

Árvore-B Virtual

Prof. Ms. Anderson Canale Garcia

Adaptado de:

Profa. Dra. Cristina D. Aguiar

Contexto

- Índice Simples X Árvore-B

Acessos a Disco

- Problema
 - encontrar uma maneira de fazer um uso eficiente de índices que são muito grandes para serem armazenados **inteiramente** em memória principal (i.e., RAM)
 - Objetivo
 - encontrar uma maneira de **diminuir** o número médio de acessos a disco para pesquisa
-

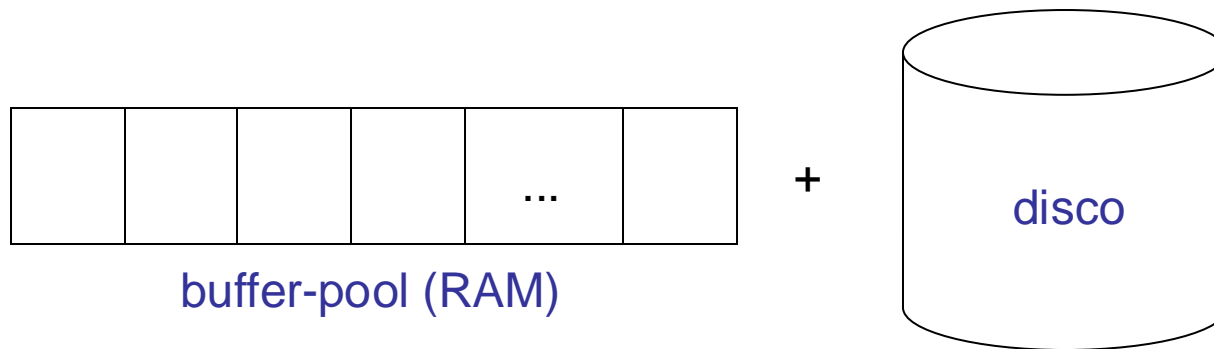
Exemplo

- 1 MB de registros
 - 256 K de RAM para uso
 - 4 K de tamanho de página
 - 64 chaves por página
 - Árvore-B em três níveis
 - Como melhorar?
-

Melhorias de Desempenho

- Manter **a página raiz** em memória principal
 - ainda deixa espaço disponível em RAM
 - diminui o número de acessos a disco em 1 no pior caso
 - Manter em um *buffer-pool* (i.e., em RAM) **um certo número de páginas** da árvore-B
 - abordagem mais genérica
-

Árvore-B Virtual



- Pesquisa
 - primeiro procura a página no *buffer-pool* para evitar acessos a disco
 - se a página não estiver no *buffer-pool*, o acesso é realizado em disco e a página é copiada para o *buffer-pool*
-

Substituição de Páginas

- ***Page Fault***

- processo de acessar o disco para trazer uma página que não está no *buffer-pool*
- causas
 - a página nunca foi utilizada
 - a página foi *substituída* no *buffer-pool* por outra página

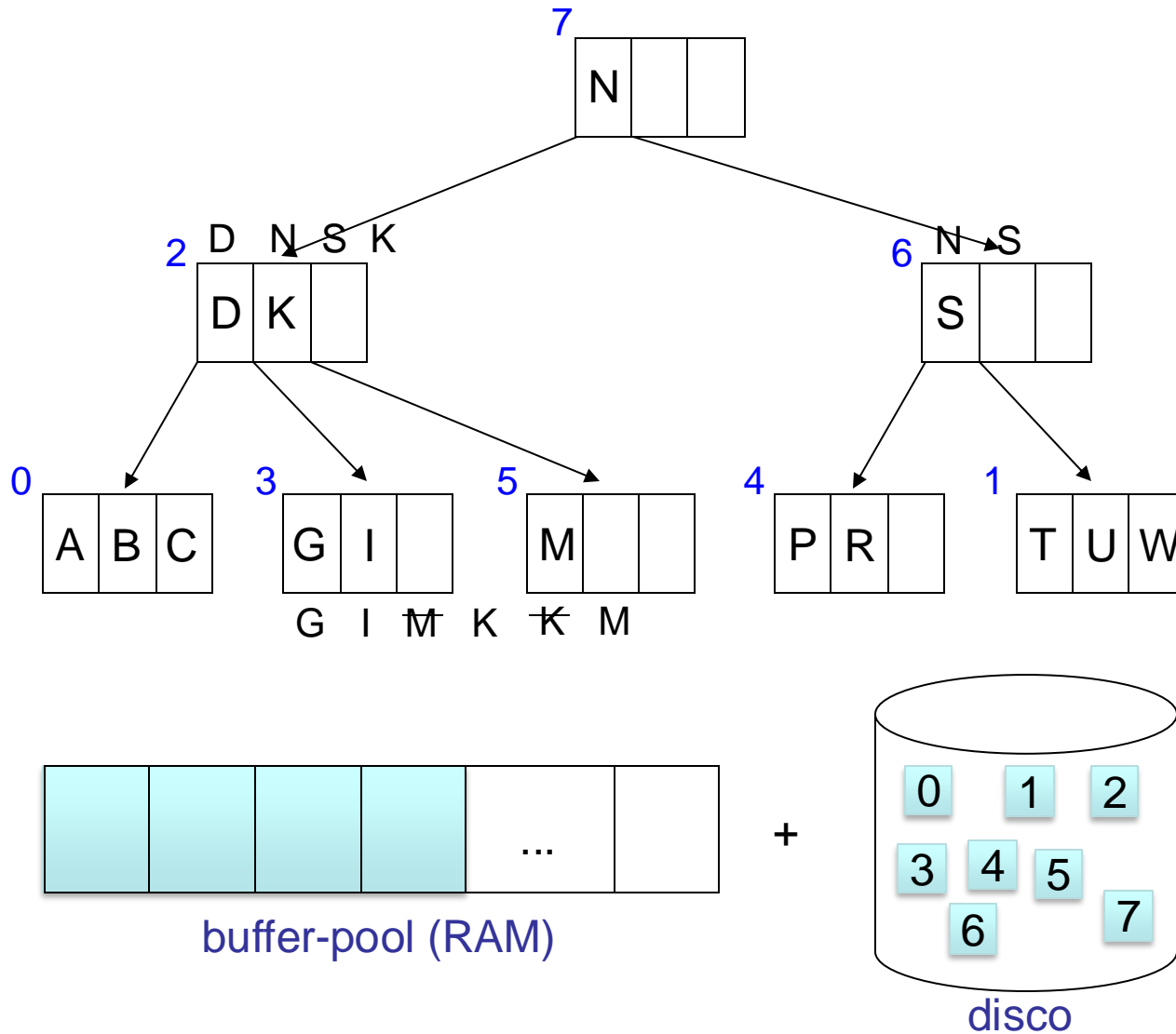
- **Decisão crítica**

- qual página deve ser substituída no *buffer-pool*, quando este encontra-se cheio?
-

Opções

- Política **LRU** (*least recently used*)
 - substitui a página que foi acessada menos recentemente
 - Substituição baseada na **altura da página**
 - mantém as páginas que estão nos níveis mais altos da árvore (i.e., próximas à raiz)
 - utiliza a política LRU para as demais páginas (i.e., páginas mais utilizadas)
-

Exemplo



Conclusão

O uso de *buffers* deve ser feito em qualquer situação real de utilização de árvore-B
