Review of Query Optimization

Discussion Session 11

Joins

- Simple Nested Loop

  |R| + ||R|| * |S|

- Page oriented nested loop

  |R| + |R| * |S|

- Block nested loop

  $$|R| + |S| * \left\lceil \frac{|R|}{B-2} \right\rceil$$

- Index nested loop

  |R| + (||R|| * cost of finding matching S tuples)

- Sort merge

Sort: $2.|R|.(1+\lceil \log_{B-1} \lceil |R|/B \rceil \rceil) + 2.|S|.(1+\lceil \log_{B-1} \lceil |S|/B \rceil \rceil)$

Merge = (|R|+|S|)

- Grace Hash join

  3(|R| + |S|) I/Os.

Ex 14.3 – 1 (Solution)

- Consider processing the following SQL projection query:

SELECT DISTINCT E.title, E.ename FROM Executives E

You are given the following information:

  - Executives has attributes ename, title, dname, and address; all are string fields of the same length.
  - The ename attribute is a candidate key.
  - The relation contains 10,000 pages.
  - Each page contains 10 tuples
  - There are 10 buffer pages.

- How many sorted runs are produced in the first pass? What is the average length of these runs? (Assume that memory is utilized well and any available optimization to increase run size is used.)

- Replacement sort can be used as an optimization.
- Replacement sort will generate runs of 16 pages.
- Since sorting pass creates sorted runs of tuples containing only attributes ename and title. So the total number of resultant pages are 5000.
- Thus, pass 0 generates 312 runs of 16 pages each, and 1 run of 8 pages

Ex 14.3 – 1

- Consider processing the following SQL projection query:

SELECT DISTINCT E.title, E.ename FROM Executives E

You are given the following information:

  - Executives has attributes ename, title, dname, and address; all are string fields of the same length.
  - The ename attribute is a candidate key.
  - The relation contains 10,000 pages.
  - Each page contains 10 tuples
  - There are 10 buffer pages. .

- What is the I/O cost of this first sorting pass?

- 10000 pages are read into memory
- Since the sorting pass eliminates dname and address, it writes out only 5000 pages.
- Total number of I/Os are 15000

Ex 14.3 – 2

- Consider processing the following SQL projection query:

SELECT DISTINCT E.title, E.ename FROM Executives E

You are given the following information:

  - Executives has attributes ename, title, dname, and address; all are string fields of the same length.
  - The ename attribute is a candidate key.
  - The relation contains 10,000 pages.
  - Each page contains 10 tuples
  - There are 10 buffer pages.

- How many additional merge passes are required to compute the final result of the projection query? What is the I/O cost of these additional passes?

- # of passes = ceil ($\log_9$ (313)) = 3
- # of I/Os = 2 * N * # of passes = 2 * 3 * 5000 = 30000 I/Os

Ex 14.5

- Consider the join $R \bowtie_{R.a=S.b} S$ given the following:
  - Relation R contains 10,000 tuples and has 10 tuples per page.
  - Relation S contains 2000 tuples and also has 10 tuples per page.
  - Attribute b of relation S is the primary key for S.
  - Both relations are stored as simple heap files.
  - Neither relation has any indexes built on it.

- If only 15 buffers were available, what would be the cost of a sort-merge join? What would be the cost of a hash join?

Ex 14.5 Solution

- Sort Merge with 15 buffers
  - R has 1000 pages. It requires 3 passes for sorting.
    - Sorting cost = 2 * 3 * 1000 = 6,000 I/Os
  - S has 200 pages. It requires 2 passes for sorting.
    - Sorting cost = 2 * 2 * 200 = 800 I/Os
  - Merge cost = 1000 + 200 = 1,200 I/Os
  - Total cost = 8,000 I/Os

Ex 14.5 Solution

- Hash Join with 15 buffers
  - First partitioning phase: Split S into 14 buckets each containing about 15 pages.
    - To store one partition we need f * 15 pages which is not available. Here f is the fudge factor to capture the notion that partitioning into buckets is not perfect.
  - Second partitioning phase: Scan to partition first phase partitions
    - Entire partitions can now fit into the memory
  - Similarly, we need two phases to create partitions of R to mirror S's partition scheme.
  - Total cost = Cost of two partitioning phases + Cost of matching phase = 2 * (2 * (M + N)) + (M + N) = 6,000 I/Os

Ex. 15.5 - 1

Consider the query $\Pi_{A,B,C.D}$ (R $\bowtie_{A=C}$S). Suppose that the projection routine is based on sorting and is smart enough to eliminate all but the desired attributes during the initial pass of the sort and also to toss out duplicate tuples on the fly while sorting, thus eliminating two potential extra passes. Finally, assume that you know the following:

- R is 10 pages long, and R tuples are 300 bytes long.
- S is 100 pages long, and S tuples are 500 bytes long.
- A is a key for R.
- The page size is 1024 bytes.
- Each S tuple joins with exactly one R tuple.
- The combined size of attributes A, B, C, and D is 450 bytes.
- A and B are in R and have a combined size of 200 bytes; C and D are in S.
- No record spans two pages.

- What is the cost of writing out the final result?

Ex. 15.5 -1 Solution

- R has 30 tuples (10 pages of 3 records each)
- S has 200 tuples (100 pages of 2 records each).
- Since every S tuple joins with exactly one R tuple, there can be at most 200 tuples after the join.
- Since the size of the result is 450 bytes/record, 2 records will fit on a page. This means 200 / 2 = 100 page writes are needed to write the result to disk.

Ex. 15.5 – 2(a)

Consider the query $\Pi_{A,B,C.D}$ (R $\bowtie_{A=C}$S). Suppose that the projection routine is based on sorting and is smart enough to eliminate all but the desired attributes during the initial pass of the sort and also to toss out duplicate tuples on the fly while sorting, thus eliminating two potential extra passes. Finally, assume that you know the following:

- R is 10 pages long, and R tuples are 300 bytes long.
- S is 100 pages long, and S tuples are 500 bytes long.
- A is a key for R.
- The page size is 1024 bytes.
- Each S tuple joins with exactly one R tuple.
- The combined size of attributes A, B, C, and D is 450 bytes.
- A and B are in R and have a combined size of 200 bytes; C and D are in S.
- No record spans two pages.

- Compute the cost of doing the projection followed by the join. (Assume 3 buffer pages and simple (page-oriented) nested loops)

Ex. 15.5 – 2(a) Solution

- Cost of projection followed by join
- Projection cost for R using 3 buffers
  - First pass to remove unwanted attributes and create 2 runs of 3 pages
    - Read 10 pages of 300 bytes
    - Write 6 pages of 200 bytes (each page has 5 tuples)
  - Need 1 additional passes to merge the 2 runs
    - Cost = 2 * 1 * 6 = 12 I/Os
- Project cost of S of using 3 buffer pages
  - First pass to remove unwanted attributes and create 16 runs of three pages and 1 run of two pages
    - Read 100 pages of 500 bytes
    - Write 50 pages of 250 bytes (each page has 4 tuples)
  - Need 5 additional passes to merge the 17 runs
    - Cost = 2 * 5 * 50 = 500 I/Os

Ex. 15.5 – 2(a) Solution

- Cost of SNL join = 6 + 6 * 50 = 306
- Total cost = Projection cost of R + Projection cost of S + SNL Merge cost = (16 + 12) + (150 + 500) + 306 = 984

Exercise 15.5 – 2(b)

Consider the query $\Pi_{A,B,C.D}$ (R $\bowtie_{A=C}$S). Suppose that the projection routine is based on sorting and is smart enough to eliminate all but the desired attributes during the initial pass of the sort and also to toss out duplicate tuples on the fly while sorting, thus eliminating two potential extra passes. Finally, assume that you know the following:

- R is 10 pages long, and R tuples are 300 bytes long.
- S is 100 pages long, and S tuples are 500 bytes long.
- A is a key for R.
- The page size is 1024 bytes.
- Each S tuple joins with exactly one R tuple.
- The combined size of attributes A, B, C, and D is 450 bytes.
- A and B are in R and have a combined size of 200 bytes; C and D are in S.
- No record spans two pages.

- Compute the cost of doing the join followed by the projection. (Assume 3 buffer pages and simple (page-oriented) nested loops)

Ex. 15.5 – 2(b) Solution

- Cost of join followed by projection
  - Cost of SNL join: 10 + 10*100 = 1010 I/Os
  - SNL results in 200 tuples each of size 800 bytes. Thus, the result has 200 pages. 200 I/Os to write the result back.
  - Projection cost using 3 buffer pages:
    - First pass to remove unwanted attributes and create 33 runs of 3 pages and 1 run of a page
      - Read 200 pages of 800 bytes
      - Write 100 pages of 450 bytes (each page has two tuples)
    - Need 6 additional passes to merge the 33 runs
      - Cost = 2 * 6 * 100 I/Os
  - Total cost = 1010 + 200 + (200 + 100 + 1200) = 2710 I/Os

Ex 15.5 – 2(c)

Consider the query $\Pi_{A,B,C.D}$ (R $\bowtie_{A=C}$S). Suppose that the projection routine is based on sorting and is smart enough to eliminate all but the desired attributes during the initial pass of the sort and also to toss out duplicate tuples on the fly while sorting, thus eliminating two potential extra passes. Finally, assume that you know the following:

- R is 10 pages long, and R tuples are 300 bytes long.
- S is 100 pages long, and S tuples are 500 bytes long.
- A is a key for R.
- The page size is 1024 bytes.
- Each S tuple joins with exactly one R tuple.
- The combined size of attributes A, B, C, and D is 450 bytes.
- A and B are in R and have a combined size of 200 bytes; C and D are in S.
- No record spans two pages.

- Compute the cost of doing the join first and then the projection on-the-fly. (Assume 3 buffer pages and simple (page-oriented) nested loops)

Ex 15.5 – 2(c) Solution

- Cost of join and projection on the fly:
  - Projection on the fly means we do not have a specific step to do projection. When a join operation is taking place, we also eliminate values in that process. So, there is not specific cost associated with projection.
  - This means that the projection cost is 0, so the only cost is the join, which we know from above is 1010 I/Os.

Ex 15.5 – 2(d)

Consider the query $\Pi_{A,B,C.D}$ (R $\bowtie_{A=C}$S). Suppose that the projection routine is based on sorting and is smart enough to eliminate all but the desired attributes during the initial pass of the sort and also to toss out duplicate tuples on the fly while sorting, thus eliminating two potential extra passes. Finally, assume that you know the following:

- R is 10 pages long, and R tuples are 300 bytes long.
- S is 100 pages long, and S tuples are 500 bytes long.
- C is a key for S, and A is a key for R.
- The page size is 1024 bytes.
- Each S tuple joins with exactly one R tuple.
- The combined size of attributes A, B, C, and D is 450 bytes.
- A and B are in R and have a combined size of 200 bytes; C and D are in S.
- No record spans two pages.

- Would your answers change if 11 buffer pages were available.

Ex 15.5 – 2(d) Solution

- If we had 11 buffer pages, then the projection sort could be done log10 instead of log2.