CS 525 – QUIZ II

SHANMUGA PRIYA ELLAPPAN

A20465677

Question 1.1.1 Estimate Result Size (5 Points)

Estimate the number of result tuples for the query q = $\sigma_{\text{instructor=Bob}}$ (course) using the first assumption presented in class (values used in queries are uniformly distributed within the active domain).

T(q) = T(course)/V(course, instructor)

= 1000/500

= 2 tuples

Question 1.1.2 Estimate Result Size (5 Points)

Estimate the number of result tuples for the query q = $\sigma_{\text{major=CS V major=EE}}$ (students) using the first assumption presented in class.

T(q) = T (student)/V (student, major) $_{CS}$ + T (student)/V (student, major) $_{EE}$

= 40,000/50 + 40,000/50

= 800 + 800

= 1600 Tuples

Question 1.1.3 Estimate Result Size (7 Points)

Estimate the number of result tuples for the query q = $\sigma_{\text{credits}\geq 29 \wedge \text{credits}\leq 40}$ (course) using the first assumption presented in class.

T(q) = [(40 – 29 + 1) * T(course)] / [max(course, credits) – min (course, credits) + 1]

= [12 * 1000] / [60 – 0 + 1]

= 197 Tuples

Question 1.1.4 Estimate Result Size (8 Points)

Estimate the number of result tuples for the query q below using the first assumption presented in class.

q = $\sigma_{credits \leq 30}$(student) $\bowtie_{CWID=student}$ registered $\bowtie_{course=title}$ course

q1 = $\sigma_{credits \leq 30}$(student)

T(q1) = [(30 – min (student, credits) + 1) * T(student)] / [max(student, credits) – min (student, credits) + 1]

= [(30 – 0 + 1) * 40,000] / [60 – 0 + 1]

= (31 * 40,000) / 61

= 20,328 Tuples

q2 = q1 $\bowtie_{CWID=student}$ registered

T(q2) = T(q1) * T(registered) / max[V(q1, CWID), V(registered, student)]

= 20,328 * 20,000 / max (20328, 5000)

= 20,000 Tuples

q = q2 $\bowtie_{course=title}$ course

T(q) = T(q2) * T(course) / max[V(q2, course), V(course, title)]

= 20,000 * 1000 / max(100, 1000)

= 20,000 Tuples

Question 1.2.1 External Sorting (5 Points)

You have a block of size of 128KB and 2GB memory. Compute the maximum size of file that can be sorted by 3-Pass Multiway Sort.

Memory in KB = 2 * 1024 * 1024 = 2,097,152 KB

M = 2,097,152 / 128

= 16,384 buffers

B(R) <= M * $(M-1)^2$ blocks

<= 16,384 * $(16,384 - 1)^2$ * 128 KB

<= 32,766 GB

Question 1.2.2 I/O Cost Estimation (25 Points)

Consider two relations R and S with B(R) = 400,000 and B(S) = 2,000 blocks, and S(R) = 1/40 , and S(S) = 1/10 . You have M = 101 memory pages available. Let R has an index on the joining attribute C. Compute the minimum number of I/O operations needed to join these two relations using tuple-based-nested-loop join (relations are not clustered), block-nested-loop join (relations are clustered), merge-join (the inputs are not sorted but clustered), and non-clustering index-join (read of clustered relation S with uniform distribution assumption and expected 20 matching tuples on R for S) and hash-join (relations are clustered but not sorted). You can assume that the hash function evenly distributes keys across buckets. Justify you result by showing the I/O cost estimation for each join method.

T(R) = 400,000 * 40 = 16,000,000

T(S) = 2000 * 10 = 20,000

- Tuple based nested loop join:

  S is smaller, thus let's consider it as outer relation

  Total Cost = T(S) * (1 + T(R))

  = 20,000 * (1 + 16,000,000)

  = 320,000,020,000 I/Os

- Block nested loop join:

  S is smaller. Thus, let's consider 100 buffers for S and 1 buffer for R

  Read chunk of S = 100 I/Os

  Read of R = 400,000 I/Os

  # of S chunks = B(S) / M-1

  = 2000/100

  = 20

  Total cost = 20 * (100 + 400,000)

  = 8,002,000 I/Os

- Merge Join:

  Sort cost of R (3 Pass Multiway join algorithm) = 6 * B(R)

  Sort cost of S (2 Pass Multiway join algorithm) = 4 * B(S)

  Total cost = sort cost + join cost

  = 7 * B(R) + 5 * B(S)

  = 7 * 400,000 + 5 * 2000

  = 2,800,000 + 10,000

  = 2,810,000 I/Os

- Non clustering Index join:

  T(S) + [T(S). T(R)/V(R,C)] = 20,000 + [20,000 * 20]

  = 20,000 + 400,000

  = 420,000 I/Os

- Hash-Join (relations are clustered but not sorted):

  3 (B(R) + B(S)) = 3 (400,000 + 2000)

  = 1,206,000 I/Os