THE INTERNATIONAL JOURNAL OF SCIENCE & TECHNOLEDGE

Teradata: Future of Databases

Saurabh Merai
Department of Information Technology
Dwarkadas J. Sanghvi College of Engineering, Mumbai, India
Neha Mendjoge
Department of Information Technology
Dwarkadas J. Sanghvi College of Engineering, Mumbai, India
Kriti Srivastava
Department of Information Technology
Dwarkadas J. Sanghvi College of Engineering, Mumbai, India
Vinaya Sawant
Department of Information Technology
Dwarkadas J. Sanghvi College of Engineering, Mumbai, India

Abstract*:*
The data is constantly increasing and thus accommodating huge amount of data and faster processing of data has become a must.Oracle RDBMS can help to a certain extent but its sequential processing limits its processing power. In the business environment data has to be delivered to customers on time so when we deal with large scale data warehouses it's a must that processing power is high. The Teradata RDBMS was designed to eliminate the technical pitfalls of data warehousing with parallel processing as one of its outstanding features. It was designed to perform parallel processing and to store terabytes of data thereby making it efficient and one of the fastest data warehouses to work with. Also it has the unique feature of prioritizing a query if it is required to be completed on urgent basis. Teradata also has the ability to fetch data from other databases like Hadoop and Aster without any overhead and without impacting the Teradata system.

Keywords*: Sequential Processing; Parallel Processing; Prioritizing*

1. Introduction

The most interesting feature in Teradata is the concept of Parsing Engine and AMPs and how they communicate effectively with each other with the help of a BYNET.Teradata is the only data warehouse where data is loaded in parallel, retrieved in parallel and backed up in parallel.The parallel processing works by simultaneously fetching the rows of the table from the AMPs. The Parsing Engine which is the optimiser makes the optimum query plan and submits the plan across the AMPs for fetching the data. The query plan includes whether the data needs to be fetched based on Index or full-table scan, what join operations need to be applied and the approximate time in which query will complete and thus the Parsing Engine is said to be the brain of Teradata.Also Teradata is dynamic as it can fetch data from other database systems such as Hadoop easily with the help of its utilities. The only thing that needs to be done is firing a simple 'select' query on the Teradata system.

2.Architecture

Teradata provides customers a centrally located architecture.

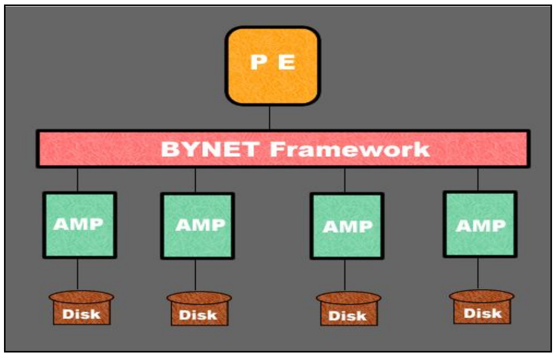


Figure 1: Architecture

Working :-
Whenever the user will fire a query it wil go through the following steps:-

- The Parsing Engine checks the syntax and then the security and comes up with a plan for the AMPs. The PE communicates with the AMPs across the BYNET. The data is then fetched from the AMPs and back to the user.
- AMPs temporarily store the answer sets in Spool Space and this space is released once the answer set is displayed

Components :-

2.1. Parsing Engine
The PE is the brain of Teradata and it comes up with most efficient query plan for ever query fired by the user.

- It checks whether the user is allowed access to the tables and then will decide whether the AMPs should use index or do a full-table scan.
- It will then call the dispatcher to pass this plan to the AMPs.

2.2. Access Module Processors
The AMPs are the workers of the Teradata system because the AMPs read and write the data to their assigned disks.
Every time a data is requested from the table the AMPs will fetch the rows simultaneously. The AMPs do the job of retrieving the answer set.

2.3.Bynet
The BYNET is the communication network between AMPs and PE.
Advantages:

- Teradata database is linearly scalable. - We can expand the database capacity by just adding more nodes to the existing database. Node is a combination of PE's and AMP's.
- 2It can handle multiple requests and users.

Disadvantages:

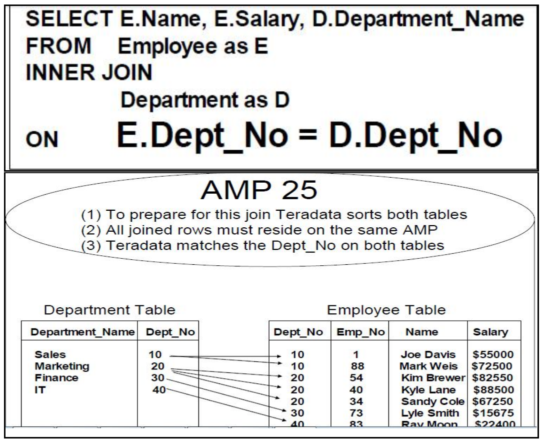
- The cost to add more P.E's and AMP's is very high.

3. Working of Queries
The rows of the table are spread across the AMPs. When the table needs to be read, each AMP only reads their part of the table. If the AMPs start reading simultaneously then parallel processing works. Also if the rows are evenly distributed amongst the AMPs parallel processing will work brilliantly since each will have almost equal number of rows. This is possible by using the Primary Index.

3.1. Primary Index
The Primary Index plays 2 roles:

- Fastest Way to Retrieve Data
- Incredibly important for Joins

Teradata distributes rows to the AMPs based on the primary index. For a join operation the 2 rows being joined must reside on the same AMP. If both tables have same primary index and if join operation is done on that, it will work like lightning as both rows will be on the same AMP as they will have the same rowhash. A rowhash is obtained by hashing the primary index value. If join operation is done on a column other than primary index then join will take much more time because the rows will need to be redistributed till they are on the same AMP. In the figure below Dept_no and Emp_no are primary indexes of Department and Employee table respectively. But for the join operation we need to rehash the Employee table by Dept_No so that the joining rows will be on same AMP. So the join operation will be a little slow. But if both the tables had same Primary Index join would be like lightning.



```
SELECT E.Name, E.Salary, D.Department_Name
FROM    Employee as E
INNER JOIN
            Department as D
ON        E.Dept_No = D.Dept_No
```

(1) To prepare for this join Teradata sorts both tables
(2) All joined rows must reside on the same AMP
(3) Teradata matches the Dept_No on both tables

Department Table

Department_Name	Dept_No
Sales	10
Marketing	20
Finance	30
IT	40

Employee Table

Dept_No	Emp_No	Name	Salary
10	1	Joe Davis	$55000
10	88	Mark Weis	$72500
20	54	Kim Brewer	$82550
20	40	Kyle Lane	$88500
20	34	Sandy Cole	$67250
30	73	Lyle Smith	$15675
40	83	Ray Moon	$22400

Figure 2: Join Operation

Also fetching the data based on primary index will be much faster as the PE will directly go to the particular AMP where the row resides. If its any other column PE will do a full-table scan until it finds the AMP which holds the row.

4. Teradata Active System Management

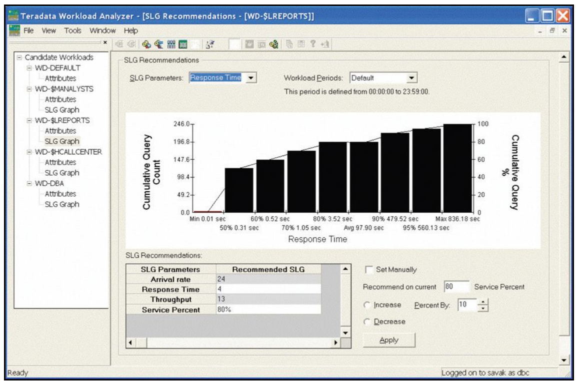
TASM is another important component of Teradata Technology. It is exclusive only to the Database developers, Database admins and other Workload schedulers.
TASM gives the ability to define and monitor Teradata performance. You can define the Workload settings, CPU consumption percentage, priority levels of every workload which varies according to the situations throughout the day.
Eg. : During Business hours, the priority is provided to the business users (9 am to 6 pm) for analysis purposes and during load hours the priority will be the load operation jobs which performs ETL operations and loads the data in the Teradata DWH.
This type of system is not seen in any other RDBMS. It gives users priority according to their type of work. Sometimes few queries need to be completed urgently. So for this a high priority is given to the query assigning it to a different workload . So on this workload the query will consume high CPU and complete in very less amount of time.
Example:-

- **Query:- select * from dp_bedw.fact where cust_id in (10,20,30) and mon_id=1373;**

Normally this query would take around 10 mins to complete. But once we change it to the appropriate workload the query will complete in less than 2 mins.This is done by:

- **Syntax:-Modify user 'username' as profile='profilename';**

Username has to be provided with what user has logged in and profile name indicates how much spool space is allocated to the user and on which workload his query will run. User is always allocated spool space and not permanent space thereby not letting him impact the system in any way. Permanent space is used to store databases and tables. Before providing any profile to the user it should be known whether his work is of high-priority or low-priority and based on that appropriate profile must be assigned to him.
This type of prioritization feature is not there in Oracle or other RDBMS.



SLG Parameters: Response Time
Workload Periods: Default
This period is defined from 00:00:00 to 23:59:00.

SLG Recommendations:

SLG Parameters	Recommended SLG
Arrival rate	24
Response Time	4
Throughput	13
Service Percent	80%

Recommend on current 80 Service Percent
Percent By: 10
Logged on to savak as dbc

Figure 3: Teradata Active System Management

Alongwith the above unique features for retrieving the data, prioritizing the queries Teradata also provides special utilities for loading and exporting the data. Normally if we have to insert a large no.of rows in a table we will have to run the 'Insert' statement a lot of times and this can be a very cumbersome and tedious task. But Teradata provides Fastload utility with the help of which all the data can be loaded in a single go.
Teradata FastLoad – This load utility is designed to move huge amount of data into an empty table. And because data conversion, movement, and loading are automatic and performed in parallel, FastLoad is more efficient than a standard application program written to load data to an empty database. Plus writing a ETL program is very complex task and needs to be rechanged once the conditions change and can be very tedious. Also Fastload script creates error tables and can capture the duplicate rows if any in that error table during loading.
Teradata FastExport – It is the reverse of the FastLoad utility. It quickly exports data sets from Teradata tables or views to a user for processing, generating reports, or for loading data into a database.

5. Conclusion

From this analysis we can say that Teradata is a must when it comes to handling large amounts of data and even though its expensive its very much needed because of its high processing power and its expandability. It provides with quite other interesting features like prioritization and deprioritization of queries, assigning appropriate session to the users, load utilities with ease to load the data into the tables and export from the same as compared to complex ETL programming and makes it a promise for the future when processing massive amount of data will be a need for the business.

6. Acknowledgement

We would like to thank respected Principal, Dr. Hari Vasudevan, D. J. Sanghvi College of Engineering for giving us facilities and guidance. We would also like to thank Shri Vile Parle Kelavani Mandal for encouraging us in such co- curricular activities.

7. References

1. Douglas P. Brown, Jeetendra Chaware, Manjula Koppuravuri(2011).Index selection in a database system.
2. Shivnath Babu,Goetz Graefe ,Harumi Anne Kuno(2012).Database Workload Management.
3. 3. Gang Luo, Jeffrey F. Naughton, Curt J. Ellmann, Michael W. Watzke(2008). Rescheduling table scan transactions.
4. Datta, A. , VanderMeer, D. , Ramamritham, K. (2002). Parallel Star Join+DataIndexes: efficient query processing in data warehouses and OLAP.Knowledge and Data Engineering,14,1299-1316.doi: 10.1109/TKDE.2002.1047769