Machine Learning

Jorell Linsangan - 767816 - linsangj@myumanitoba.ca

April 6, 2016

1. Bitmap indexes

   (a) Create a bitmap for each value of Boolean attribute "evenNum".

$BV(evenNum = Y)$	$BV(evenNum = N)$
1	0
0	1
0	1
0	1
0	1
0	1
0	1
0	1
0	1
0	1

   (b) Create a bitmap vector for each numeric attribute "numDigits"

$BV(numDigits = 1)$	$BV(numDigits = 2)$
1	0
1	0
1	0
1	0
0	1
0	1
0	1
0	1
0	1
0	1

   (c) Create a bitmap vector for each Boolean attribute "endsWith3"

$BV(endsWith3 = Y)$	$BV(endsWith3 = N)$
0	1
1	0
0	1
0	1
0	1
1	0
0	1
0	1
1	0
0	1

   (d) Find the rIDs of all the records with 2-digit values that end with 3.

$$BV(numDigits = 2) = 0000111111$$
$$BV(endWith3 = Y) = 0100010010$$
$$BV(Result) = 0000010010$$

   Which means that the rids of records with 2-digit values that end with 3 are 6 & 9.

2. I/O Cost

   (a) 2-way external mergesort.

$$\begin{aligned} Cost &= 2(N)(1 + \lceil \log_2(N) \rceil) \\ &= 2(200)(1 + \lceil \log_2(200) \rceil) \\ &= 400(1 + \lceil \log_2(200) \rceil) \\ &= 400 + 3200 \\ \text{Total Cost} &= 3600 \text{ I/Os} \end{aligned}$$

   (b) General multi-way external mergesort with B = 15

$$\begin{aligned} Cost &= 2(N)(1 + \lceil \log_{B-1}(\lceil \frac{N}{B} \rceil) \rceil) \\ &= 2(200)(1 + \lceil \log_{14}(\lceil \frac{200}{15} \rceil) \rceil) \\ &= 400(2) \\ \text{Total Cost} &= 800 \text{ I/Os} \end{aligned}$$

3. Something...Not sure what to name this question. Query Processing? If there are 2000 tuples that are sorted in ascending order and a page can hold 10 student tuples. $P_r = 200$.

   (a) Binary Search. I am assuming that sID is the primary key, therefor it is unique.

$$\begin{aligned} \text{Cost to locate tuple} &= \lceil \log_2(200) \rceil \\ &= 8 \text{ I/Os} \end{aligned}$$

   (b) 3 I/Os

   (c) 4 I/Os

4. (a) Root to leaf traversal is 3. There are 200 pages and there are 2000 tuples where 2% of those are 'G. Raymond'. So $3 + (.02)200 = 7$ I/Os for clustered index.

   (b) For non-clustered, it is $3 + (0.02)2000 = 43$ I/Os.

5. (a) We create a tuple of only the wanted attribute/s which is 'sName'. Let this tuple be tuple T. Since size(sName) == size(sID), we can assume that we can fit 20 student tuples in a page now. So $P_t = 2000/20 = 100$ and $P_E = 2000/10 = 200$. The formula for cost is:

$$P_E + P_T + (2(P_T)\lceil(\log_{B-1}\lceil \frac{P_T}{B} \rceil)\rceil - 1) + P_T$$

$$200 + 100 + (2(100)\lceil(\log_{20}\lceil \frac{100}{21} \rceil)\rceil - 1) + 100 = 400$$

   (b) $P_r + 2P_T$ is the formula to get the number of I/Os for hash based projection. $P_r = 200$ and depending if I should assume that size(sName) == size(sID). P_T will have different values. $P_T = 100$ if I make the assumption or $P_T = 200$ if no assumption.

   Answer with assumption: $200 + (2)(100) = 400$
   Answer without assumption $200 + (2)(200) = 600$

6. $P_S = 200$, $P_E = 800$

   (a) Inner: Enrolled, Outer: Student. Cost: $P_S + N_S * P_E = 200 + 2000(800) = 1600200$.

   (b) Inner: Student, Outer: Enrolled. Cost: $P_E + N_R * P_S = 800 + 8000(200) = 1600800$.

   (c) $P_E + P_E(P_S) = 800 + 800(200) = 160800$

   (d) "Block" nested-loop join.
   With E as outer. $B - 2 = 19$ pages of E per chunk.
   Cost of scanning E: $P_E = 800$ pages.
   Number of chunks $= \lceil \frac{800}{19} \rceil = 43$ chunks.
   Per chunk of E, we scan S: $\lceil \frac{P_E}{B-2} \rceil * P_S = 43(200)$ pages.
   Total cost $= 800 + 8600 = 9400$ pages.

   (e) Sort-merge join.
   Cost for sorting enrolled is $2(800) * (1 + \lceil \log_{20} \lceil \frac{800}{21} \rceil \rceil) = 4800$
   The cost for the merging phase is $P_E + P_S = 800 + 200 = 1000$.
   Cost for Sort-merge join is $4800 + 1000 = 5800$.

7. Schedules

   (a) Schedule 1:

      i. Yes

      ii. Yes

T1	T2
R(A)	
W(A)	
R(B)	
W(B)	
	R(A)
	W(A)

      iii. Yes

T1	T2
R(A)	
W(A)	
R(B)	
W(B)	
	R(A)
	W(A)

   (b) Schedule 2:

      i. No

      ii. Yes

T1	T2
	R(C)
	W(C)
	R(D)
	W(D)
R(C)	
W(C)	

iii. Yes

T1	T2
	R(C)
	W(C)
	R(D)
	W(D)
R(C)	
W(C)	

(c) Schedule 3:

i. No

ii. No

iii. Yes

T1	T2
R(E)	
W(E)	
	W(E))
	W(E)

(d) Schedule 4:

i. No

ii. No

iii. No