Database Homework 8

10185101210 陈俊潼
2020.6

12.1

First run: store first three element into the memory and sort, then next three element. Here I omit the second attribute of each tuples.

`emu kangaroo wallaby || lion platypus warthog || hyena meerkat zebra || baboon hornbill`

Second run: merge first two sections and last two sections:

`emu kangaroo lion platypus wallaby warthog || baboon hornbill hyena meerkat zebra`

Third run: merge two sections:

`baboon emu hornibill hyena kangaroo lion meerkat platypus wallaby warthog zebra`

12.2

The relational-algebra expression is shown as follows:

$$\Pi_{\text{T.branch\_name}}\left(\Pi_{\text{T.branch\_name, assets}}(\rho_T(branch)) \bowtie_{\text{T.assets}>\text{S.assets}} \Pi_{assets}(\sigma_{\text{S.branch\_city="Brooklyn"}}(\rho_S(branch)))\right) \tag{1}$$

12.3

By calculating, there're 20,000 / 25 = 800 blocks for r_1 and 45,000 / 30 = 1500 blocks for r_2。

Let relation r_1 be the outer loop, r_2 be the inner loop.

Algorithm	Block Transfers	Seeks
Nested-loop join	800 + 20,000 * 1500 = 30,000,800	800 + 20000 = 20,800
Block nested-loop join	800 + 800 * 1500 = 1,200,800 (Worst case, where memory can only store one block each time) 800 + 1500 = 2300 (Best case, where memory can store all inner relations)	800 + 800 = 16,00 (Worst case) 2 (Best case)
Merge join	800 + 1500 = 2300	$\lceil 800/1500 \rceil + \lceil 1500/800 \rceil = 3$
Hash join	Assume there are n partitions. 2 * (800 + 1500) + 800 + 1500 + 2 * 2 * n = 6900 + 4n	$2(\lceil 800/1500 \rceil + \lceil 1500/800 \rceil) = 6$

- ## 12.10

a.

The cost of merge sort is [1] :

$$b_r(2\lceil \log_{M-1}(b_r/M)\rceil + 1) * \text{Block transfer time} + \\ \Big(2\lceil b_r/M\rceil + \lceil b_r/b_b\rceil(2\lceil \log_{M-1}(b_r/M)\rceil - 1)\Big) * \text{Disk seek time} \tag{2}$$

Cost of a seek: $5 * 10^{-3}$ s

Cost of a block transfer: 4KB / 40(MB/s) = $1 * 10^{-4}$ s

Block count the memory can hold (M): 40MB / 4KB = $1 * 10^{4}$

Block count in the relation (b_r): 40GM / 4KB = 10^{7}

By calculating $b_r(2\lceil \log_{M-1}(b_r/M)\rceil + 1)$, the blocks to be transfered = $3 * 10^{7}$.

b_b = 1: cost = $(1*10^{-4})*(3*10^{7})+(5*10^{-3})*(2000+10^{7}*1)$ = 53010 s

b_b = 100: cost = $(1*10^{-4})*(3*10^{7})+(5*10^{-3})*(2000+10^{5}*1)$ = 3510 s

b.

There will be $\lceil \log_{M-1}(b_r/M)\rceil$ runs.

When b_b = 1 or 100, the time of merge pass is the same, which $\approx$ 1.

c.

The new cost for a seek: $1 * 10^{-6}$ s.

b_b = 1: cost = $(1*10^{-4})*(3*10^{7})+(1*10^{-6})*(2000+10^{7}*1)$ = 3010.002s

b_b = 100: cost = $(1*10^{-4})*(3*10^{7})+(1*10^{-6})*(2000+10^{5}*1)$ = 3000.012 s

---

1. *Database System Concepts* Textbook, Page 549 ↩