

Algoritmos e Estruturas de Dados II

Ordenação Externa III

Prof. Ricardo J. G. B. Campello

Intercalação Multi-Dispositivos

- ◆ O modelo de ordenação externa visto anteriormente assume que se dispõe de um único canal para L/E em um único dispositivo
- ◆ O processamento interno se dá usualmente de forma quase instantânea quando comparado ao tempo de acesso, que representa praticamente todo o tempo de ordenação
- ◆ Uma primeira forma de evitar esse gargalo (*bottleneck*) é utilizar 2 dispositivos independentes, um para leitura e outro para escrita
- ◆ Nesse caso, pode-se ler e escrever em paralelo, o que representa um ganho de 2 vezes no tempo

Intercalação Multi-Dispositivos

- ◆ De forma mais geral, quando se dispõe de $2m$ dispositivos, pode-se utilizar m dispositivos para leitura e m dispositivos para escrita
- ◆ Para tanto, o procedimento Merge-Sort Externo deve ser modificado
- ◆ Ao invés de 2 arquivos de origem (leitura) e 2 arquivos de destino (escrita) se alternando nessas funções, passa-se a trabalhar com m arquivos de origem e m arquivos de destino em $2m$ dispositivos
- ◆ Se usarmos $2m$ buffers de L/E independentes em memória interna, m para leitura e m para escrita, pode-se reduzir em até m vezes os tempos de leitura e de escrita

3

Intercalação Multi-Dispositivos

- ◆ A redução do tempo total t de L/E pode ser então de até $2m$ vezes, ou seja, t pode ser reduzido para algo da ordem de $t/2m$
 - Essa é uma estimativa aproximada de melhor caso
- ◆ Adicionalmente, o no. de passagens requeridas é também reduzido
- ◆ De fato, após cada passagem o tamanho da rodada aumenta em m vezes. Com rodadas iniciais unitárias, são necessárias i passagens, onde $m^i \geq n$, sendo n o número de registros
- ◆ Logo, $O(\log_m n) = O(\log n / \log m)$ passagens são suficientes
- ◆ Isso significa uma redução da ordem de $\log m$ passagens

4

Intercalação Multi-Dispositivos

- ◆ Como exemplo, lembramos que Merge-Sort Externo demanda em torno de 1450 acessos de 330Mb para ordenar um arquivo de 40Gb composto de registros de 1Kb quando se dispõe de 1Gb de RAM
 - 6 passagens pelos arquivos
- ◆ Com $m = 3$, $6/\log_2(3) = 4$ passagens são suficientes
 - porém, isso demanda 6 buffers, ou seja, blocos de 165Mb
 - isso implica em torno de 1940 acessos de 165Mb
 - ◆ mas acessos em parte paralelizados

5

Intercalação Multi-Dispositivos

- ◆ Exemplo: Ordenar por ordem alfabética das chaves o seguinte arquivo com 22 registros:

I N T E R C A L A C A O B A L A N C E A D A
- ◆ Assumindo que temos $2m = 6$ fitas magnéticas independentes e que a memória primária só tem espaço para 3 registros, tem-se após a organização inicial nas 3 fitas de origem:

Fita 1: I N T | A C O | A D E
 Fita 2: C E R | A B L | A
 Fita 3: A A L | A C N

6

Intercalação Multi-Dispositivos

- Na primeira passagem pelas fitas as rodadas de tamanho 3 são intercaladas em rodadas de tamanho 9 direcionadas de forma alternada para as fitas de saída:

Fita 4: A A C E I L N R T

Fita 5: A A A B C C L N O

Fita 6: A A D E

- Mais uma passagem (rodadas de 27 registros) e o processo termina com todos os registros ordenados em uma única fita. As duas outras ficam vazias já que temos apenas 22 registros

Fita 1: A A A A A A A B C C C D E E I L L N N O R T

7

Intercalação Multi-Dispositivos

- O procedimento anterior é chamado **Intercalação Balanceada Multi-Caminhos** (*multi-way balanced merging*) via Múltiplos Dispositivos
- Como em qualquer intercalação multi-caminhos, existe um preço computacional a pagar: a busca interna pela menor dentre m chaves
- Em outras palavras, cada passo do procedimento de fusão não demanda mais apenas a comparação de apenas 2 chaves
- É possível encontrar a menor chave via:
 - $m - 1$ comparações \Rightarrow tempo $O(m)$;
 - Implementação de uma fila de prioridade (heap) \Rightarrow tempo $O(\log m)$

8

Intercalação Multi-Dispositivos

- ◆ Nivio Ziviani sugere que a abordagem de heap passa a compensar a partir de $m \geq 8$ (Ziviani, 2004), pág. 131.
- ◆ De qualquer forma, os ganhos computacionais decorrentes do aumento no número m de dispositivos de armazenamento externo tenderiam a inexistir a partir de um dado limite
- ◆ A partir desse limite o gargalo passaria a ser o tempo de fusão realizado em memória interna, que cresce na ordem $O(m)$ ou $O(\log m)$, dependendo da abordagem adotada para busca da menor chave

9

Intercalação Multi-Dispositivos

- ◆ Uma pergunta que surge é porque precisamos de $2m$ dispositivos externos para uma intercalação de m caminhos
- ◆ Além da paralelização da escrita, a razão é que os dispositivos de leitura e escrita se alternam nessas funções a cada passagem
- ◆ Uma alternativa seria utilizar apenas $m + 1$ dispositivos, m para leitura e 1 para escrita. Bastaria fazer uma passada adicional pelo arq. de escrita para redistribuir as rodadas pelos m arqs. de leitura
- ◆ No entanto, existe um método que elimina a necessidade da passagem adicional para redistribuição de rodadas quando se utiliza $m + 1$ dispositivos: **Intercalação Polifásica**

10

Intercalação Polifásica

- ◆ A idéia é revezar um a um os dispositivos como sendo o dispositivo de saída
- ◆ Especificamente, faz-se com que a cada passagem um dos dispositivos de leitura se esvazie, tornando-se o próximo dispositivo de escrita
- ◆ Nessa abordagem, as rodadas **não** precisam ser distribuídas de forma balanceada entre os dispositivos

11

Intercalação Polifásica

- ◆ Exemplo:
($m = 2$; $n = 34$)

após a passagem	f_1	f_2	f_3
inicial	13 (1)	21 (1)	Ø
1	Ø	8 (1)	13 (2)
2	8 (3)	Ø	5 (2)
3	3 (3)	5 (5)	Ø
4	Ø	2 (5)	3 (8)
5	2 (13)	Ø	1 (8)
6	1 (13)	1 (21)	Ø
7	Ø	Ø	1 (34)

Notação das células: No. de rodadas (No. registros por rodada)

- ◆ A cada passagem as rodadas do dispositivo com menor número de rodadas são intercaladas com um número igual de rodadas dos demais dispositivos
- ◆ O dispositivo correspondente se torna vazio (próximo dispositivo de saída)

Intercalação Polifásica

◆ Exemplo:
($m = 2$; $n = 34$)

após a passagem	f_1	f_2	f_3
inicial	13 (1)	21 (1)	∅
1	∅	8 (1)	13 (2)
2	8 (3)	∅	5 (2)
3	3 (3)	5 (5)	∅
4	∅	2 (5)	3 (8)
5	2 (13)	∅	1 (8)
6	1 (13)	1 (21)	∅
7	∅	∅	1 (34)

Notação das células: No. de rodadas (No. registros por rodada)

- ◆ A distribuição inicial das rodadas pode ser obtida de baixo para cima: tome o maior número de rodadas de uma linha, faça-o igual a zero na linha de cima, e adicione-o aos demais números na mesma linha para obter os números correspondentes da linha acima.

13

Intercalação Polifásica

◆ Exemplo:
($m = 2$; $n = 34$)

após a passagem	f_1	f_2	f_3
inicial	13 (1)	21 (1)	∅
1	∅	8 (1)	13 (2)
2	8 (3)	∅	5 (2)
3	3 (3)	5 (5)	∅
4	∅	2 (5)	3 (8)
5	2 (13)	∅	1 (8)
6	1 (13)	1 (21)	∅
7	∅	∅	1 (34)

Notação das células: No. de rodadas (No. registros por rodada)

- ◆ Nesse caso, exceto pela ausência do primeiro valor unitário, a soma total dos números de rodadas em cada passagem segue uma sequência de Fibonacci: 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

14

Intercalação Polifásica

◆ Exemplo:
($m = 3$)

f_1	f_2	f_3	f_4
Ø	13	11	7
7	6	4	Ø
3	2	Ø	4
1	Ø	2	2
Ø	1	1	1
1	Ø	Ø	Ø

Células: No. de rodadas

- ◆ Conforme ilustrado acima, a idéia pode ser generalizada para qualquer m
- ◆ Nesse caso tem-se uma seqüência de *números de Fibonacci generalizados*

15

Intercalação Polifásica

- ◆ A eficiência depende da distribuição inicial dos registros
- ◆ Segundo (Aho, Hopcroft & Ullman, 1983), para $m = 2$ e rodadas iniciais unitárias, o ideal é se for possível fazer as quantidades de rodadas/registros nos dois dispositivos de origem como números consecutivos de Fibonacci
 - P. ex. 13 e 21, como no exemplo dado anteriormente
- ◆ No caso geral, busca-se por números de Fibonacci generalizados
 - Tais números da 1a linha da tabela de intercalação podem ser obtidos pelo procedimento "de-baixo-para-cima" descrito anteriormente

16

Intercalação Polifásica

- ◆ Quanto menos iterações “de-baixo-para-cima” melhor, já que cada iteração representa uma passagem completa pelos arqs.
 - Quanto menos iterações “de-baixo-para-cima”, menores são os números de rodadas e maiores são os tamanhos dessas rodadas
 - O número de iterações deve ser determinado de forma que as rodadas iniciais sejam do tamanho máximo determinado pela capacidade de pré-ordenação dessas rodadas em memória interna
- ◆ No exemplo anterior para $m = 2$, se 5 registros podem ser ordenados em RAM, podemos iniciar já na 3ª passagem

17

Intercalação Polifásica

- ◆ É possível distribuir as rodadas exatamente segundo nos. de Fibonacci (generalizados) de duas maneiras:
 - Utilizando rodadas de tamanho diferentes
 - ◆ Ver (Ziviani, 2004)
 - Utilizando rodadas fictícias (*dummy runs*)
 - ◆ Ver (Sedgewick, 1997)
- ◆ Ziviani sugere que a intercalação polifásica pode ser mais eficiente que a intercalação balanceada multi-caminhos para m pequeno, i.e., $m \leq 8$
 - Ver (Ziviani, 2004), pág. 135

18

Exercícios

- ◆ Seja um arquivo de 120Gb composto de registros de 500 bytes cada. Supondo que se dispõe de 600Kb de RAM disponível:
 - Qual o máximo no. de registros em cada rodada inicial que se pode constituir para minimizar o no. de passagens requeridas pelo algoritmo de Intercalação Balanceada Multi-Caminhos?
 - Qual é esse no. mínimo de passagens assumindo que se dispõe de $2m = 6$ dispositivos ($m = 3$)?
 - Qual a quantidade b de registros que pode ser transferida em cada acesso assumindo que a RAM disponível será dividida igualmente em buffers independentes para cada dispositivo?
 - Quantos acessos de L/E são realizados pelo algoritmo nesse caso?

19

Exercícios

- ◆ Escolha uma seqüência não ordenada contendo 29 números (chaves) e assuma que temos $2m = 6$ fitas magnéticas independentes e uma memória primária capaz de ordenar no máximo apenas 2 registros:
 - Ilustre em detalhes o procedimento de intercalação balanceada multi-caminhos dessa seqüência mostrando o conteúdo de cada dispositivo após cada passagem
 - ◆ Destaque quais são as rodadas em cada um dos passos

20

Exercícios

- ◆ Escolha uma sequência não ordenada contendo 21 números (chaves) e ilustre o procedimento de intercalação polifásica dessas chaves com $m + 1 = 3$ dispositivos:
 - Mostre o conteúdo de cada um dos dispositivos após cada passagem do algoritmo, destacando quais são as rodadas em cada um deles
 - Inicie com rodadas unitárias e distribua as rodadas iniciais nos dispositivos de forma a obter uma sequência de Fibonacci no que se refere ao no. de rodadas em cada dispositivo ao longo das passagens

21

Bibliografia

- ◆ A. V. Aho, J. E. Hopcroft & J. Ullman, *Data Structures and Algorithms*, Addison Wesley, 1983.
- ◆ N. Ziviani, *Projeto de Algoritmos*, Thomson, 2a. Ed., 2004.
- ◆ R. Sedgewick, *Algorithms in C: Parts 1-4: Fundamentals, Data Structures, Sorting, Searching*, Addison-Wesley, 3rd Ed., 1997.

22