Consistência em sistemas de arquivos e Desempenho do sistema de arquivos

Grupo 7 – Gabriel Sabaudo e Guilherme Henrique

Consistência do sistema de arquivos.

- Muitos sistemas de arquivos **leem os blocos, modificam seu conteúdo** e só depois os **escrevem.** Se o sistema cair antes que todos os blocos alterados tenham sido escritos, o sistema de arquivos poderá ficar em um estado inconsistente.

- Esse problema será especialmente crítico se alguns dos blocos ainda não escritos forem de **i-node, blocos de diretório** ou blocos que contenham a lista de **blocos livres.**

I-nodes

- Como você provavelmente já sabe, **tudo é considerado um arquivo no Linux.** Isso inclui dispositivos de hardware, processos, diretórios, arquivos regulares, soquetes, links e assim por diante.

- Além disso, geralmente, o sistema de arquivos se divide em blocos de dados e i-nodes. Com isso dito, os i-nodes **são como uma base do sistema de arquivos Linux.** Para explicar mais claramente, um i-node é **uma estrutura de dados que armazena metadados sobre cada arquivo em seu sistema de PC.**

Programa utilitário

- Para tratar do problema de inconsistência nos sistemas de arquivos, a maioria dos computadores **tem um programa utilitário** que verifica a consistência do sistema de arquivos. Por exemplo, o UNIX tem o fsck e o Windows tem o scandisk.

- **Esse utilitário** pode ser executado sempre que o sistema estiver sendo inicializado, especialmente depois de uma queda.

- O princípio geral de **usar a redundância inerente do sistema de arquivos** para consertá-lo vale para os dois. Todos os verificadores conferem cada sistema de arquivos (participação do disco) independentemente dos outros.

Verificações de consistência

- Existem dois tipos de verificações de consistência: **por blocos** e **por arquivos.**

Consistência dos blocos

- Para verificar a consistência dos blocos, o programa constrói duas tabelas, cada uma com um contador para cada bloco, inicialmente contendo 0.

- Os contadores da **primeira tabela monitoram quantas vezes cada bloco está presente em um arquivo**; os contadores da **segunda tabela registram quantas vezes cada bloco está presente na lista de livres** (ou mapas de bits de bloco livres).

Consistência dos blocos

- O programa, então, lê todos os i-nodes usando um dispositivo cru (raw), isto é, que ignora a estrutura dos arquivos e **apenas devolve todos os blocos do disco começando no 0.**

- A partir de um i-node, é possível **construir uma lista de todos os números de blocos usados no arquivo** correspondentes, conforme **cada número de bloco é lido seu contador na primeira tabela é incrementado.**

- O programa então verifica a lista ou o mapa de bits de livres para encontrar todos os blocos que não estiverem sendo usados. **Cada ocorrência de um bloco na lista de livres faz com que seu contador na segunda tabela seja incrementado.**

Consistente

- Se o sistema de **arquivos estiver consistente,** cada bloco terá um 1 ou na primeira ou na segunda tabela, como mostra a figura A.

Bloco desaparecido

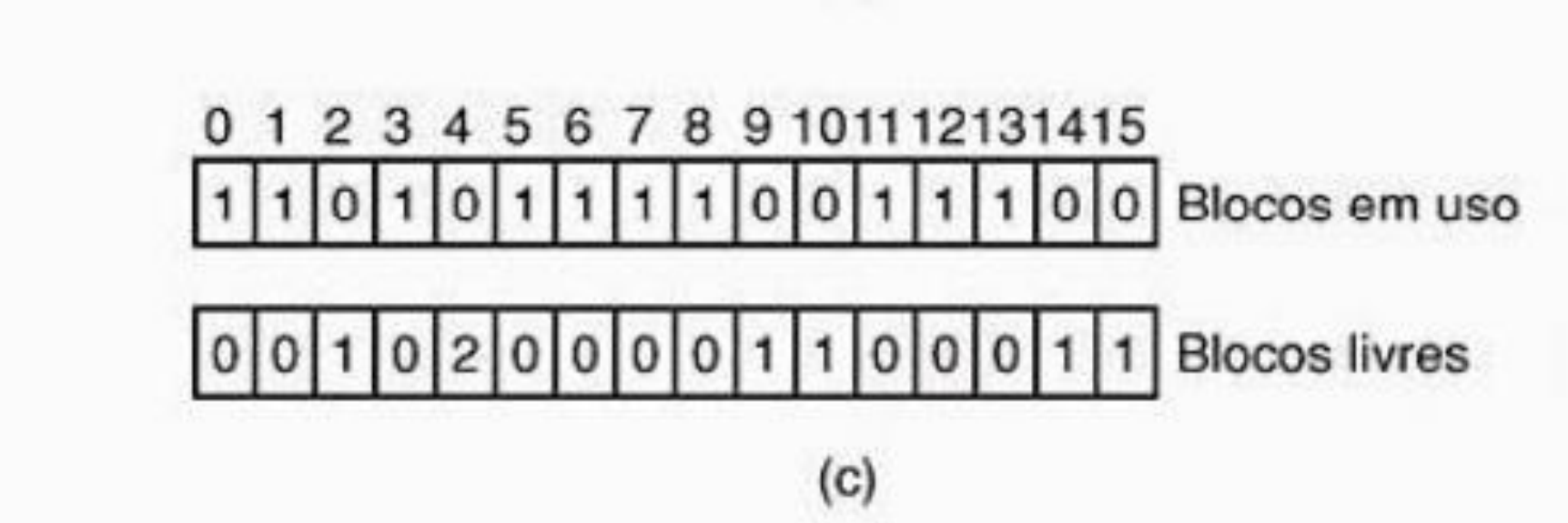
- Contudo, depois de uma queda no sistema, as tabelas podem ficar como mostrado na figura B na qual o bloco 2 não ocorre em nenhuma tabela. Será reportado como um **bloco desaparecido.**

- Embora os **blocos desaparecidos** não causem nenhum mal, eles ocupam espaço e, portanto, reduzem a capacidade do disco. A solução para os blocos desaparecidos é simples: o verificador de sistema de arquivos apenas os inclui na lista de blocos livres.

Bloco duplicado na lista de livres

- Outra situação possível é a mostrada na figura C. Nela vemos um bloco, o número 4, que **ocorre duas vezes na lista de livres.**

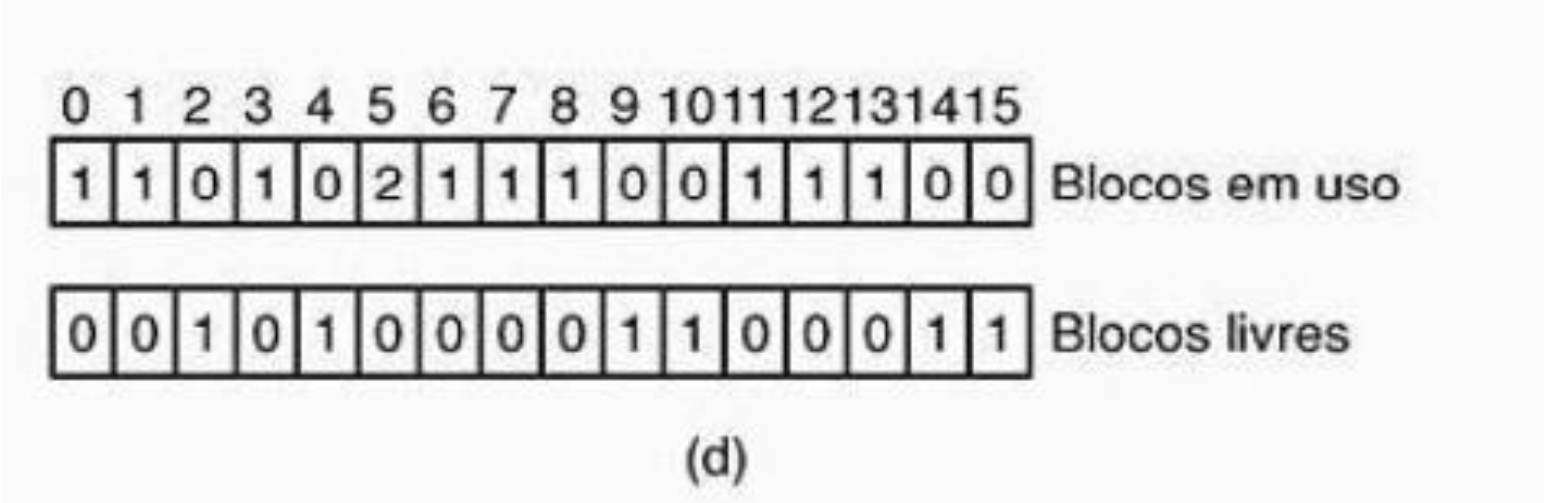
- A solução aqui também é bem simples: reconstruir a lista de livres.



Bloco de dados duplicados

- A pior coisa que pode acorrer é o mesmo bloco de dados **estar presente em dois ou mais arquivos,** como mostra a figura D com o bloco 5.

- Se um desses arquivos for removido, o bloco 5 será colocado na lista de livres, resultando na situação em que o mesmo bloco está em uso e livre ao mesmo tempo. Se ambos os arquivos forem removidos, o bloco será colocado na **lista de livres duas vezes.**



Bloco de dados duplicados

- O que o verificador de sistema de arquivos deve fazer é alocar um bloco livre, copiar o conteúdo do bloco 5 nele e inserir a cópia em um dos arquivos. Desse modo o cont**eúdo dos arquivos permanece inalterado,** mas a estrutura do sistema de arquivos, pelo menos fica consistente. O erro deve ser reportado para permitir que o usuário impressione o dano.

Consistência por arquivos

- Além de verificar se cada bloco está contabilizando corretamente, o verificador de sistemas de arquivos também **checa os sistemas de diretórios.** Além disso, ele usa uma tabela de contadores, mas **por arquivos,** e não por blocos.

- Ele **inicializa a partir do diretório raiz** e recursivamente percorre a árvore, inspecionando cada diretório do sistema de arquivos. Para cada arquivo no diretório, ele incrementa um contador para **contar o uso de arquivos.** Lembre-se de que, por causa de ligações estritas (hard links), um arquivo pode aparecer em 2 ou mais diretórios. As ligações simbólicas não contam e não fazem com que o contador inclemente para o arquivo alvo.

Consistência por arquivos

- Quando estiver tudo terminado, haverá uma lista, indexada pelo número de i-node, **indicando quantos diretórios cada arquivo contém.**

- Eles então **compararam** esses números com a contagens de ligações armazenadas nos próprios i-nodes. Essas contagens iniciam em 1 quando um arquivo é criado e **são implementadas a cada vez que é uma ligação estrita é feita para o arquivo.**

- Em um sistema de arquivos consistente, os computadores devem ser iguais. Contudo, há a possibilidade de ocorrer 2 tipos de erros: a contagem de ligações no i-node pode ser **alta** ou **baixa** demais.

Contagem de ligações alta

- Se a contagem de ligações for mais alta que o número de entradas de diretório, então, mesmo que todos os arquivos sejam removidos dos diretórios, **a contagem ainda será diferente de zero** e o i-node não será removido.

- Esse erro não é grave, mas **consome espaço de disco** com os arquivos que não ocupam nenhum diretório. Ele deve ser reparado atribuindo-se o valor correto a contagem de ligações no i-node.

Contagem de ligações baixa

- O outro erro é potencialmente catastrófico. Se duas entradas de diretório forem redirecionadas para um arquivo, mas o i-node indicar que há somente uma, quando uma das entradas de diretório for removida, **a contagem do i-note irá para zero.**

- Quando uma contagem do i-node irá para zero, **o sistema de arquivos marca como não usado e libera todos os seus blocos.** Essa ação resultará em um dos diretores apontando agora para um i-node não usado, cujo blocos podem logo ser atribuídos a outros arquivos.

- Novamente, a solução é forçar a contagem de ligações a assumir o número real de entradas no diretório.

Desempenho do Sistema de Arquivos

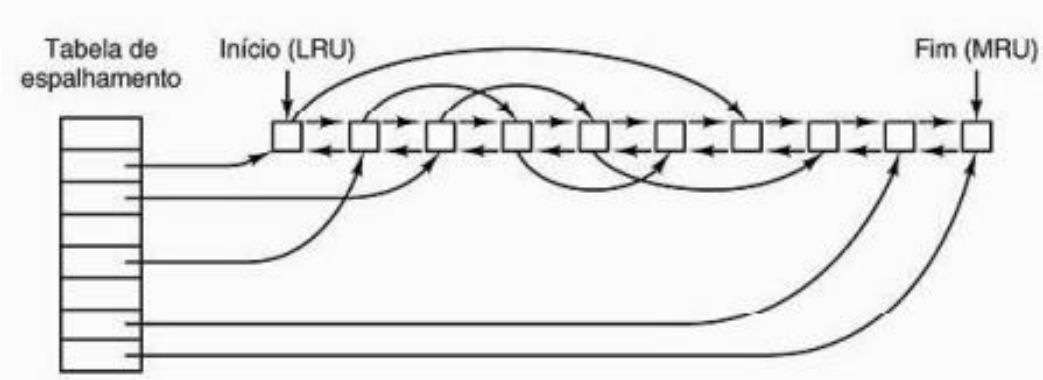
- Em geral, o acesso a disco é mais lento do que o acesso à memória
- Como consequência dessa demora ao acesso, muitos sistemas de arquivos foram implementados com várias otimizações para melhorar o desempenho
- Cache de blocos, Leitura antecipada de blocos e Redução do movimento do braço do disco

Desempenho do Sistema de Arquivos - Cache de blocos

- É a técnica mais utilizada para reduzir o acesso ao disco
- A cache é uma coleção de blocos que pertencem ao disco mas são mantidos na memória por questões de desempenho
- Existem vários algoritmos para o gerenciamento da cache em disco
- O mais comum verifica todas as requisições de leitura para averiguar se o bloco está na cache
- Se o bloco estiver na cache, a requisição será satisfeita sem um acesso ao disco
- Se não estiver, primeiro ele será lido do disco para a cache, e então será copiado para onde for necessário
- As próximas requisições para o mesmo bloco serão então feitas a partir da cache

Desempenho do Sistema de Arquivos - Cache de blocos

A figura abaixo representa a operação da cache



- Há muitos blocos na cache, então é preciso determinar de forma rápida se um dado bloco está presente

Desempenho do Sistema de Arquivos - Cache de blocos

- O modo comum é realizar o mapeamento entre o dispositivo e o endereço de disco numa hash table (tabela de espalhamento)
- Todos os blocos com o mesmo valor na tabela poderão ser identificados e encadeados numa lista para que a cadeia possa continuar
- Caso algum bloco precisar ser carregado numa cache que já está cheia, algum bloco deverá ser removido
- Além da cadeia de colisões a partir da hash table, há também uma lista bidirecional que é responsável por mudar os blocos de posição (a ordem está pelo bloco menos recentemente utilizado no início e o mais utilizado no fim)

Desempenho do Sistema de Arquivos - Cache de blocos

- Essa ordem LRU exata pode trazer problemas de inconsistência no sistema de arquivos
- Se algum bloco crítico como um i-node por exemplo, for para a cache, modificado, mas não for escrito no disco, poderá afetar o sistema de arquivos para um estado inconsistente, assim como outras ocasiões
- Dependendo da importância do bloco em questão, ele pode ser escrito imediatamente no disco para evitar perda de informações
- Se caso o bloco precisar ser reutilizado mais vezes, ele pode ser inserido em posições estratégicas da lista LRU para melhor desempenho

Desempenho do Sistema de Arquivos - Leitura antecipada de blocos

- Essa técnica se baseia em tentar transferir os blocos para a cache antes que eles sejam necessários para aumentar sua taxa de assertividade
- Ao obter um bloco K, ele o faz e também verifica na cache se há um próximo bloco K+1
- Ideal para a leitura de arquivos sequenciais, pois facilita a previsão de blocos que serão necessários
- Não funciona para arquivos com acesso aleatório por conta de sua imprevisibilidade, mas isso pode ser contornado com monitoramentos no padrão de acesso de cada arquivo aberto

Desempenho do Sistema de Arquivos - Redução do movimento do braço do disco

- Consiste em reduzir o número de movimentos do braço do disco
- Coloca os blocos sujeitos a mais acesso em sequência, próximos uns aos outros
- Pode ser feita através de uma reorganização inteligente dos blocos dos arquivos em disco
  - Monitorar o armazenamento do disco em grupos consecutivos para reduzir o número de posicionamentos do braço
  - Ler arquivos muito pequenos em sistemas que usam i-nodes requer 2 acessos, então posicionar os i-nodes no centro do disco melhora no desempenho de acesso