

5) Replacement Sort no passo 0.

$$a) B_1 = \frac{5.000}{2 \times 5} = 500 \text{ sub-arquivos ordenados}$$

$$B_2 = \frac{5.000}{2 \times 50} = 50 \text{ sub-arquivos ordenados}$$

$$B_3 = \frac{5.000}{2 \times 500} = 5 \text{ sub-arquivos ordenados}$$

$$b) B_1 = \left\{ \log_{(4)}^{500} \right\} + 1 = 5 + 1 = 6 \text{ passos}$$

$$B_2 = \left\{ \log_{(4)}^{50} \right\} + 1 = 2 + 1 = 3 \text{ passos}$$

$$B_3 = \left\{ \log_{(4)}^5 \right\} + 1 = 1 + 1 = 2 \text{ passos}$$

$$c) B_1 = 2 \times 5.000 \times 6 = 60.000 \text{ I/Os}$$

$$B_2 = 30.000$$

$$B_3 = 20.000$$

$$5) B = 500$$

$$\text{velocidade média} = 10 \text{ m/s}$$

$$\text{atraso} = 5 \text{ m/s}$$

$$\text{transparência} = 1 \text{ m/s por pg}$$

$$p = \left\lceil \log_{500} \frac{5000}{500} \right\rceil + 1 = 2$$

$$t = (10 + 5 + 1) \times 2 \times 5.000 \times 2 =$$

$$320.000 \text{ ms}$$

$$320 \text{ s}$$

$$10) N = 10.000.000 \text{ p\u00e1gs}$$

External Merge Sort

$$B = 320 \text{ p\u00e1ginas}$$

$$seqs = 10 \text{ ms}$$

$$atrac = 5 \text{ ms}$$

$$transp = 1 \text{ ms}$$

$$p\u00e1gina = 4.096 \text{ bytes}$$

$$\text{Custo I/O} = 16 \text{ ms} \text{ (normal)}$$

$$\text{Custo I/O} = \left(\frac{15}{b+1}\right) \text{ ms} \text{ (fun\u00e7\u00e3o por bloco)}$$

$$b = (\text{p\u00e1ginas do bloco})$$

$$a) \left\lceil \log_{(319)} \left(\frac{10.000.000}{320} \right) \right\rceil = 2 \downarrow \text{ passes}$$

$$\begin{aligned} \text{Custo}_{\text{rec}} &= ((15/1) + 1) = 16 * 10.000.000 * 2 * 2 = \\ &= 640.000.000 \text{ ms} \\ &= 640.000 \text{ s} \downarrow \end{aligned}$$

$$b = 1 \text{ p\u00e1g p/ bloco}$$

b) 256-vias com 1 p\u00e1g entrada, buffer s\u00e1o 64 p\u00e1gs

$$\left\lceil \log_{(256)} \left(\frac{10.000.000}{320} \right) \right\rceil = 2 \text{ passes} \downarrow$$

$$\begin{aligned} \text{Custo Leitura} &= (15 + 1) * 10.000.000 * 2 = \\ &= 320.000.000 \text{ ms} = \\ &= 320.000 \text{ s} \downarrow \end{aligned}$$

$$\begin{aligned} \text{Custo Escrita} &= \left(\frac{15}{64} + 1\right) * 10.000.000 * 2 = \\ &= 24.687.500 \text{ ms} = \\ &= 24.687,5 \text{ s} \downarrow \end{aligned}$$

c) 16- vias, 16 buffers de entrada com 16 págs e
1 buffer saída com 64 págs

$$\left\lceil \log_{(16)} \left(\frac{10.000.000}{320} \right) \right\rceil = 4 \text{ passes} \downarrow$$

$$\text{Custo Leitura} = \left(\frac{15}{16} + 1 \right)^4 \cdot 10.000.000 \cdot 4 =$$

$$= 77.500.000 \text{ ms}$$

$$\text{Custo Escrita} = \left(\frac{15}{64} + 1 \right)^4 \cdot 10^7 \cdot 4 =$$

$$= 49.315.000 \text{ ms}$$

$$= 49.315 \text{ s} \downarrow$$

d) 8- vias, 8 buffers de entrada com 32 págs e
1 buffer saída com 64 págs

$$\left\lceil \log_{(8)} \left(\frac{10.000.000}{320} \right) \right\rceil = 5 \text{ passes}$$

$$\text{Custo Leitura} = \left(\frac{15}{32} + 1 \right)^5 \cdot 10^7 \cdot 5 = 73.437.500 \text{ ms}$$

$$= 73.437,5 \text{ s} \downarrow$$

$$\text{Custo Escrita} = \left(\frac{15}{64} + 1 \right)^5 \cdot 10^7 \cdot 5 = 61.718.750 \text{ ms}$$

$$= 61.718,75 \text{ s} \downarrow$$

e) 4- vias, 4 buffers com 64 págs (entrada) e
1 buffer com 64 págs (saída)

$$\left\lceil \log_{(4)} \left(\frac{10.000.000}{320} \right) \right\rceil = 8 \text{ passes} \downarrow$$

$$\text{Custo Leitura} = \left(\frac{15}{64} + 1 \right)^8 \cdot 10^7 \cdot 5 = 61.718.750 \text{ ms}$$

$$= 61.718,75 \text{ s} \downarrow$$

$$\text{Custo Escrita} = \left(\frac{15}{64} + 1 \right)^8 \cdot 10^7 \cdot 5 = 61.718,75 \text{ s} \downarrow$$

Lista de GBD-5

1) $R(a, b, c, d, e)$

5000 registros

1 pág - 10 registros

Arquivo ordenado por R.a

↳ chave conditida, valores: 0 - 4.999.999

a) Árvore B+ agrupada com chave R.a

b) Hash linear com chave R.a

c) Direto ao arquivo ordenado

d) Árvore B+ agrupada com chave (R.a, R.b)

e) Árvore B+ agrupada com chave R.a

f) Árvore B+ agrupada com chave R.b

g) Hash Linear com chave (R.a, R.b)

h) Árvore B+ agrupada com chave R.b

i) Árvore B+ agrupada com chave (R.a, R.b)

2) Block Nested Loops Join: + Uso de blocos diminui o custo de I/O
+ Le a menor relação caber no buffer, I/O = $M \times N$
- Se nenhuma relação ~~caber~~ ^{caibem} inteira no buffer, impacta o custo de I/O.

Sort-Merge Join: + Muito bom se o arquivo estiver ordenado
+ Pode ser necessário ordenar o arquivo antes

Hash Join: - Tem risco de desempenho se a distribuição não for uniforme
+ Se a relação não cabe no buffer, Hash join lerá muitas vezes que o Block Nested Loops Join

3) Se a condição é desigualdade o Hash Join é o pior, pois tem pior performance para desigualdades e pior tempo de função. Já as outras duas algoritmos tem performances parecidas para desigualdade e podem usar o Índice B+ como estrutura de indexação.

4) Considerando as políticas MRU e LRU, como política de LRU, serão trocadas as páginas menos recentemente usadas, o isso para uma substituição da página no buffer por uma nova, o reaproveitamento é bem baixo. Já, se usarmos do MRU, mais recentemente usado, as primeiras páginas inseridas no buffer podem ser reaproveitadas, já que somente a última página (mais recente) vai ser trocada. Com MRU, o custo de I/O pode diminuir bastante.

5) $R = 10.000$ tuplas, não ordenado, 10 tuplas por página

$$\frac{1 \text{ pag} - 10}{x - 10.000} \Rightarrow x = 1.000$$

$S = 2.000$ tuplas, não ordenado, 10 tuplas por página

$$\frac{1 \text{ pag} - 10}{x - 2.000} \Rightarrow x = 200$$

$S.b$ é chave primária de S

$B = 52$, não paga buffer pool

a) $\text{Custo IO} = 1.000 + 1.000 \times 200 = 201.000 \text{ I/O}$

O número necessário para que o custo não aumente é 3.

b) $\text{Custo IO} = (\text{não cabe nenhuma relação no buffer}) = 1.000 + (200 \times (1.000/50)) = 5.000 \text{ I/O}$

O número necessário é 202.

c) $52 > \sqrt{1.000} = \text{Custo IO} = 3(M+N) = 3.600 \text{ I/O}$

O número necessário é 32.

d)

$\text{Custo IO} = 3(M+N) = 3.600 \text{ I/O}$

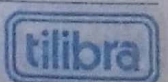
O número necessário é 2 páginas

e) O custo ótimo é $(M+N)$, Local Amishchabel Blockade consegue esse resultado, se a menor relação caber no buffer.

Hash Join Híbrida consegue, se uma das duas relações caber no memória.

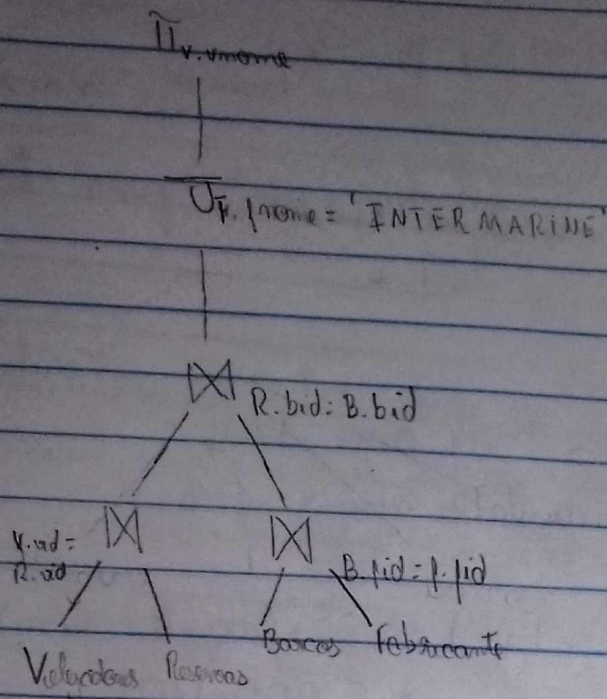
f) O máximo produzido são $10.000 + 2.000 = 12.000$ Tuplas. e serão necessários 1200 págs pro armazenamento no Disco

g) Mudaria que teria menos tuplas, já que seria uma função de $R.a = S.b$.



6) a)

$\pi_{V.name} (\sigma_{F.name = 'INTERMARINE'} (V \bowtie R \bowtie B \bowtie F))$



b) $\pi_{V.name} (on-the-fly)$

$\sigma_{F.name = 'INTERMARINE'}$
(on-the-fly)

\bowtie (Logo Amarelo Bloco)

\bowtie (Logo Amarelo Bloco)

V R B F (Vendedor)

$\pi_{V.name} (on-the-fly)$

(Logo Amarelo Bloco)

$\sigma_{F.name = 'INTERMARINE'}$ (Hash)
(Vendedor)

F
(Vendedor)

c) 1000 pgs do Disco

$$B = 100$$

Redução de F. por nome: 'INTERMEDIATE' é 5%.

plano 1) Loços Bloqueados: $M + N \times (W/B - 2)$

$$\frac{V}{1.000} \times \frac{R}{1.000} \approx 11.204 \text{ } \perp \text{ } T_0$$

$$\frac{B}{1.000} \times \frac{F}{1.000} \approx 11.204 \text{ } \perp \text{ } T_0$$

$$\times \approx 1.211.539 \text{ } T_0 \text{ } \perp \text{ } \frac{\pi}{8} \text{ } \text{---} \text{ } \pi$$

on-the-fly
on-the-fly

plano 2) $F \bar{D} = 50 \text{ } \text{---} \times R \approx 1.050$

$$\frac{B}{11.000} \times \frac{V}{11.000}$$

$$\times \approx 119.164 \text{ } \text{---} \text{ } \pi$$

on-the-fly