Homework 2 for Distributed Database Systems

(Issue: 2012 / 5 / 5 Due: 2012 / 5 / 16)

Question 1.

Consider a database consisting of the following two relations

Employees (eid:integer, did:integer, sal:real)

Department (did:integer, mgrid:integer, budget:integer)

The mgrid field of Department is the eid of the manager. Each of these relations contains 20-byte tuples, and the sal and budget fields both contain uniformly distributed values in the range 0 to 1,000,000. The Employees relation contains 100,000 pages, the Departments relation contains 5000 pages, and each processor has 100 buffer pages of 4000 bytes each. The cost of one page I/O is t_d and the cost of shipping one page is t_s. There are no indexes.

The database is stored in distributed DBMS with 10 sites. The Departments tuples are horizontally partitioned across the 10 sites by did, with the same number of tuples assigned to each site, and with no particular order to how tuples are assigned to sites. The Exployees typles are similarly partitioned, by sal ranges, with sal <= 100,000 assigned to the first site, 100,000 < sal <= 200,000 assigned to the second site, etc. In addition, the partition sal <= 100,000 is frequently accessed and infrequently updated, and it is therefore replicated at every site. No other Employees partition is replicated.

1) Give the cost for computing the natural join of Employees and Departments using the strategy of shipping all fragments of the smaller relation to every site containing tuples of the larger relation.

2) Describe the best plan and its cost for each of the following queries:
   a) Find the highest paid employee.
   b) Find the highest paid employee, with salary greater than 450,000 and less than 550,000.
   c) Find the highest paid manager for those departments stored at the query site.

Answer:

(1)

Let E be Employees and D be Departments.

E is divided into ten fragments: E1, E2, ..., E10

D is divided into ten fragments: D1, D2, ..., D10

Now the query is to E natural join D

= (E1 U E2 U ... U E10) natural join (D1 U D2 U ... U D10)

The strategy employed is to ship ALL fragments of the smaller relation to EVERY site containing tuples of the larger relation.

Obviously, the smaller relation is D, while the larger is E, so according to the strategy, we send all fragments of D to each site (because each site contains E's tuples). As a result, at each site, we have all fragments of D and thus we can reconstruct D.

The natural join equation can be transformed to:
((D1 U D2 ... U D10) natural join E1) U
((D1 U D2 ... U D10) natural join E2) U
...
((D1 U D2 ... U D10) natural join E10)

At each site i, we perform Ei natural join D and send the result to the query site. At the query site, we perform unions of them as the final result.

Cost calculation: (I/O cost + communication cost)
Assume there are only 10 sites.

Step 1: At each site i, send Di to all sites $j \neq i$.

Cost = Retrieve Di from disk to buffer and then send Di to 9 sites.
= 10 (500td + 9(500ts))

Step 2: At each site i, locally perform D natural join Ei.
Cost = cost for Hash join = 10(2(5000)td + 3(10000)td)

Step 3: At each site i, if it is not the query site, send its join result to the query site.
Let x be the length of the join attribute – the did attribute.
Cost = cost for sending the join result to query site
= 9 (ceiling ((10000 * (4000/20) * (20 + 20 – x)) / 4000)) ts

Total cost is the sum of the cost at each step.

Question 2.
Suppose that an Employees relation (the schema of Employees is described in Question 1) is stored in Hong Kong and the tuples with sal <=100,000 are replicated at Beijing.

Consider the following three options for lock management:
- All locks managed at a single site called Shenzheng;
- Primary copy: the copy of Employees at Hong Kong is chosen as the primary copy;
- Fully distributed

For each of the lock management options, explain locks are set at which site for the following queries, and state which site the page is read from:

1) A query submitted at Shanghai wants t read a page containing Employees tuples with sal<=50,000.

2) A query submitted at Hong Kong wants to read a page containing Employees tuples with sal <=50,000.

3) A query submitted at Beijing wants to read a page containing Employees tuples with sal <=50,000.

4) An update transaction, "Give all employees a 10% raise", is issued in the DDBS described in Question 1. Describe the sites visited and the locks obtained.
(Hint: ROWA policy is used for replications, and the conditions of the original partitioning of Employees must still be satisfied after update.)

Answer:
The best plan is to send the query to site 10, and perform a linear scan on E10 to search for the highest paid employees. (The highest paid employee(s) resides in E10, since sal is uniformly distributed values and therefore E10 is not empty.) The search result is then sent back to the query site.

Cost calculation:
Step 1: Send the query to the site 10 and linear scan of E10 for highest paid employees:
Cost = 10000td
(Note that we ignore the cost of sending the query to site 10 here)

Step 2: Send the search result back to query site (Note that this step is not necessary if the query site is site 10):
Since sal is uniformly distributed in the range 0 to 1,000,000 while the cardinality of Employees is 100000(4000/20) = 20,000,000, there are 20,000,000/1,000,000 = 20 highest paid employees, which fits in one page. Therefore,

the cost of this step is ts.

Total cost is 10000td + ts.

(2b)

The best plan is to send the query to site 6 (containing E6 (500,000 < sal <= 600,000), and perform a linear scan on E6 to search the employees having salary of $549,999.

Cost calculation:

Similar to 2a, total cost is 10000td + ts.

(2c)

Let the query site be site i. The best plan is to ask site j =10: send $\Pi_{mgrid}(Di)$ to site j and joins Ej on Ej.eid and search for the highest paid employee(s). Then send the search results back to the query site. At the query site, while search result is empty, ask site j-1.

Cost calculation:

Let x be the length of mgrid attribute.

When asking a site j, and j is not query site,

Step 1: send $\Pi_{mgrid}(Di)$ to site j

Cost = cost to retrieve Di from disk to buffer + cost to send $\Pi_{mgrid}(Di)$ to site j

= 500td + 500(x/20) ts

Step 2: locally hash join of $\Pi_{mgrid}(Di)$ and Ej and select the highest paid employee.

Cost = 2 (500 (x/20))td + 3 (10000)td

Step 3: send the search result back to query site (must fit into one page): Cost = ts

The search result must fit into one page (explained before), so cost is ts.

When asking a site j and j is the query site, just perform hash join of $\Pi_{mgrid}(Di)$ and Ej locally, and the cost is (500)td + 2(500)(x/20)td + 3(10000)td.

The total cost in this question depends on which site in which search result is found, but is the sum of the cost when asking sites.