  - Arquivos e diretórios
  - Compartilhamento
  - Cache
  - Replicação
  - Estudo de caso: NFS e AFS

Sistema de arquivos

- Responsável por organizar, armazenar, recuperar, nomear, compartilhar e proteger arquivos
- Fornece uma interface de programação que caracteriza a abstração de arquivo
  - Libera a aplicação dos detalhes de alocação e layout de arquivos

Sistema de arquivos distribuídos

- Serviço de arquivo: o que o sistema de arquivos oferece aos clientes
  - Primitivas, parâmetros e ações
- Servidor de arquivos: processo que roda em uma máquina do sistema e implementa o serviço de arquivos

Requisitos de um sistema de arquivos distribuídos

- Transparência
  - Acesso
  - Localização
  - Mobilidade
  - Desempenho
  - Escala

Requisitos de um sistema de arquivos distribuídos

- Atualizações concorrentes de arquivo
  - Alterações em um arquivo por um cliente não deve interferir nas operações simultâneas de outros clientes
  - Lock de arquivos ou registros
- Replicação de arquivos
  - Distribuição de carga → escalabilidade
  - Tolerância a falha

Requisitos de um sistema de arquivos distribuídos

- Heterogeneidade de hardware e sistema operacional
- Tolerância a falha
  - Operações idempotentes
  - Servidor stateless
  - Replicação

Requisitos de um sistema de arquivos distribuídos

- Consistência
  - Problemas com o uso de arquivos replicados ou cache
- Segurança
  - Necessidade de autenticar requisições dos usuários
- Eficiência
  - Desempenho e confiabilidade comparáveis ao sistema de arquivos local

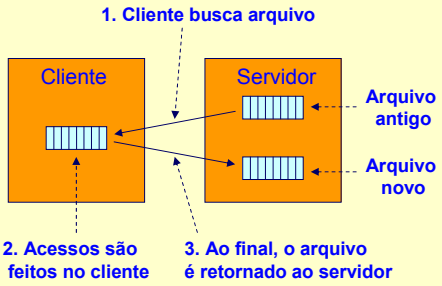
Arquivos

- O que é um arquivo?
  - Seqüência de bytes
  - Seqüência de registros
- Atributos de arquivos: informações sobre os arquivos
- Proteção:
  - Listas de capacidades (associadas aos usuários)
  - Listas de controle de acesso (associadas aos arquivos)

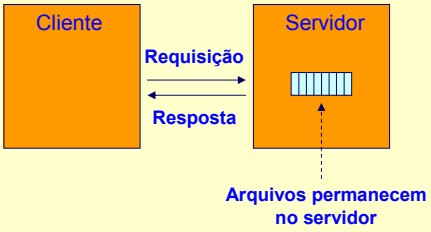
Modelos de acesso

- Remoto: grande número de operações no arquivo que permanece no servidor
- Local (upload/download): somente operações de leitura e escrita que transferem o arquivo inteiro entre cliente e servidor
  - Vantagem: simplicidade conceitual
  - Desvantagem: ocupa memória do cliente

Modelos de acesso



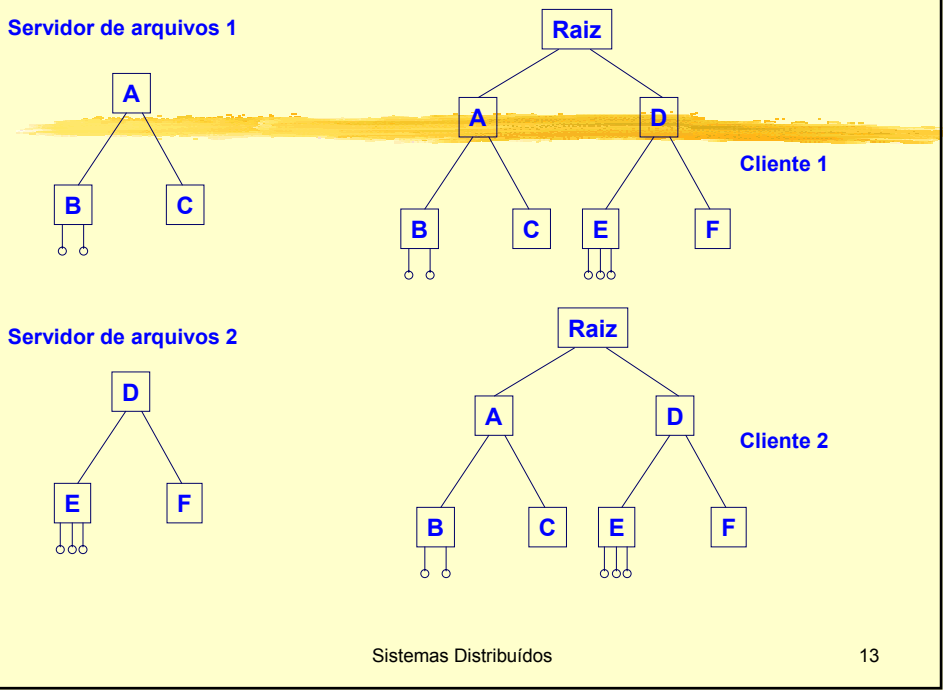
(a) Modelo upload/download

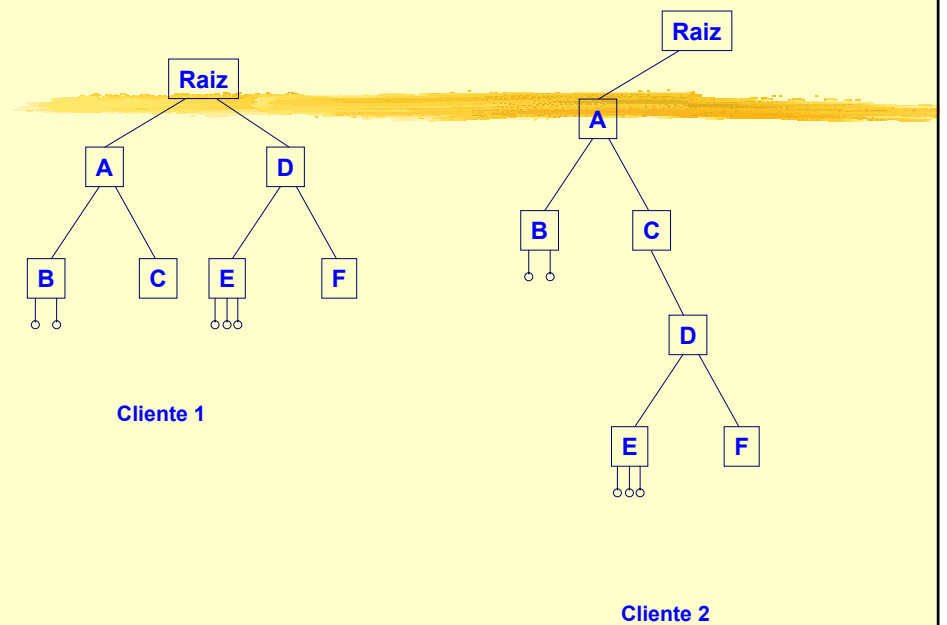


(b) Modelo de acesso remoto

Serviço de diretórios

- Fornece operações para criação e remoção de diretórios, identificação e mudança de nome de arquivos, movimentação de arquivos entre diretórios
  - Independe do modelo do serviço de arquivos
- Permitem a criação de subdiretórios que podem conter seus próprios subdiretórios → sistema de arquivos hierárquico


Servidor de arquivos 1
A
B
C
Raiz
A
D
B
C
E
F
Cliente 1
Servidor de arquivos 2
D
E
F
Raiz
A
D
B
C
E
F
Cliente 2


Raiz
A
D
B
C
E
F
Cliente 1
Raiz
A
B
C
D
E
F
Cliente 2

Transparência de nomeação

- Transparência de localização: o nome do caminho não deve dar nenhuma identificação de onde o arquivo está localizado
  - Ex.: servidor1/dir1/dir2/x - o arquivo x está no servidor1 mas não informa onde este servidor está localizado
  - Se o arquivo é grande e o servidor1 não tem espaço enquanto há bastante espaço no servidor2. Poder-se-ia mover o arquivo para o servidor2 mas isto mudaria o nome do caminho.

Transparência de nomeação

- Independência de localização: sistema no qual os arquivos podem ser movidos sem que seu caminho mude
- Três métodos para identificar arquivos e diretórios:
  - nome da máquina + nome do caminho
  - montagem dos sistemas de arquivos remotos na hierarquia de arquivos locais
  - um único espaço de nomes que devem ser os mesmos em todas as máquinas

Compartilhamento de arquivos

- Consistência seqüencial
  - No sistema com um processador as operações ocorrem seqüencialmente
    - Ordenação absoluta de tempo de todas as operações
  - Obtida facilmente em um sistema distribuído se houver um único servidor e os clientes não fizerem cache dos arquivos
  - Se fizer cache: enviar imediatamente as modificações realizadas → ineficiente

Compartilhamento de arquivos

- Semântica de sessão
  - As mudanças em um arquivo aberto só são visíveis ao processo que modificou o arquivo
  - Só quando o arquivo for fechado é que estas modificações serão visíveis aos outros processos
  - Quando 2 processos estiverem modificando um arquivo ao mesmo tempo o resultado final vai depender de quem fechou o arquivo por último

Compartilhamento de arquivos

- Arquivos imutáveis
  - Os arquivos podem ser criados e lidos mas não modificados.
  - Para modificar um arquivos deve-se fazer a substituição por um novo arquivo.
  - E se 2 processos estão tentando substituir um determinado arquivo ao mesmo tempo?
    - Mesmo problema da solução anterior

Compartilhamento de arquivos

- Transações atômicas
  - Utilização das transações atômicas para tratar arquivos compartilhados.
  - Desta forma todas as modificações tem a propriedade tudo-ou-nada

Propriedades de sistemas de arquivos

- Estudo realizado em 1981 (Satyanarayanan):
  - a maioria dos arquivos são pequenos (menores que 10K)
  - leitura é mais comum que escrita
  - leituras e escritas grande parte das vezes são feitas seqüencialmente
  - a maioria dos arquivos tem vida curta
  - não é usual arquivo compartilhado
  - processos em média usam poucos arquivos
  - existem classes de arquivos diferentes

Estrutura do sistema

- Distinção entre cliente e servidor
- Forma de estruturação do serviço de arquivos e de diretórios (juntos ou separados)
- Manutenção de informações de estado.
  - Sem informação: tolerante a falhas, não há necessidade de OPEN/CLOSE, não gasta espaço na memória do servidor com tabelas, não há limite de arquivos abertos e não há problemas quando o cliente falha
  - Com informação: mensagens de solicitação menores, melhor desempenho, possível ler informações adiantadas, fácil obter idempotência e possível bloquear arquivos.

Armazenamento em cache

- Em um sistema cliente-servidor existem quatro lugares para armazenamento de informação:
  - disco do servidor (melhor opção a princípio)
  - memória principal do servidor
  - memória principal do cliente
  - disco do cliente
- Uso de cache ajuda a melhorar o desempenho do sistema

Armazenamento em cache

- Qual é a unidade de transferência de cache?
  - Arquivos ou blocos?
- Qual é o algoritmo de substituição?
  - LRU com listas ligadas
- Para evitar atraso de acesso à rede é conveniente manter uma cache no lado cliente
- Opções para cache:
  - espaço de endereçamento do processo
  - dentro do kernel
  - processo gerenciador de cache

Consistência de cache

- Clientes diferentes podem ter versões diferentes de arquivos ou blocos em suas caches → semântica de sessão
- Write-through: quando uma entrada da cache (arquivo ou bloco) o novo valor é mantido na cache e enviado imediatamente ao servidor. Problemas:
  - gerente mantém cache após finalização do processo
  - tráfego alto → escrita retardada

Consistência de cache

- Write-on-close: escrever o arquivo de volta no servidor somente quando ele for fechado.
  - Arquivos temporários: esperar para escrever de volta
- Algoritmo de controle centralizado

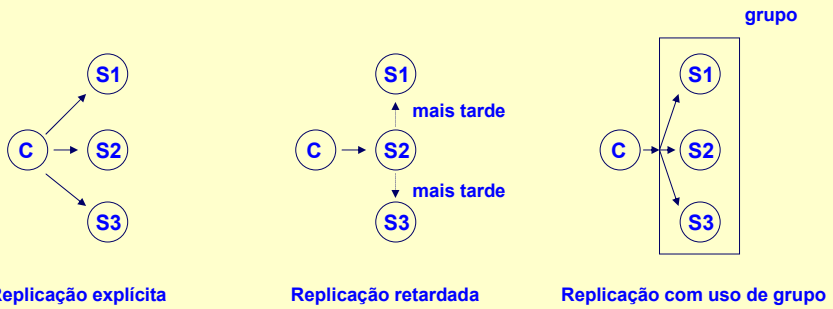
Replicação

- Serviço de replicação de arquivo: são mantidas várias cópias de arquivos que são armazenadas em um servidores separados
  - aumento de confiabilidade: se um servidor sair do ar, não haverá perda de dados
  - melhora da disponibilidade: acesso possível mesmo que um servidor esteja fora do ar
  - aumento de desempenho: carga de trabalho dividida entre os vários servidores

Replicação

- Transparência de replicação: o sistema se encarrega da replicação e o processo não fica ciente do serviço
- Implementação de replicação
  - replicação explícita: o processo controla o procedimento
  - replicação retardada: o servidor se encarrega de atualizar as réplicas
  - replicação com uso de comunicação em grupo: as chamadas são transmitidas a todos os servidores

Replicação



Replicação explícita | Replicação retardada | Replicação com uso de grupo

Protocolos de atualização

- Como os arquivos podem ser modificados?
  - Garantir a coerência das modificações
- Algoritmos:
  - Replicação da cópia principal
  - Método eletivo

Protocolos de atualização

- Replicação da cópia principal:
  - um dos servidores é escolhido para ser o principal
  - as modificações de arquivos são feitas no servidor principal que se encarrega de enviar mensagens de atualização aos servidores secundários
  - as leituras podem ser feitas em qualquer servidor
  - as atualizações são escritas em memória estável antes de modificar a cópia principal
- Vantagem: simplicidade
- Desvantagem: ponto único de falha (servidor principal)

Protocolos de atualização

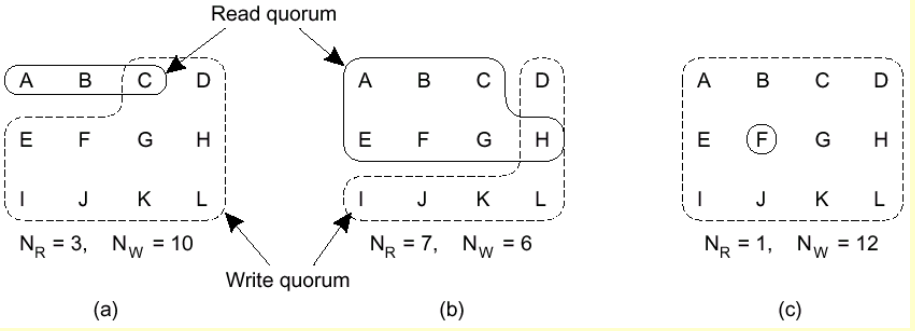
- Método eletivo: Guifford (1979)
  - os clientes devem solicitar e adquirir permissão de vários servidores antes de fazer modificações
  - se um arquivo foi replicado em N servidores, o cliente entrar em contato com no mínimo N/2 + 1 servidores a fim de obter consentimento para atualizações
  - caso haja concordância, o arquivo é modificado e um novo número de versão é associado ao arquivo

Protocolos de atualização

- para ler um arquivo replicado, o cliente solicita o número da versão do arquivo de metade mais um dos servidores, se todas forem as mesmas então esta versão é a mais recente

- O esquema é um pouco mais geral:
  - quorum de leitura (N_R)
  - quorum de escrita (N_W)
  - $N_R + N_W > N$

Protocolos de atualização



Alguns sistemas de arquivos distribuídos

- 9P
- AFS
- AppleShare
- Coda
- DFS
- Freenet
- NFS
- OpenAFS
- SMB (CIFS)

NFS - Network File System

- Desenvolvido pela Sun
  - Introduzido em 1985
- As interfaces chave foram colocadas em domínio público e um código fonte de referência disponibilizado sob licença → padrão de fato

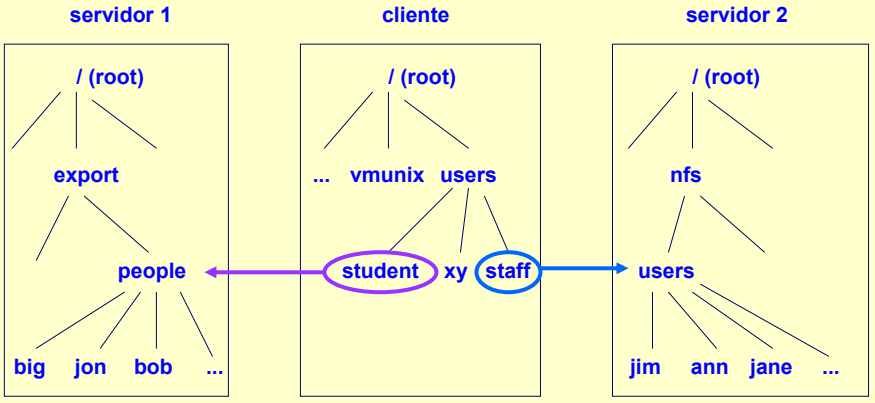
NFS - Network File System

- NFS fornece o acesso transparente a arquivos remotos para programas cliente
- Tipicamente cada computador possui módulos cliente e servidor NFS instalados no kernel
- A relação cliente-servidor é simétrica

Características

- Transparência de acesso: modulo NFS apresenta uma API ao processo idêntica à API do SO
- Transparência de localização: cada cliente estabelece um espaço de nomes adicionando os sistemas de arquivo remotos ao seu sistema local
  - O sistema é exportado pelo servidor e montado pelo cliente

Características



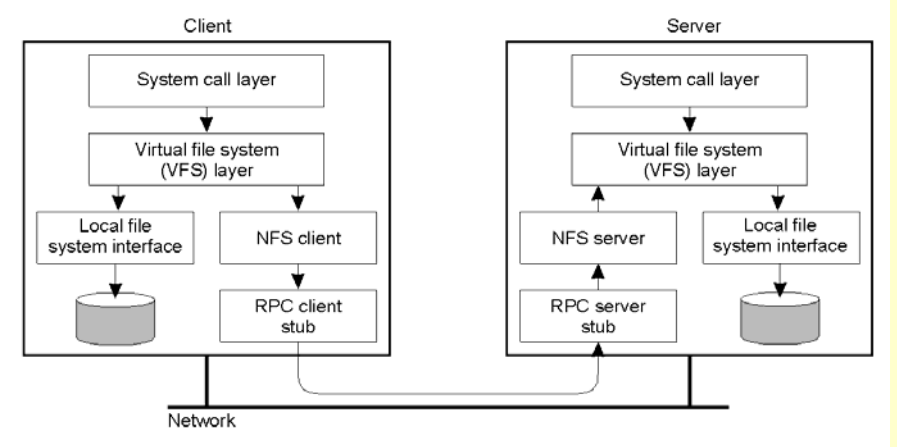
Características

- Transparência de falha: serviço sem informação de estado e a maioria das operações são idempotentes
- Transparência de desempenho: cliente e servidor empregam a técnica de cache para melhorar desempenho
- Transparência de migração: serviço de montagem fornece uma interface RPC para montagem e desmontagem de diretórios remotos

Arquitetura NFS

- Implementação em três níveis:
  1. Nível da chamada de sistema: trata as chamadas OPEN, READ e CLOSE
  2. Sistema de arquivos virtual (VFS): mantém uma tabela com uma entrada para cada um dos arquivos abertos (nó-V)
  3. SO ou cliente NFS

Arquitetura NFS



Montando diretório

- Programa mount executado pelo administrador do sistema

1. Recebe um nome de diretório remoto e um local
2. Comunica-se com a máquina remota
3. Se o arquivo existir e puder ser montado remotamente o servidor autoriza o cliente a fazer a montagem

Montando diretório

4. O cliente faz uma chamada de sistema MOUNT para o kernel
5. O kernel constrói um nó-V para o diretório remoto e solicita ao cliente a criação de um nó-R (nó-i remoto) em suas tabelas internas
   - O nó-V (nível VFS) aponta para o nó-R (cliente NFS) ou nó-I (SO local)

Abrindo arquivo

- Quando um arquivo remoto é aberto:
  - o kernel descobre o diretório onde o arquivo vai ser montado (caminho), verifica se o diretório é remoto e procura o diretório nos nós-V o ponteiro para o nó-R
  - O controle passa para o cliente NFS que prepara uma entrada de nó-R em suas tabelas e passa esta informação para o VFS
  - O VFS coloca em suas tabelas um nó-V apontando para o nó-R recém criado
  - O processo recebe um descritor de arquivo

Lendo arquivo

- Quando um arquivo remoto for lido:
  - o VFS localiza o nó-V correspondente ao descritor de arquivo e a partir dele determina se o arquivo é local (nó-I) ou remoto (nó-R)
- Leitura antecipada: as transferências de dados são feitas usando números grandes de bytes, normalmente 8K (eficiência)
- Operação de escrita é semelhante
  - Escrita retardada: os dados são cumulados localmente

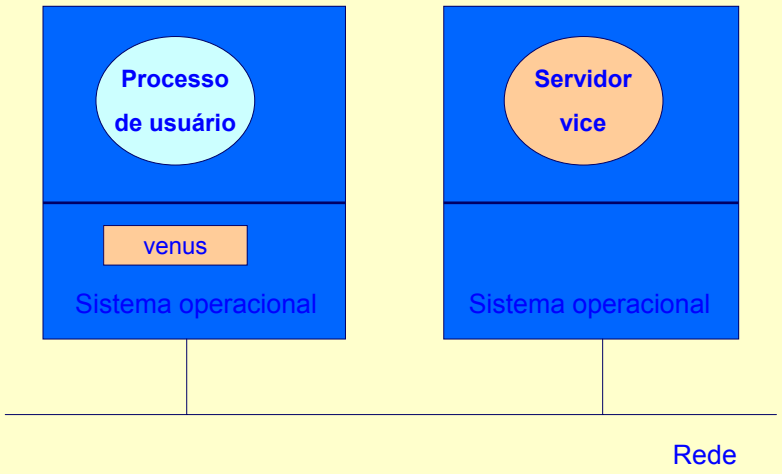
Cache

- Os servidores colocam dados na cache para evitar acessos a disco
- Para resolver coerência de cache:
  - temporizador associados ao bloco de cache
    - 3 s para arquivos e 30 s para diretórios
  - sempre que um arquivo da cache for aberto é verificada a data de última modificação no servidor
  - a cada 30 s todos os blocos são enviados ao servidor

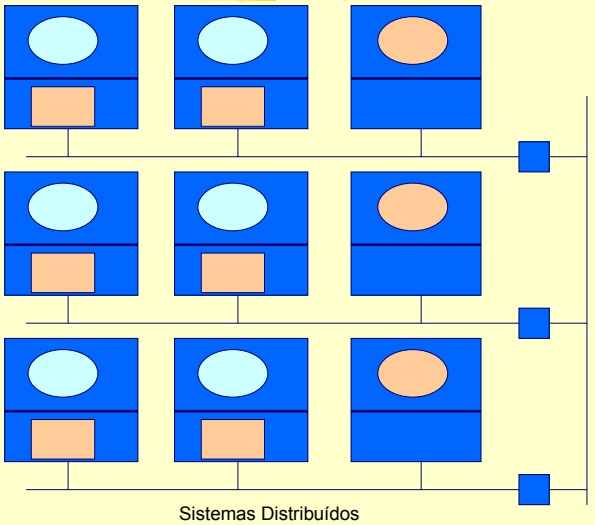
AFS - Andrew File System

- Desenvolvido pela Universidade de Carnegie-Mellon
- Objetivo: suporte ao compartilhamento de informação de larga escala, minimizando a comunicação entre cliente e servidor
- Cada usuário tem uma estação de trabalho modificada executando uma versão do SO, que inclui um pedaço de código denominado **venus** e um servidor de arquivos no espaço do usuário denominado **vice**

AFS - Andrew File System



AFS - Andrew File System



AFS - Andrew File System

- As estações de trabalho são agrupadas em células
- Espaço de nomes inclui diretórios
  - /cache: contém cópias dos arquivos remotos em cache de disco
  - /cmu: traz nomes das células remotas compartilhadas
    - Abaixo delas estão seus respectivos sistemas de arquivos
- Semântica parecida com sessão

Abrindo arquivo

- Quando um arquivo é aberto, o código venus chama um OPEN e transfere o arquivo todo ou, se for um arquivo gigantesco, uma parte grande dele para o disco local e o insere no diretório /cache
  - Permite que o usuário faça o máximo localmente
- O descritor retornado aponta para o arquivo em /cache
- As leituras e escritas operam na cópia local
- Ao fechar ele é copiado de volta

Consistência de arquivos

- Quando um arquivo é transferido para a cache, venus diz a vice se pode ou não haver aberturas subseqüentes por outros processos
- Em caso negativo vice registra a localização do arquivo
- Se outro processo pedir o mesmo arquivo, vice envia msg para venus solicitando envio de cópia arquivo caso tenha sido modificado