CS3223 Tutorial 5: Join Algorithms

Wk 7, Sem 2, 2022/23

Questions to be discussed: 2 and 3.

1. (Exercise 14.4 R&G) This question is a continuation of tutorial 4's question 2.

   Consider the join $R \bowtie_{R.a=S.b} S$, given the following information about the relations to be joined. The cost metric is the number of page I/Os unless otherwise noted, and the cost of writing out the result should be uniformly ignored.

   - 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.
   - 52 buffer pages are available.

   (a) What is the cost of joining R and S using a **sort-merge join** (use the optimized variant whenever it's applicable)? What is the minimum number of buffer pages required for this cost to remain unchanged?

   (b) What is the cost of joining R and S using the **Grace hash join**? What is the minimum number of buffer pages required for this cost to remain unchanged?

2. (Exercise 14.5 R&G) This question is a continuation of tutorial 4's question 3.

   Consider the join $R \bowtie_{R.a=S.b} S$, given the following information about the relations to be joined. The cost metric is the number of page I/Os unless otherwise noted, and the cost of writing out the result should be uniformly ignored.

   - 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.
   - 52 buffer pages are available.
   - Assume that any B^+-tree index on R has two levels of internal nodes, and any B^+-tree index on S has one level of internal nodes.

   (a) If only 15 buffer pages were available, what would be the cost of a **sort-merge join** (use the optimized variant whenever it's applicable)? What would be the cost of a **Grace hash join**?

   (b) If the size of S were increased to also be 10,000 tuples, but only 15 buffer pages were available, what would be the cost of a **sort-merge join** (use the optimized variant whenever it's applicable)? What would be the cost of a **Grace hash join**?

   (c) If the size of S were increased to also be 10,000 tuples, and 52 buffer pages were available, what would be the cost of a **sort-merge join** (use the optimized variant whenever it's applicable)? What would be the cost of a **Grace hash join**?

3. (Exercise 15.9, R&G) Consider the following scenario:

   - Emp (eid: integer, sal: integer, age: real, did: integer)
   - Dept (did: integer, projid: integer, budget: real, status: char(10))
   - Proj (projid: integer, code: integer, report: varchar)

   Assume that each Emp record is 20 bytes long, each Dept record is 40 bytes long, and each Proj record is 2000 bytes long on average. There are 20,000 tuples in Emp, 5000 tuples in Dept (note that did is not a key), and 1000 tuples in Proj. Each department, identified by did, has 10 projects on average. The file system supports 4000 byte pages, and 12 buffer pages are available. All following questions are based on this information. You can assume uniform distribution of values. State any additional assumptions. The cost metric to use is the number of page I/Os. Ignore the cost of writing out the final result.

   In this question, consider the following join algorithms: block nested loop join, indexed nested-loop join, sort merge join (use the optimized variant whenever applicable), and Grace hash join.

   Consider the following query: SELECT * FROM Emp E, Dept D WHERE E.did=D.did

   (a) Suppose that there is a format-1 hash index on Emp.did. List all the plans that are considered and identify the plan with the lowest estimated cost.

   (b) Assume that both relations are sorted on the join column. List all the plans that are considered and show the plan with the lowest estimated cost.

   (c) Suppose that there is a format-1 B+-tree index on Emp.did and Dept is sorted on did. List all the plans that are considered and identify the plan with the lowest estimated cost.