GBC053–Gerenciamento de Banco de Dados
Armazenamento de Dados

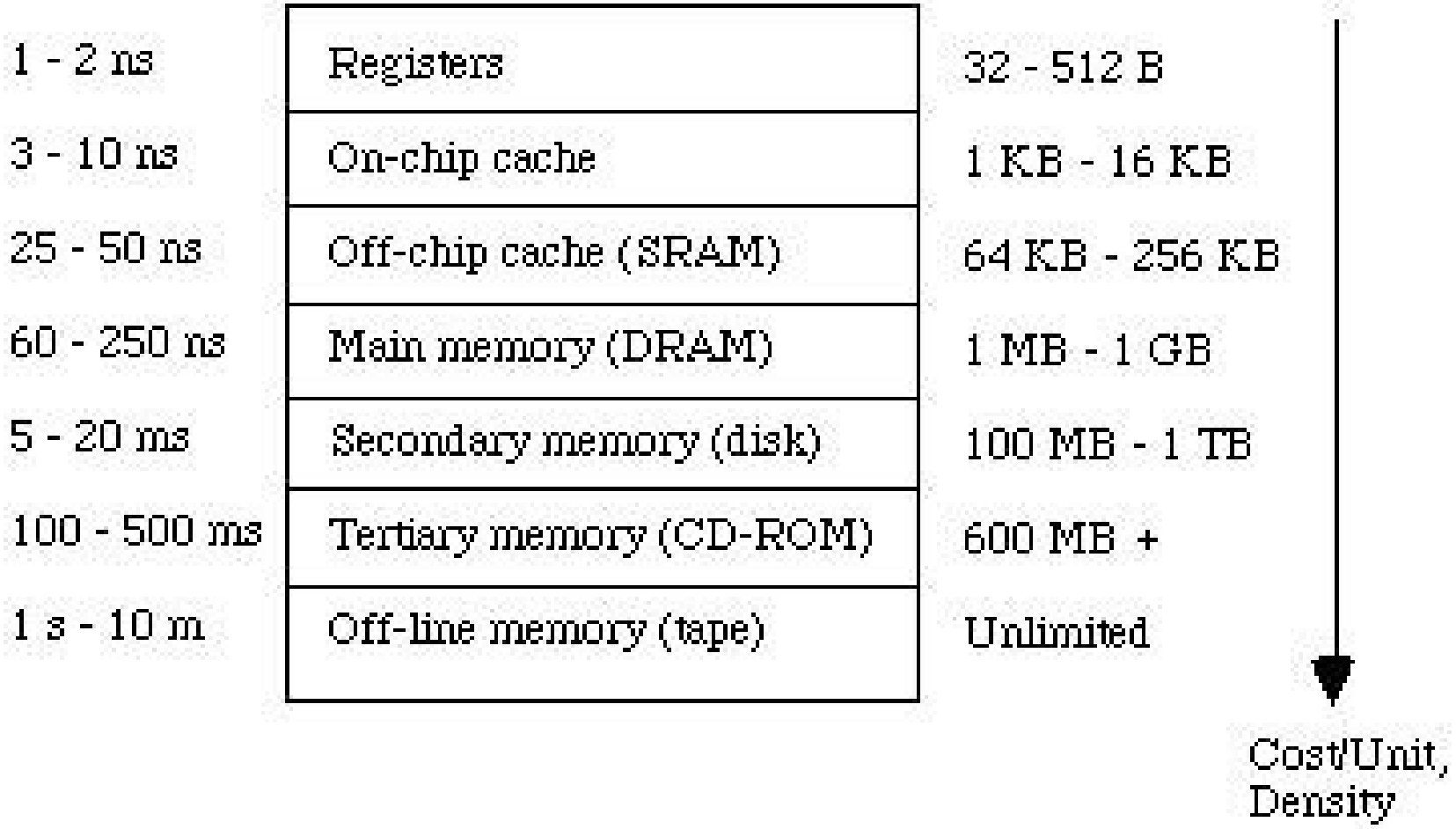
Ilmério Reis da Silva
ilmerio@facom.ufu.br
www.facom.ufu.br/~ilmerio/gbd
UFU/FACOM/BCC

ROTEIRO

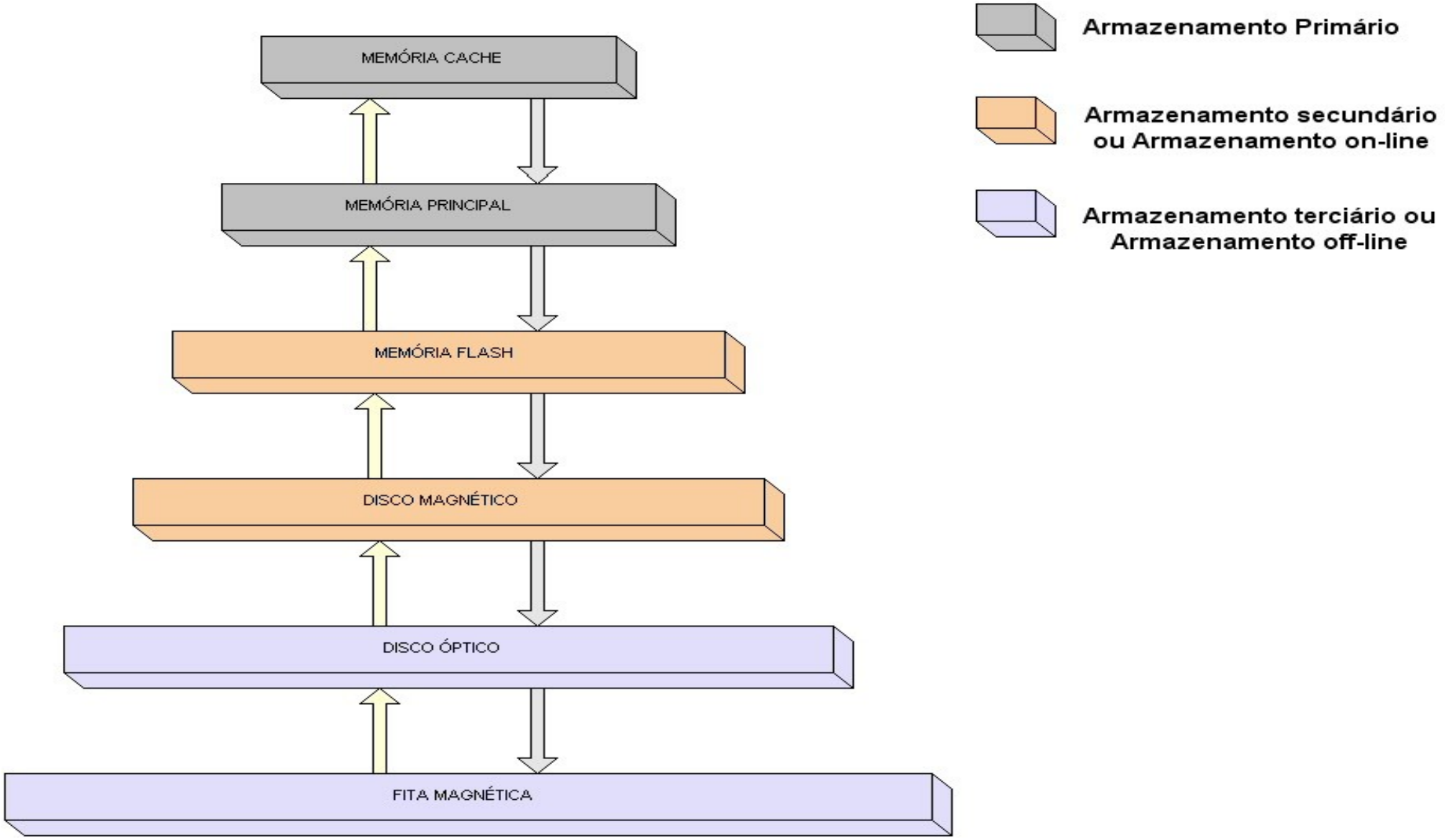
- ***Hiearquia de memórias e desempenho de discos***
- ***Gerência de espaço em disco***
- ***Gerência de buffer pool***
- ***Formatos de registros e páginas***

Hiearquia de memórias e desempenho de discos

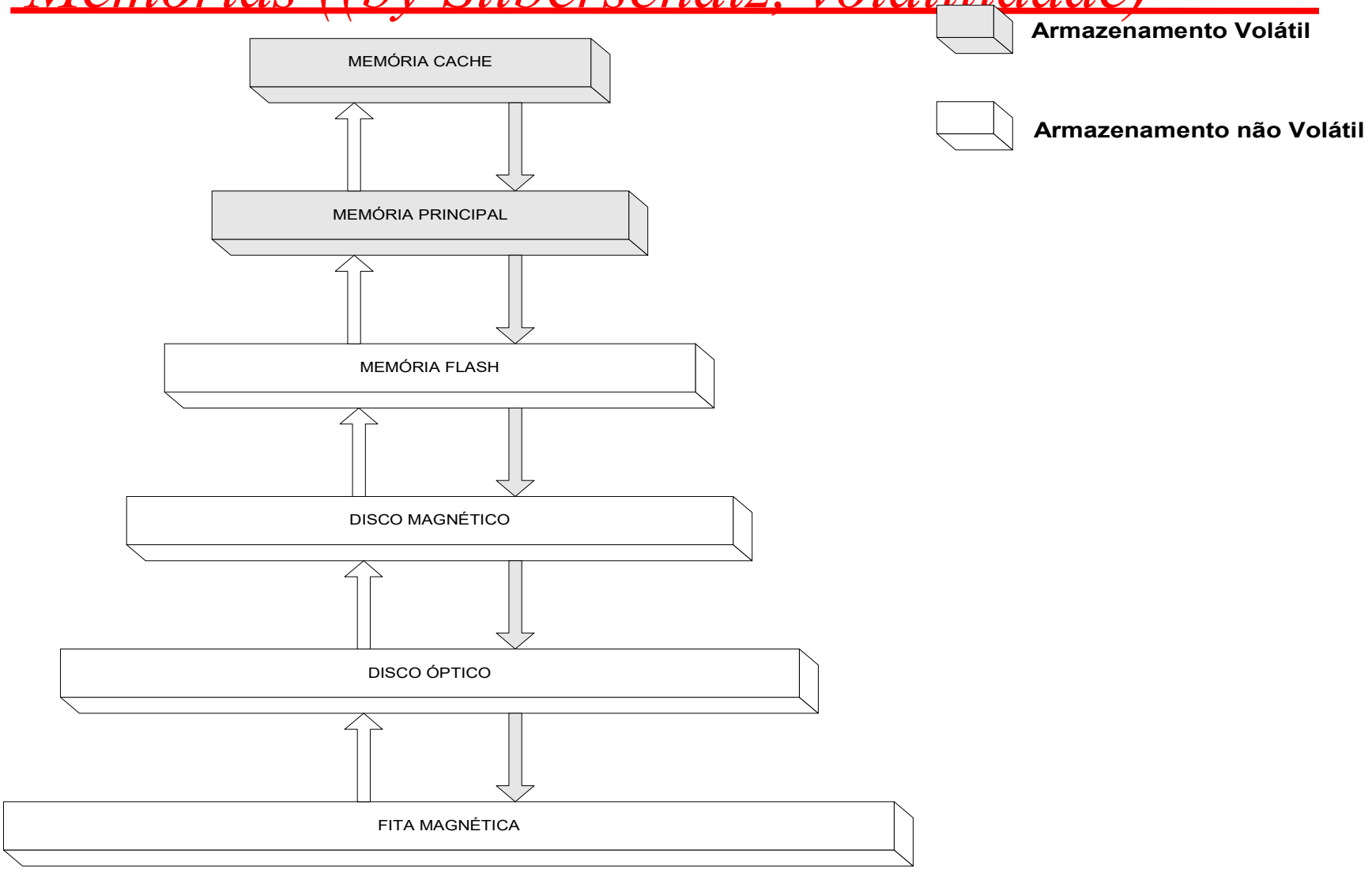
Armazenamento de Dados - Hierarquia de Memórias – Quadro 1



Armazenamento de Dados- Hierarquia de Memórias (by Silberschatz)



Armazenamento de Dados - Hierarquia de Memórias ((by Silberschatz, volatilidade)



Armazenamento de Dados - Hierarquia de Memórias - SSD

Memórias Flash

- Origem em EEPROM(*Electrically-Erasable Programmable Read-Only Memory*), mas com regravação em blocos, o que a torna bem mais barata que as EEPROMs originais
- Criada em 1980 e comercializada a partir de 1988 a memória *Flash do tipo NOR* tem alta velocidade de leitura e baixa velocidade de gravação, e é usada principalmente em cartões de memória, BIOS e alguns *firmwares*
- *A memória Flash do tipo NAND* foi criada em 1989 tem maior velocidade de gravação, entretanto não faz acesso aleatório, mas somente leitura sequencial em grandes blocos. Seu custo é inferior à NOR.

Armazenamento de Dados - Hierarquia de Memórias – SSD vantagens/desvantagens

Memórias Flash - Caraterísticas

VANTAGENS

- Não volátil
- Mais resistentes a choques do que discos
- Velocidade
  - leitura na ordem de 100 ns, entre RAM e DISCO
  - Latência é grande, mas menor que dos discos
  - Transferência de leitura/gravação da ordem de 100MBs (A DDR2-400 chega a 3,2GBs)

DESVANTAGENS

- Gravação por bloco
- Número limitado de ciclos de regravações (entre mil e um milhão de vezes)

Armazenamento de Dados Hierarquia de Memórias – ssd considerações finais

Memórias Flash – Considerações finais (dados de 2008)

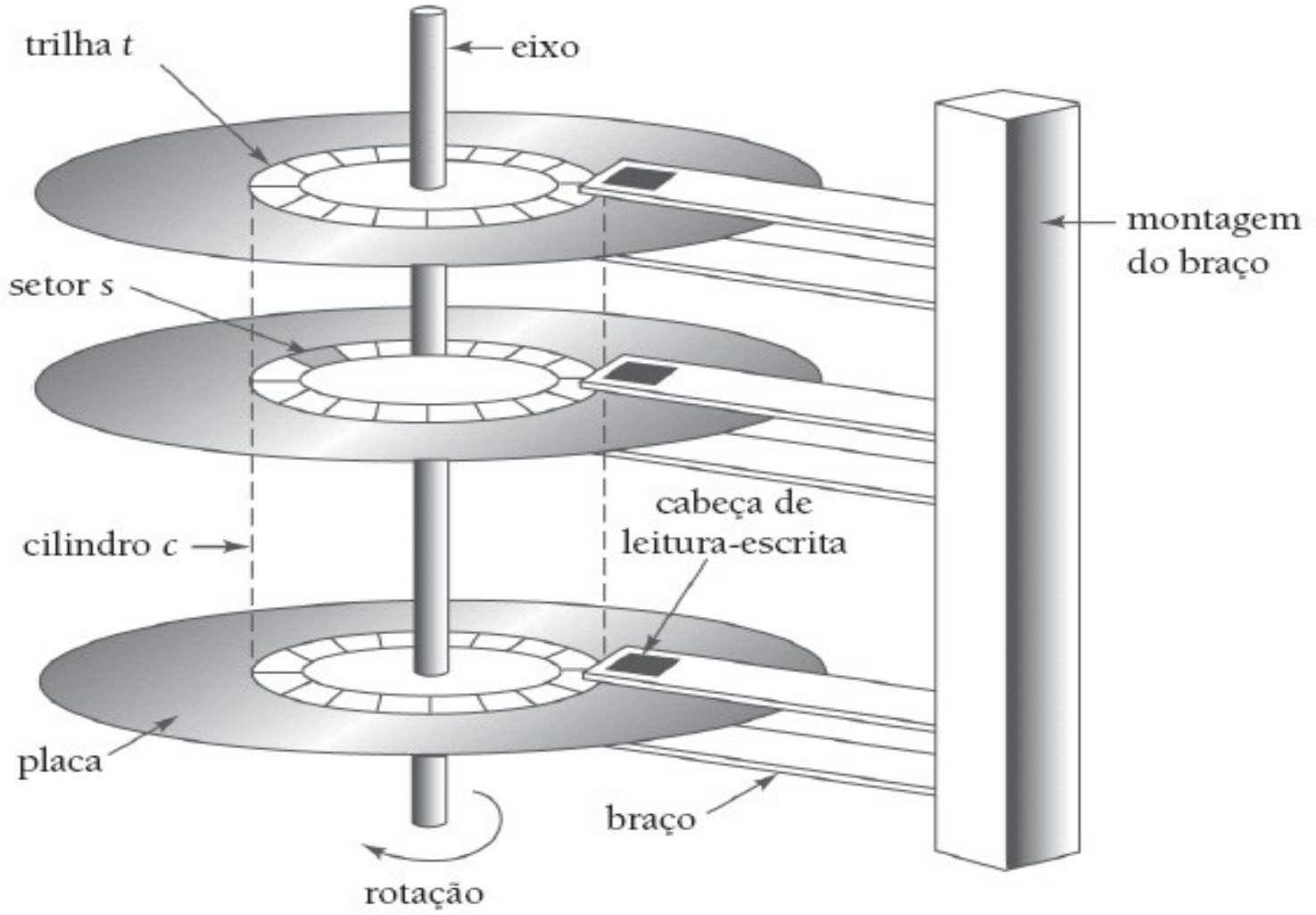
- Largamente utilizadas em dispositivos móveis como câmeras digitais, celulares, etc.
- Usada em substituição a HD em computadores móveis, aumentando ligeiramente o custo
- Principalmente pelo custo, em grandes banco de dados a solução atual ainda é o disco rígido

(Ver: (1)"E. Gal, S. Toledo, Algorithms and Data Structures for Flash Memories, ACM Computing Surveys, Vol. 37, No. 2, June 2005, pp. 138–163"; (2) Web)

Motivações para uso de discos:

- ***Custo***
- ***Capacidade***
- ***Limitações de endereçamento em RAM***
- ***Durabilidade***

Armazenamento de Dados - Estrutura de Discos - Figura



Armazenamento de Dados - Estrutura de Discos - Texto

- ***Setor é uma divisão física de acesso, analogamente, bloco (ou página) é divisão lógica definida por software***
- ***Trilhas podem estar em superfícies de dupla face***
- ***Cilindro é virtual, um conjunto de trilhas***
- ***Cabeças de leitura/gravação movem-se conjuntamente por meio do braço***

Armazenamento de Dados - Estrutura de Discos – Tempo de Acesso

- ***Controladora de disco: interface entre o disco e a memória RAM***
- ***Controle de erro por meio de "check sum" por setor, que é conferido na leitura.***

- ***Tempo acesso =***

  Seek + atraso de rotação + tempo_transferência

- ***OBS: seek e atraso rotação em geral são valores médios***

Armazenamento de Dados – Desempenho de Discos - Proximidade

- ***IO em geral domina o custo***
- ***Otimização depende de localização estratégica dos dados***
- ***Acesso sequencial permite um seek por trilha (ou por cilindro), minimizando o tempo acesso***
- ***Proximidade de blocos***
  - Mesma trilha
  - Mesmo cilindro
  - Cilindros adjacentes
- ***Pré-fetching minimiza tempo médio de acesso***

Armazenamento de Dados - Desempenho de Discos – Exemplo 1

EXEMPLO:

- ***tempo médio de seek = 8 ms***
- ***rotação = 10.000 rpm***
  - 1 rotação completa = 1/10000 minutos = 6 ms
  - média de atraso rotacional = 3 ms
- ***setores por trilha = 170 setores***
- ***tamanho setor = 512 bytes***
- ***transferência = 6 ms / 170 setores = 0,035 ms / setor***

LOGO:

tempo médio de acesso a um setor =

8 + 3 + 0,035 = 11,035 ms

Armazenamento de Dados - Desempenho de Discos - Comparação

COMPARANDO ACESSO ALEATÓRIO COM SEQUENCIAL

- ***DISCO: seek=8; rotação=10.000rpm; trilha com 170 setores de 512 bytes.***
- ***ARQUIVO: 34.000 registros de 256 bytes ocupando 100 trilhas distribuídas aleatoriamente no disco***
- ***Tempo de leitura sequencial x aleatória***

  SEQUENCIAL: (seek+atraso+transferência) por trilha

  Uma trilha = 8 + 3 + 6 = 17ms

  O arquivo = 17 * 100 ms= 1,7 s

  ALEATÓRIA: (seek+atraso+transf. setor) por registro

  Um registro = 1 setor = 11,035ms

  O arquivo 34.000 x 11,035ms = 371,1 s

  OBS: sequencial é 218 vezes mais rápido

- ***Com uma melhor alocação de espaço podemos ter um seek por cilindro, atraso e transferência por trilha (recalcule).***

Armazenamento de Dados - Desempenho de Discos – Exemplo 2

Exemplo no quadro: ***recalcule os tempos anteriores considerando um disco de 5 placas de dupla superfície e alocação ótima.***

OBS: considere primeiramente acesso sequencial e depois acesso paralelo às trilhas de um mesmo cilindro.

Armazenamento de Dados - Desempenho de Discos – Exercício 1

EXERCÍCIO: ***Considere um arquivo contendo um registro para cada habitante do planeta com tamanho igual a 1 setor de um disco de exemplo (use as configuraçõe do HD de seu Desktop e faça as suposições de configuração que não conseguir localizar). Calcule o tempo de leitura de todo o arquivo nas seguintes situações de***

Leitura/ Alocação :

1)aleatória/ aleatória;

2) sequencial/ aleatória;

3) sequencial/ "ótima"; e

4) paralela/ ótima.

OBS: entregar no início da próxima aula.

Armazenamento de Dados - Desempenho de Discos – RAID Motivação

Acesso paralelo às trilhas de um cilindro é de difícil sincronização, uma solução para prover paralelismo e maior confiabilidade em discos é a Tecnologia RAID.

RAID - Redundant Arrays of Independent Disks ou Conjunto Redundante de Discos Independentes

Uma tecnologia para acesso a múltiplos discos!

Armazenamento de Dados - Desempenho de Discos – RAID Espelhamento e Espalhamento

- ***Melhoria da confiabilidade por meio da redundância (Espelhamento-Mirroring).***
- ***Melhoria do desempenho por meio do paralelismo (Espalhamento-Striping)***

Armazenamento de Dados - Desempenho de Discos – Espalhamento em RAID

- ***Espalhamento melhora desempenho***
  - partições de mesmo tamanho distribuídos em discos
  - Para D discos a partição i é escrita no disco (i mod D)
  - Permite leitura em paralelo
  - Partição pode ser por bit ou bloco

Armazenamento de Dados - Desempenho de Discos – Espelhamento/Bit de paridade RAID

- ***Redundância melhora a confiabilidade***
  - Espelhamento ou
- ***Discos de dados com espalhamento + disco de verificação com bit de paridade:***
  - permite reconstrução de discos com falha, por exemplo:
    - ✓Paridade 1 sse número de 1's é ímpar
    - ✓bit do disco que falhou é inferido pelo valor do bit de paridade

Armazenamento de Dados - Desempenho de Discos – NÍVEIS em RAID

- ***RAID nível 0 espalhamento nível de bloco, sem qualquer redundância => melhora write; diminui confiabilidade***
- ***RAID nível 1 espelhamento => melhora confiabilidade***
- ***RAID nível 0 + 1 espelhamento e espalhamento (RAID10)***
- ***RAID nível 2 espalhamento com bits de paridade => melhora confiabilidade***
- ***RAID nível 3 espalhamento por bit com bits de paridade para correção de erro de uma forma otimizada => indentifica disco que falhou***
- ***RAID nível 4 espalhamento por bloco com bits de paridade de uma forma otimizada => explora melhor paralelismo***
- ***RAID nível 5 espalhamento por bloco combinado com bits de paridade distribuídos => elimina gargalo***
- ***RAID nível 6 semelhante ao Raid nível 5, mas armazena informações redundantes extras para proteger contra múltiplas falhas de disco.***

Armazenamento de Dados - Desempenho de Discos – NÍVEIS em RAID - ILUSTRAÇÃO

C = Cópia de Dados

P = Bits de Paridade

Armazenamento de Dados - Desempenho de Discos – NÍVEIS em RAID e suas Indicações

- ***RAID nível 0 caso não haja problemas com perdas***
- ***RAID nível 0 + 1 para pequeno volume de dados e muita gravação (write)***
- ***RAID nível 2 e 4 não são utilizados, pois 3 e 5 substituem***
- ***RAID nível 3 grandes transferências de blocos contíguos***
- ***RAID nível 5 genérico com bom desempenho médio***
- ***RAID nível 6 sistemas que necessitam alta confiabilidade***

Armazenamento de Dados - Desempenho de Discos - MTTF

MTTF (mean-time-to-failure)

- Exemplo de MTTF em um disco: 50000 horas (5,7 anos)
- Em 100 desses discos : 50000/100 horas (21 dias)
- Usando 10 discos de verificação podemos melhorar a MTTF do sistema:

(100 discos de dados + 10 de verificação) > 250anos,

pois o sistema falha se houver falha simultanea de um disco de dados e de um disco de verificação:

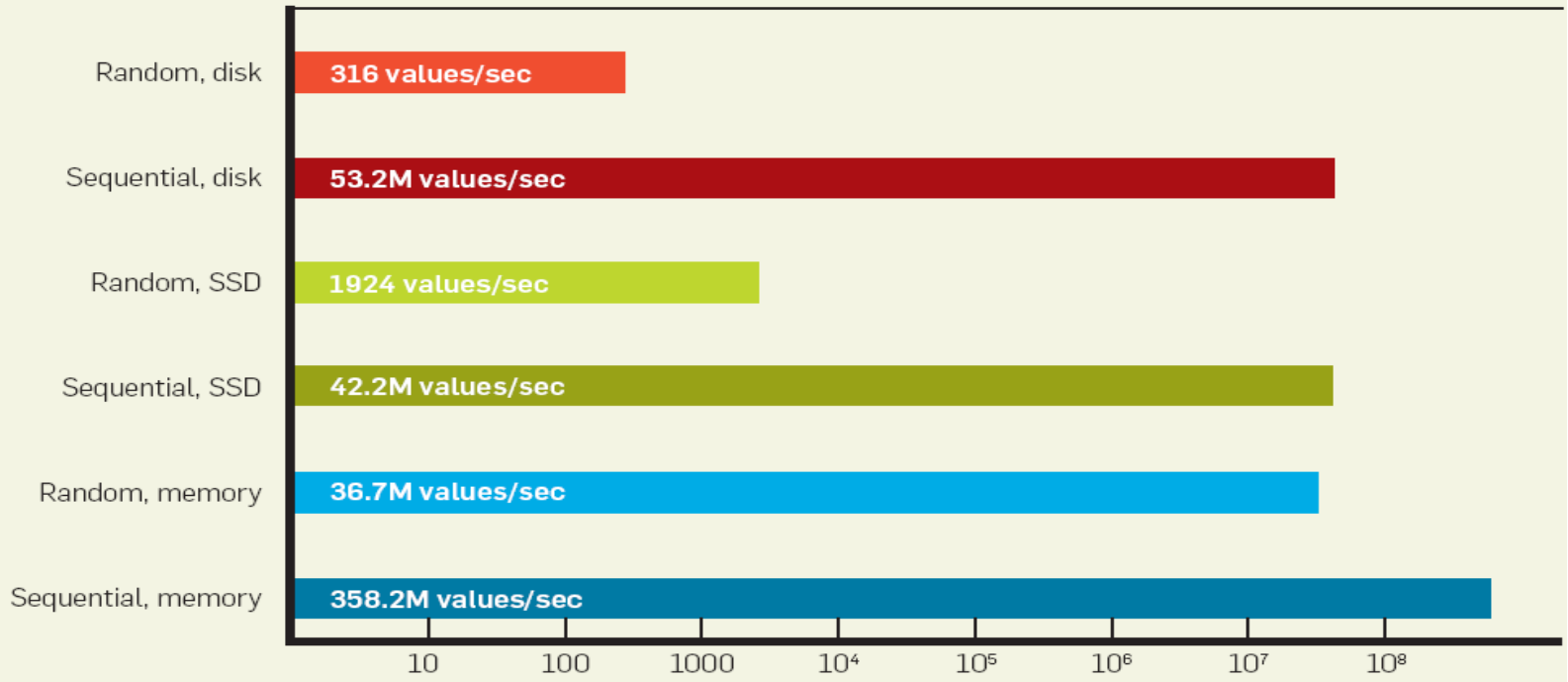
(50000/100) * (50000/10)=2.500.000 horas

Armazenamento de Dados - Desempenho de Discos - MTTR

- ***A tecnologia RAID 5 recupera falha de um disco***
- ***A tecnlogoia RAID 6 pode recuperar falha de mais de um disco***
- ***Entretanto, para calcular precisamente a nova confiabilidade precisamos definir o tempo de reparo do disco (MTTR-Mean Time To Repair)***

Armazenamento de Dados - Desempenho de Discos, Exemplo HD – SSD - RAM

Figure 3. Comparing random and sequential access in disk and memory.



* Disk tests were carried out on a freshly booted machine (a Windows 2003 server with 64GB RAM and eight 15,000RPM SAS disks in RAID5 configuration) to eliminate the effect of operating-system disk caching. SSD test used a latest generation Intel high-performance SATA SSD.

*** Jacobs, A. ,"The Patologies of Big Data", CACM, V.52, N.8, August, 2009**

Gerência de espaço em disco

Armazenamento de Dados – Gerência de Espaço em Disco

- ***Página ou bloco é a unidade de acesso definida pelo software, no caso o SGBD***
- ***Otimização de acesso sequencial é feita por meio de alocação de blocos contíguos (mesma trilha, mesmo cilindro, cilindros adjacentes)***
- ***Modificações podem criar espaços livres***
- ***Gerência de espaços livres pode ser por lista de blocos livres ou bitmap***

Armazenamento de Dados – Gerência de Espaço em Disco

- ***Quem gerencia o espaço?***
  - Sistema operacional ou sistema de arquivos; ou
  - Camada de baixo nível do SGBD
    - ✓ dá maior portabilidade ao sistema e melhora gerência de *buffer pool* (próxima seção)
  - Gerência compartilhada (SO + SGBD)
  - Deixando a alocação física de páginas para camadas de baixo nível, podemos trabalhar com a seguinte abstração:
    - ✓ Arquivo: array de bytes (ou de páginas)
    - ✓ Solicitação: acesso byte *i* (ou página *i*) do arquivo *f*
    - ✓ Execução pelas camadas de baixo nível: acesso ao bloco *m* da trilha *t* do cilindro *c* no disco *d*

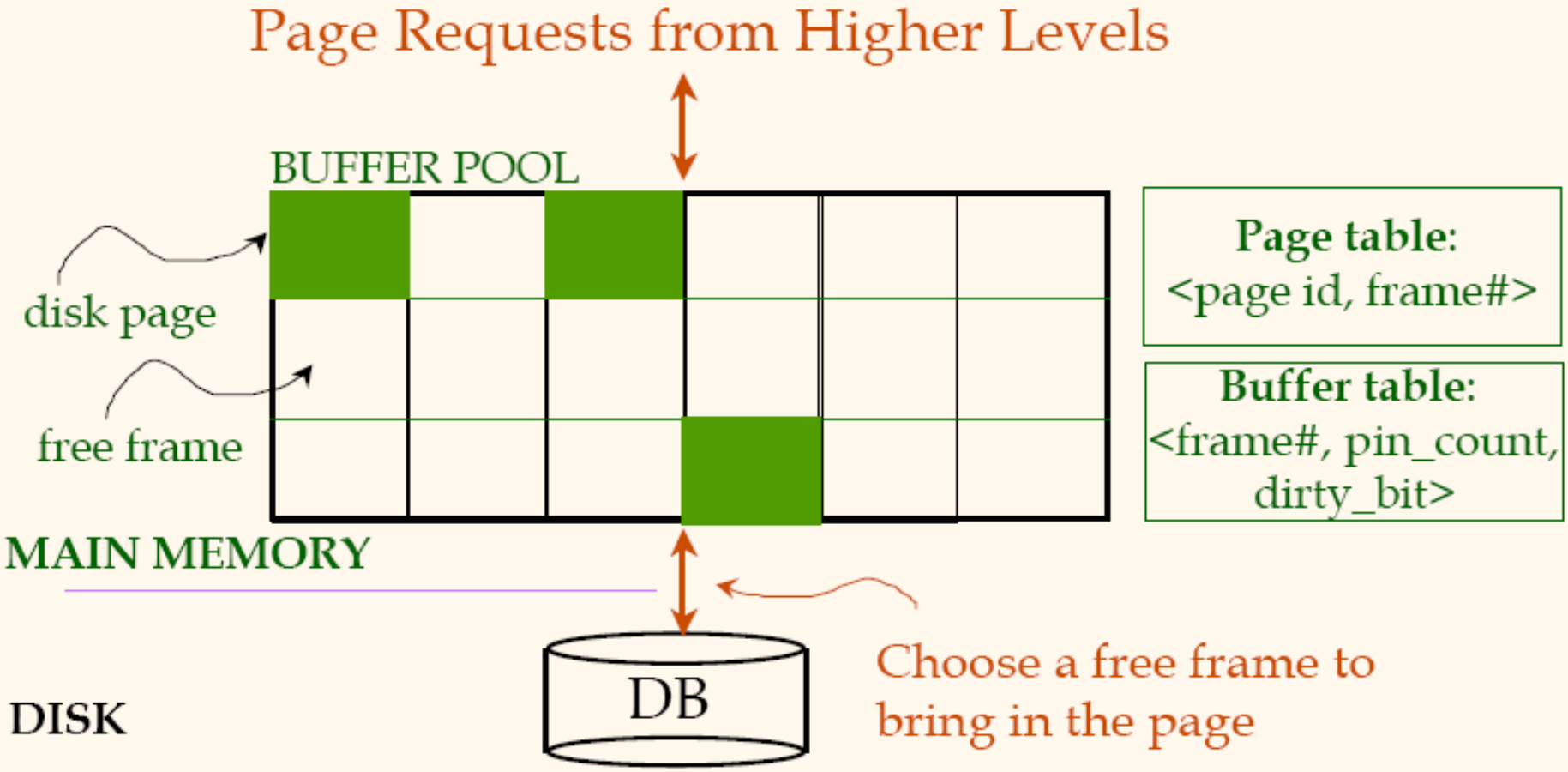
Gerência de buffer pool

Armazenamento de Dados – Gerência de Bufferpool

Motivação

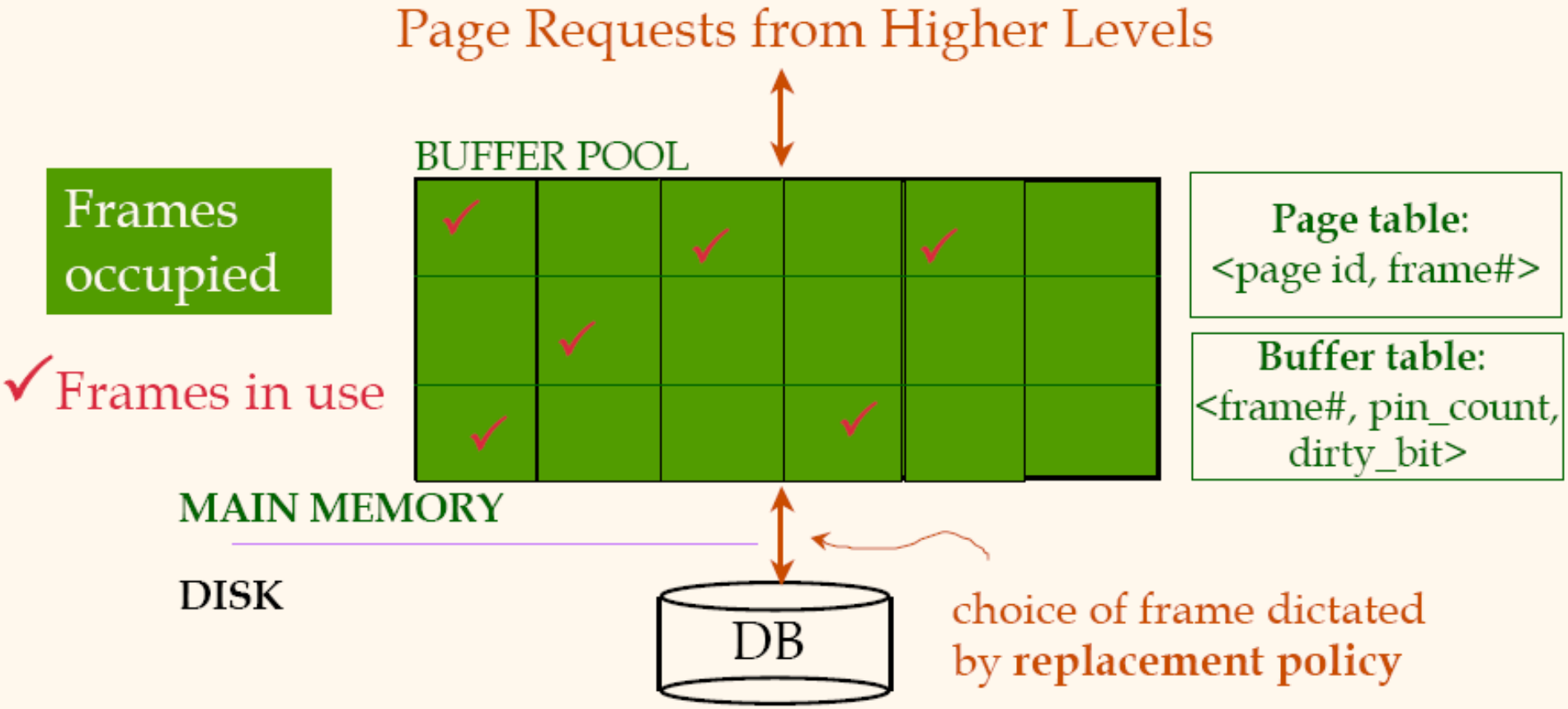
O banco de dados não cabe na memória primária

Armazenamento de Dados – Gerência de Bufferpool



- *Data must be in RAM for DBMS to operate on it!*

Armazenamento de Dados – Gerência de Bufferpool



- *When all frames are occupied, pick one frame not in use using the replacement policy.*

Armazenamento de Dados – Gerência de Bufferpool

Conceitos

- ***Frame ou Slot:*** ***área na memória RAM que será/está ocupada por uma página do disco***
  - pin_count: número de requisições ao frame
  - dirty_bit: indica se o frame foi modificado(1) ou se contem uma imagem do que está no disco(0)

Armazenamento de Dados – Gerência de Bufferpool- Processamento de requisição

(1) SE (EXISTE slot com a página solicitada?)
INCREMENTA pin_count;
RETORNA endereço do slot;
(2) SENÃOSE (EXISTE slot com pin_count == 0?)
ESCOLHE um slot com pin_count == 0
(Usando uma política de substituição);
(3) SENÃO (WAIT e RETORNA EM (2)) ou (ABORTA);
(4) SE (dirty_bit do slot escolhido == 1?)
GRAVA slot na página correspondente no disco;
(5) LÊ página solicitada e GRAVA no slot escolhido
(6) INICIA pin_count do slot com 1
(7) RETORNA endereço do slot escolhido;

Armazenamento de Dados – Gerência de Bufferpool – Fim de Transação e Pré-fetching

Fim de Transação:

- ***Os pin_count de todos os slots em uso pela transação serão decrementados quando a transação termina;***
- ***A transação pode liberar slots durante seu processamento;***

Pré-fetching:

- ***Requisições de páginas podem ser previstos por meio de pre-fetching***

Armazenamento de Dados – Gerência de Bufferpool - Políticas de Substituição

Como escolher slots com pin_count == 0 (?)

- ***FILA CIRCULAR ou ALEATÓRIA:*** ***sem overhead de estrutura, pois basta um contador que é incrementado na fila circular ou é aleatório;***

- ***FIFO:*** ***fila por tempo de entrada na memória;***

- ***LRU (Least Recently Used):*** ***o slot entra em uma fila quando seu pin_count é decrementado para 0;***

- ***MRU(Mosts Recently Used):*** ***o slot entra em uma pilha quando pin_count é decrementado para 0;***

Armazenamento de Dados – Gerência de Bufferpool - Comparação MRU x LRU

- ***Escolha depende do padrão de uso***

- ***Repetidas varreduras sequenciais favorecem MRU, exemplo***

JUNÇÃO$_{s.k=r.k}$ (R, S) : Algoritmo de Laços Aninhados Paginado

```
PARA CADA pr em R
        PARA CADA ps em S
                PARA CADA r em pr {
                        PARA CADA s em ps {
                                SE s.k = r.k imprima (r + s)
                        }
                }
                Liberre o slot de ps;
        }
        Libere o slot de pr;
}
```

Simular MRU e LRU e verificar a inundação sequencial ocorrida em LRU.

Armazenamento de Dados – Gerência de Bufferpool - SGBD x SO

- ***SO usa políticas de paginação para memória virtual***

- ***Mas SGBD pode***
  - Prever padrões de uso
  - Necessita de controle para recuperação de falhas
  - Portabilidade

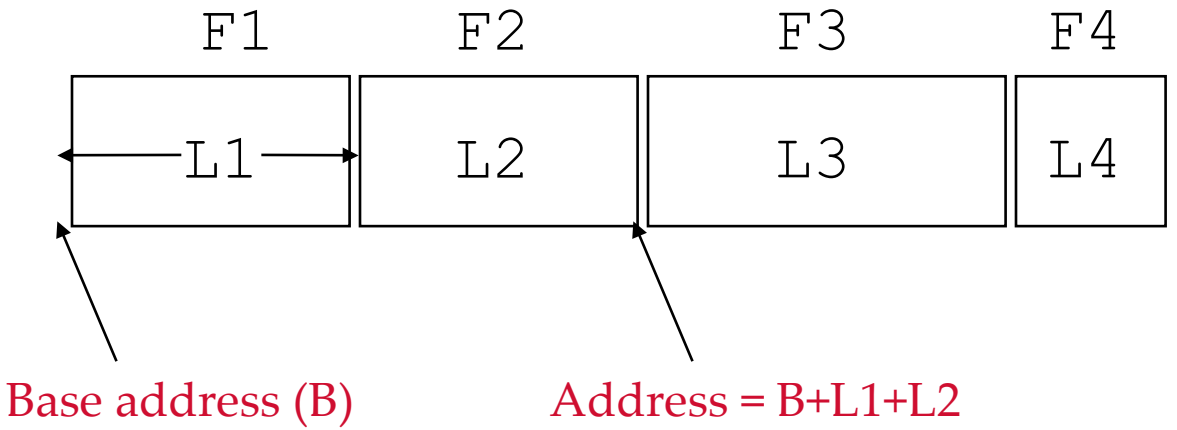
- ***Gerência pode ser compartilhada***

Formatos de registros e páginas

(Como organizar campos em registros e estes em páginas?)

Armazenamento de Dados – Formato de Registros e Páginas - Registro de tamanho fixo

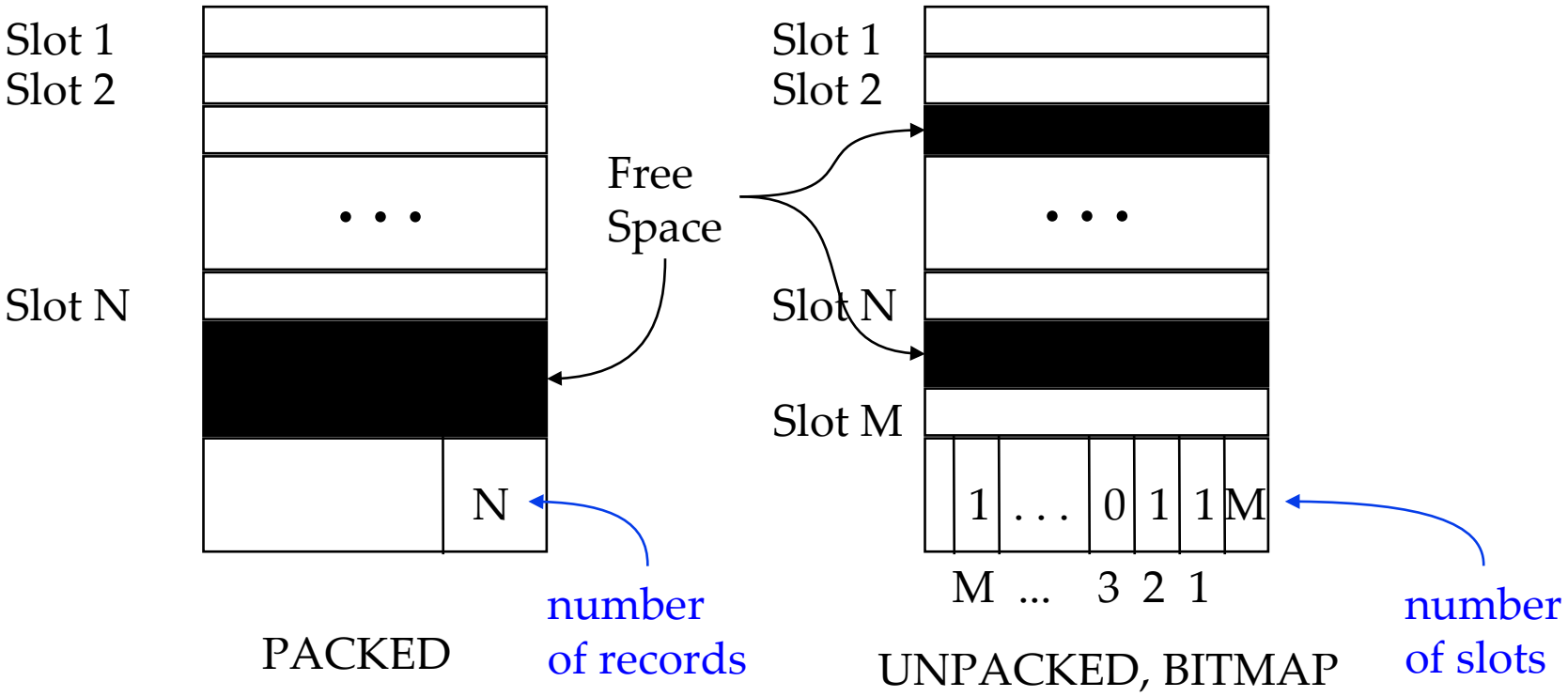
- ***Formato de registro de tamanho fixo***
  - Dados dos campos armazenados no catálogo
  - Localização do campo calculada, exemplo, B+L1+L2(figura)



Armazenamento de Dados – Formato de Registros e Páginas – PACKED e BITMAP

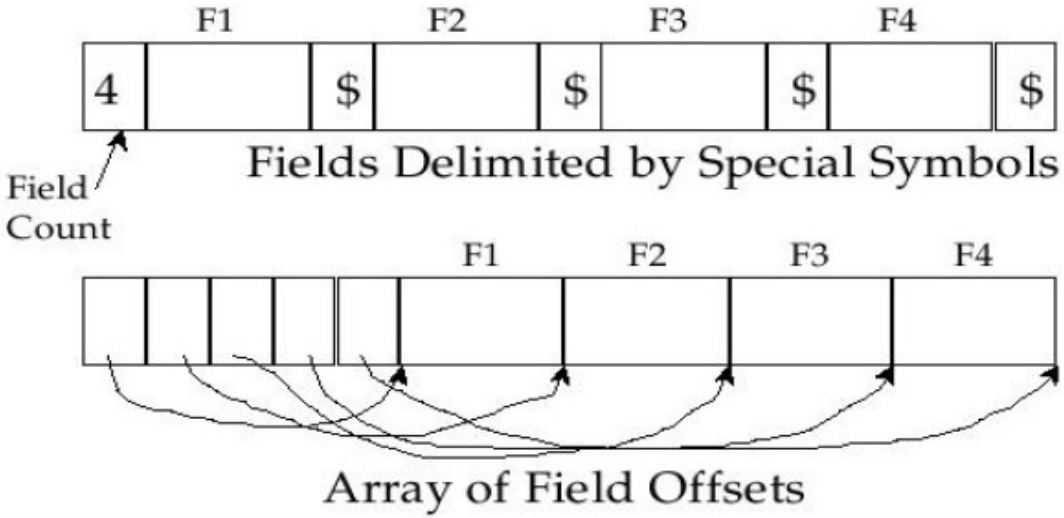
Formatos de página com registros de tamanho fixo

- Localização de um registro no arquivo: **rid=<pid, slot>**
- *PACKED* alteração exige *shift* e altera rid de vários reg
- UNPACKED alteração pode gerar espaços vazios



Armazenamento de Dados – Formato de Registros e Páginas - Reg Tamanho Variável

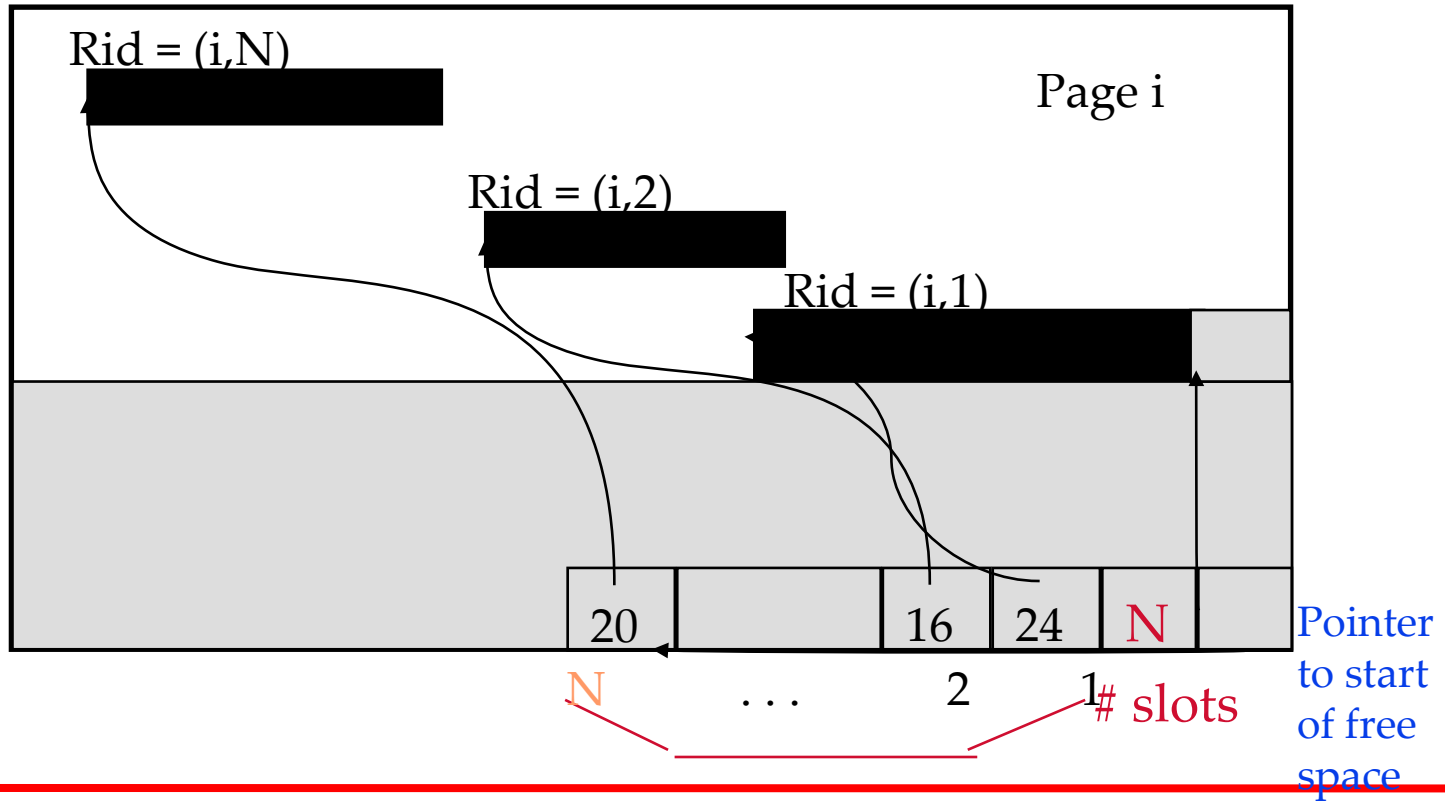
- Alternativas de formato de registros de tamanho variável:
  - ✓ delimitadores
  - ✓ ponteiros
  - ✓ par (tamanho, conteúdo) para cada campo



Armazenamento de Dados – Formato de Registros e Páginas – Páginas com Reg Var.

Formatos de página com registros de tamanho variável

- Move registros na página sem alterar o rid=(pageId, slotId)
- Cada entrada no diretório com (offset, tamanho)

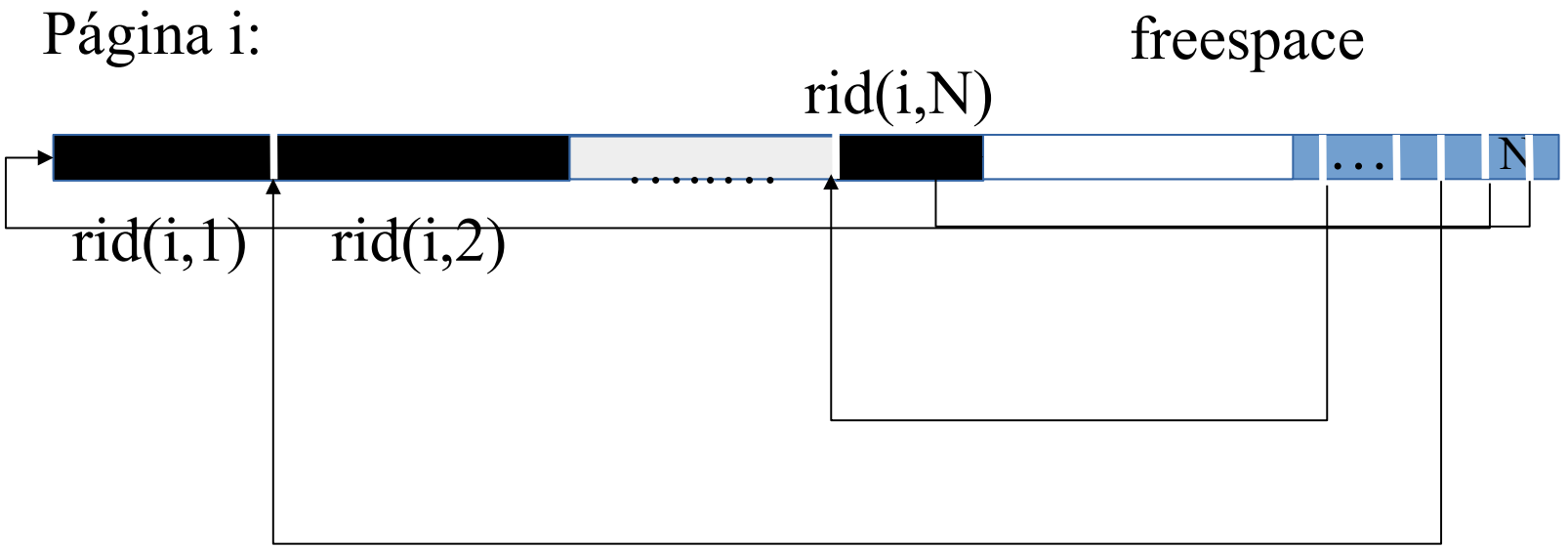


Armazenamento de Dados – Formato de Registros e Páginas – Formato alternativo

Formato alternativo de páginas:

Slot do diretorio com ponteiro para inicio registro

Fim do registro é identificado pelo início do próximo



Armazenamento de Dados – Formato de Registros e Páginas – Operações em Arquivos

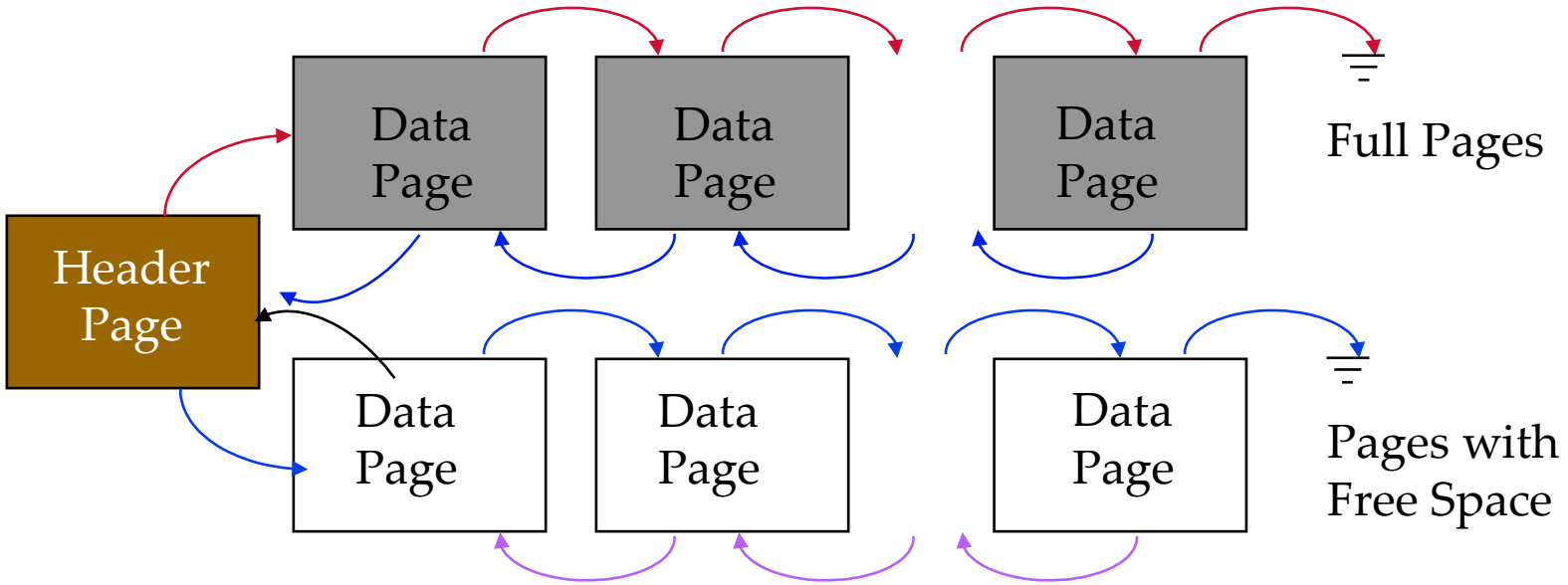
Lembrando que o IO é sempre baseado em páginas

- ***Operações via rid=(pageId, slotId)***
  - inserção;
  - remoção;
  - atualização;
  - leitura.
- ***Operação de varredura sequencial total ou de intervalos;***
- ***Pode haver alocação de novas páginas nas inserções/atualizações;***
- ***Pode haver liberação de páginas nas remoções;***
- ***Problemas:***
  - Controlar sequência de páginas no arquivo
  - Controlar espaços livres na(s) página(s)
  - Controlar registros armazenados na página

Armazenamento de Dados – Formato de Registros e Páginas – Inserção em "Heap"

Arquivo não Ordenado

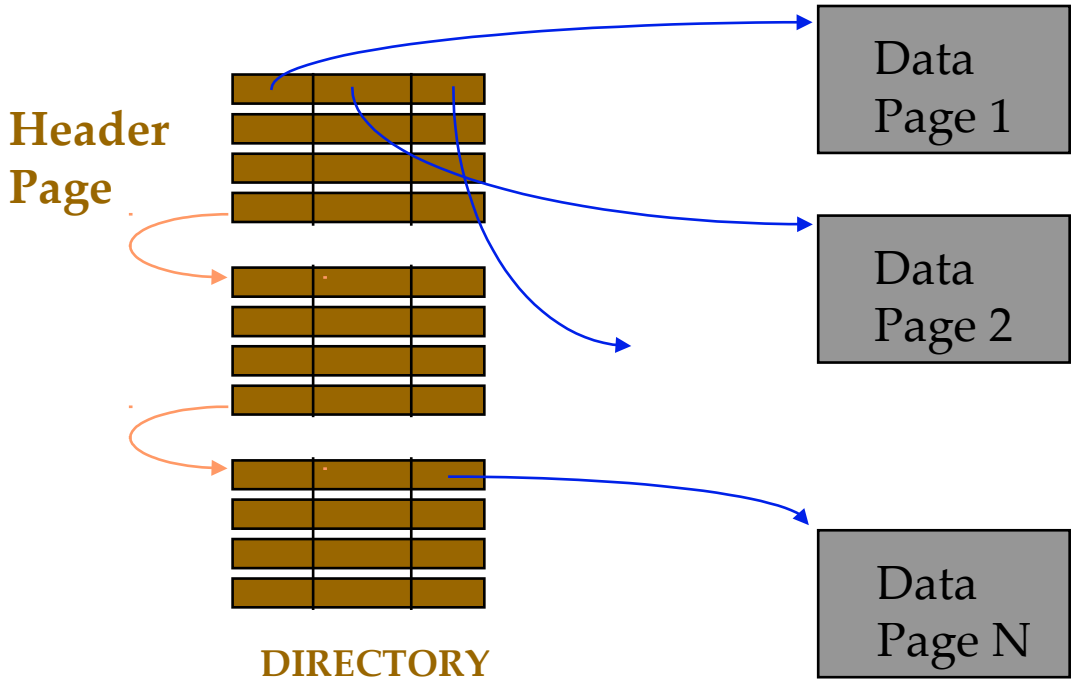
- ***Alternativa 1: Lista de páginas duplamente ligada***
  - nome do arquivo e "Header Page" no catálogo;
  - Inserção: encontrar página que caiba o registro;



Armazenamento de Dados – Formato de Registros e Páginas – Inserção - Diretório

Arquivo não Ordenado

- ***Alternativa 2: diretório de páginas***
  - PageId e bytes livres podem ser armazenados no diretório
  - O diretório, em geral, cabe na memória RAM



Armazenamento de Dados – Formato de Registros e Páginas - O Catálogo

- ***para cada índice: tipo e campos da chave***
- ***para cada relação:***
  - nome, arquivo, tipo (heap, ordenado, etc..)
  - nome e tipo de cada atributo
  - nome de cada índice
  - restrições de integridade
- ***para cada visão: nome e definição***
- ***estatísticas***
- ***autorizações***
- ***tamanho do buffer pool***
- ***o catálogo também é uma relação, portanto, armazenada em arquivo***

Armazenamento de Dados – Formato de Registros e Páginas - Considerações finais

- ***Motivação principal para armazenamento em disco é custo e durabilidade***
- ***Acesso aleatório exige localização da página (seek + atraso rotacional)***
- ***Arranjo das páginas pode minimizar seek e atraso rotacional***
- ***Políticas adequadas para substituição de slots reduzem IO***
- ***Pré-fetch de várias páginas também pode reduzir IO***

Exercícios Capítulo 9 do Livro texto

Trabalho de Implementação

FIM - Armazenamento de Dados