Solution of CIS 452/552 Assignment Two

1. *Explain the difference between data replication in a distributed system to the maintenance of a remote backup site? Compare the advantages and disadvantages of the data replication and data fragmentation approaches in a distributed system. When is it useful to have replication or fragmentation of data?*

Answer:

Data replication provides greater availability and also parallelism as multiple sites are able to cater to more transactions. But there is a reduction in data transfer as more systems have a local copy. Also, there is an update overhead as all replicas need to remain consistent.

Remote Backup systems carry out actions such as concurrency control and recovery at a single site. Also remote backup systems help avoid two phase commits and all the overhead resulting from it. Since transactions contact only one site, the overhead of running transactions at multiple sites are avoided. Therefore remote backup sites provide a lower cost approach.

Replication is useful when the data is required at multiple sites in a distributed system. Fragmentation is useful, as the data is made available only at sites where it is needed and will be useful, thus reduces redundancy. Both replication and fragmentation facilitate parallelism.

2. *What is the major reason to cause false deadlock in the distributed databases? Please provide an example of false deadlock which is different from the one shown in the lecture slides.*

Answer: The major reason for false deadlock in the distributed databases is the network delay or the timestamps are not synchronized, therefore transaction coordinator may receive the messages in wrong order. An earlier delete message may be received later than an adding message for the wait-for graphs, then a false deadlock may be formed.

3. *Consider the relations:*

employee (name, address, salary, plant number)

machine (machine, number, type, plant number)

Assume that the employee relation is fragmented horizontally by plant number, and that each fragment is stored locally at its corresponding plant site. Assume that the machine relation is stored in its entirety at the Armonk site. Describe a good strategy for processing each of the following queries.

a. Find all employees at the plant that contains machine number 1130.

b. Find all employees at plants that contain machines whose type is "milling machine."

c. Find all machines at the Almaden plant.

d. Find employee $\bowtie$ machine.

Answer:

a. i. Perform $\Pi_{\text{plant number}} (\sigma_{\text{machine number}=1130} (\text{machine}))$ at Armonk.

ii. Send the query Π_{name} (employee) to all site(s) which are in the result of the previous query.

iii. Those sites compute the answers.

iv. Union the answers at the destination site.

b. This strategy is the same as a), except the first step should be to perform

$\Pi_{\text{plant number}} (\sigma_{\text{type=“milling machine”}} (\text{machine}))$ at Armonk.

c. i. Perform $\sigma_{\text{plant number} = x}$ (machine) at Armonk, where x is the plant

number for Almaden.

ii. Send the answers to the destination site.

d. Strategy 1:

i. Group *machine* at Armonk by plant number.

ii. Send the groups to the sites with the corresponding plant number.

iii. Perform a local join between the local data and the received data.

iv. Union the results at the destination site.

Strategy 2:

Send the machine relation at Armonk, and all the fragments of the employee relation to the destination site. Then perform the join at the destination site.

There is parallelism in the join computation according to the first strategy but not in the second. Nevertheless, in a WAN the amount of data to be shipped is the main cost factor. We expect that each plant will have more than one machine, hence the result of the local join at each site will be a cross-product of the employee tuples and machines at that plant. This cross-product's size is greater than the size of the employee fragment at that site. As a result the second strategy will result in less data shipping, and will be more efficient.

4. a) Depends on the site which the query was entered. b) Depends on the site which the result was desired.

Answer: a) The cost of transferring the query itself is much less expensive than cost of transferring the data. The site at which the query was entered only needs to transfer the query to each plant_number site, which is not very expensive. Hence, our strategy does not depend on the site at which query was entered.

b) For the first query, we compute the plant_number at Armonk site and then compute the employee tuple at each site locally which are then transferred to the destination site. Our strategy is relatively independent of the site at which the results is desired.

For the second query, we again compute the plant_number at Armonk site and then compute employee tuple at each site locally (based on plant_number result) which are then transferred to destination site. Again, this strategy is relatively independent of the site at which the result is desired.

For the third query, the query is computed at Armonk site and then result is send to destination site, which means that the strategy is dependent of the site at which the result is desired.

For the fourth query, both possible strategies involve the data (or result) being migrated from each local site to the destination site regardless of the choice of destination. Hence, again this strategy is independent of the site at which the result is desired.

5. In a mediator (data integration) system, the mediated schema is

Customer (ID, name, statecode)

State(statecode, statename)

There are two data resources:

S1:

Customer (ID, full_name, region)

S2:

Customer (ID, first_name, last_name, region_code)

Region (region_code, region_name)

Assume "name" in the mediated schema means "full_name" in S1. "Region" in data resources means "state" in the mediated schema. For example, "state_code" or *"region_code" of "Oregon" is "OR." Can you do schema mediation (mapping) in Global-as-View or Local-as-View, or both? Write down your solution(s) in SQL views.*

Write a query "List all customer names in 'Oregon'" in SQL based on the mediated schema. Can the query be answered with your views?

Answer:

Global-as-View

Create View Customer as

Select S1.Customer.ID AS ID, S1.Customer.full_name AS name, S2.Region.region_code AS statecode

From S1.Customer, S2.Region

Where S1.Customer.region = S2.Region.region_name

Union

Select ID, CONCAT(first_name, " ", last_name) AS name, region_code AS statecode

from S2.Customer

Create View State as

Select region_code AS statecode, region_name AS state_name

from S2.Region

However, if join across tables in different databases is hard to implement. Another way is:

Create State View first,

Create View State as

Select region_code AS statecode, region_name AS state_name

from S2.Region

Then use it in the Customer view definition.

Create View Customer as

Select S1.Customer.ID AS ID, S1.Customer.full_name AS name, State.statecode AS statecode

From S1.Customer, State

Where S1.Customer.region = State.state_name

```
Union

Select ID, CONCAT(first_name, “ “, last_name) AS name, region_code AS statecode

from S2.Customer
```

Local-as-View

If we have a way extract first name and last name correctly from a full name, then we can define that

```
Create Source S1.Customer AS

Select Customer.ID, Customer.name, State.statename

From Customer, State

Where Customer.statecode = State.statecode


Create Source S2.Customer AS

Select ID,

ExtractFirstName (name) AS first_name,

ExtractLastName (name) AS last_name,

statecode AS region_code

from Customer


Create Source S2.Region AS

Select statecode AS region_name, statename AS region_name

From State
```

However, it is hard to use existing functions in SQL to create ExtractFirstName and ExtractLastName, considering there are different types full names, such as “Last, First”, “First Middle Last”, or the names with “de”, “von” etc.

Query:

```
Select name

From Customer, State

Where Customer.statecode = State.statecode AND State.statename= “Oregon”
```