CS561_HW3_Boya_Zhou

Question1:

	Partition1	Partition2	Partition3
Query1	1	2	3
Query2	4	5	6
Query3	7	8	9

There is **m** machine.

1. Query1, Partition1
    a. Parallel group by city: Server i partitions chunks Ri using a hash function (Customer.city mod P) on city (there is p city), get Ri1, Ri2….Ri,p-1.
    b. Server i send partition Rij to server j(assume j less than m)
    c. Server j computers AVG(age) on R0j, R1j, ……, Rp-1,j
    d. Send the result from server j to other server
2. Query1, Partition2:
    a. Since the data is evenly distributed to all nodes, still need to exchange data.
    b. The rest is as same as 1
3. Query1, Partition3:
    a. No need to do parallel group by
    b. Calculate the avg(age) just on the local machine
4. Query2, Partition1
    a. Parallel duplicate elimination: Duplicate elimination the age on local machine on Customer.age since is a range-based partition, no need to exchange the data
    b. Each sever deal its own data on its own server.
5. Query2, Partition 2
    a. Since all the records is evenly distributed on each server.
    b. Each server partition chunks Ri using a hash function (Customer.city mod P) on city, get Ri1, Ri2….Ri,p-1
    c. Server j send partition Rij to other server
    d. Right now, each city is only in one server. Server j eliminate duplicate (age duplication) on R0j, R1j, ……, Rp-1,j
6. Query2, Partition3
    a. Since it is based on city hash-based partition, each server perform duplication eliminate on its own server when considering age.
7. Query3, Partition 1
    a. Since the partition is range-based partition on age. Only the nodes contain age > 30 will work on data.
    b. Then select city=”Boston” or city=”New York”
8. Query3, Partition 2
    a. Since the data is well distributed on all servers, each server will perform the selection and projection on its own server.
9. Query3, Partition 3

a. Only the server contain city=Boston and city = New York do selection on age > 30

Question2:

1. First question
   - Send only S.y to R's location
   - Select R.x1 on R
   - Do the join based on Y columns in R's location after selection
   - Send the records of R that will join (without duplicates) to S's location
   - Perform the final join in S's location
2. Second question
   - Select on S.z1
   - Send all the records in S fit the selection to R's location
   - Perform the final join based on column y in R's location.
3. Third question
   - We assume here are two relations : R(<u>cusID</u>, age, saraly), S(<u>cusID</u>, phoneNum, gender), cusID is primary key, two relations have same number of primary key and the two relations are almost the same large, we assume the is C customers.
   - When using semi-join, the cost is S.cusID to R. The whole relation R to S.
   - When just send S to R, the cost is whole relation S to R.
   - Since the two relations are almost the same large, so the semi-join is less efficient here.
   - Beside this scenario, if column y is large, semi-join will be less effective, too.

Question 3

1. First question
   - The purpose of two-phase commit is coordinates all the processes that participate in a distributed atomic transaction on whether to commit or abort the transaction
   - Why it is widely is because The protocol achieves its goal even in many cases of temporary system failure.
2. Second question
   - Since the coordinator receive "ready T" from all sites, it will go to the Phase 2, after coordinator send "commit T" to all sites.
     i. After just the site recover, it will frozen in ready state. It will re-contact the coordinator, if can not, it will contact other participants, if there is any site in abort of commit state, this site will do the same thing. Abort T or Commit T.
3. Third question
   - Right now when coordinator recover, it is in prepare state, it will either
     i. Resend "prepare T" to other sites and wait for response.
     ii. Put "abort T" in its own log, send global "abort T" to all participants.
     iii. If after a long time, coordinator recover, maybe the participants already selected a new coordinator to replace it.