Query Processing in Distributed DBMS

Last Updated : 06 Dec, 2023

Query processing in a distributed database management system requires the transmission of data between the computers in a network. A distribution strategy for a query is the ordering of data transmissions and local data processing in a database system. Generally, a query in Distributed DBMS requires data from multiple sites, and this need for data from different sites is called the transmission of data that causes communication costs. Query processing in DBMS is different from query processing in centralized DBMS due to the communication cost of data transfer over the network. The transmission cost is low when sites are connected through high-speed Networks and is quite significant in other networks.

The process used to retrieve data from a database is called query processing. Several processes are involved in query processing to retrieve data from the database. The actions to be taken are:

- Costs (Transfer of data) of Distributed Query processing
- Using Semi join in Distributed Query processing

Costs (Transfer of Data) of Distributed Query Processing

In Distributed Query processing, the data transfer cost of distributed query processing means the cost of transferring intermediate files to other sites for processing and therefore the cost of transferring the ultimate result files to the location where that result is required. Let's say that a user sends a query to site S1, which requires data from its own and also from another site S2. Now, there are three strategies to process this query which are given below:

1. We can transfer the data from S2 to S1 and then process the query
2. We can transfer the data from S1 to S2 and then process the query
3. We can transfer the data from S1 and S2 to S3 and then process the query. So the choice depends on various factors like the size of relations and the results, the communication cost between different sites, and at which the site result will be utilized.

Commonly, the data transfer cost is calculated in terms of the size of the messages. By using the below formula, we can calculate the data transfer cost:

Data transfer cost = C * Size

Where C refers to the cost per byte of data transferring and Size is the no. of bytes transmitted.

Example: Consider the following table EMPLOYEE and DEPARTMENT.

Site1: **EMPLOYEE**

EID	NAME	SALARY	DID

EID- 10 bytes
SALARY- 20 bytes
DID- 10 bytes
Name- 20 bytes
Total records- 1000
Record Size- 60 bytes

Site2: **DEPARTMENT**

DID	DNAME

DID- 10 bytes
DName- 20 bytes

Total records- 50
Record Size- 30 bytes

Example:

1. Find the name of employees and their department names. Also, find the amount of data transfer to execute this query when the query is submitted to Site 3.

Answer: Considering the query is submitted at site 3 and neither of the two relations is an EMPLOYEE and the DEPARTMENT not available at site 3. So, to execute this query, we have three strategies:

- Transfer both the tables that are EMPLOYEE and DEPARTMENT at SITE 3 then join the tables there. The total cost in this is **1000 * 60 + 50 * 30 = 60,000 + 1500 = 61500** bytes.
- Transfer the table EMPLOYEE to SITE 2, join the table at SITE 2 and then transfer the result at SITE 3. The total cost in this is **60 * 1000 + 60 * 1000 = 120000** bytes since we have to transfer 1000 tuples having NAME and DNAME from site 1,
- Transfer the table DEPARTMENT to SITE 1, join the table at SITE 2 join the table at site1 and then transfer the result at site3. The total cost is **30 * 50 + 60 * 1000 = 61500** bytes since we have to transfer 1000 tuples having NAME and DNAME from site 1 to site 3 which is 60 bytes each.

Now, If the Optimisation criteria are to reduce the amount of data transfer, we can choose either 1 or 3 strategies from the above.

Using Semi-Join in Distributed Query Processing

The semi-join operation is used in distributed query processing to reduce the number of tuples in a table before transmitting it to another site. This reduction in the number of tuples reduces the number and the total size of the transmission ultimately reducing the total cost of data transfer. Let's say that we have two tables R1, R2 on Site S1, and S2. Now, we will forward the joining column of one table say R1 to the site

where the other table say R2 is located. This column is joined with R2 at that site. The decision whether to reduce R1 or R2 can only be made after comparing the advantages of reducing R1 with that of reducing R2. Thus, semi-join is a well-organized solution to reduce the transfer of data in distributed query processing.

Example: Find the amount of data transferred to execute the same query given in the above example using a semi-join operation.

Answer: The following strategy can be used to execute the query.

- Select all (or Project) the attributes of the EMPLOYEE table at site 1 and then transfer them to site 3. For this, we will transfer NAME, DID(EMPLOYEE) and the size is 30 * 1000 = 30000 bytes.
- Transfer the table DEPARTMENT to site 3 and join the projected attributes of EMPLOYEE with this table. The size of the DEPARTMENT table is 30 * 50 = 1500

Applying the above scheme, the amount of data transferred to execute the query will be **30000 + 1500 = 31500** bytes.

Conclusion

In Conclusion, query processing in a distributed database management system (DBMS) is a complex procedure that tackles issues with transaction management, data dissemination, optimization, and fault tolerance. Distributed database systems' performance, scalability, and dependability depend on effective concurrency management, optimization, and query decomposition techniques.

Next Article

Fragmentation in Distributed DBMS