Data Warehousing – Tuning and Testing

Data Warehousing - Tuning

- A data warehouse keeps evolving and it is unpredictable what query the user is going to post in the future. Therefore it becomes more difficult to tune a data warehouse system.
- Tuning a data warehouse is a difficult procedure due to following reasons −
  - Data warehouse is dynamic; it never remains constant.
  - It is very difficult to predict what query the user is going to post in the future.
  - Business requirements change with time.
  - Users and their profiles keep changing.
  - The user can switch from one group to another.
  - The data load on the warehouse also changes with time.

Performance Assessment

- Here is a list of objective measures of performance −
  - Average query response time
  - Scan rates
  - Time used per day query
  - Memory usage per process
  - I/O throughput rates
- Following are the points to remember.
  - It is necessary to specify the measures in service level agreement (SLA).
  - It is of no use trying to tune response time, if they are already better than those required.
  - It is essential to have realistic expectations while making performance assessment.
  - It is also essential that the users have feasible expectations.
  - To hide the complexity of the system from the user, aggregations and views should be used.
  - It is also possible that the user can write a query you had not tuned for.

Data Load Tuning

There are various approaches of tuning data load that are discussed below

- The very common approach is to insert data using the **SQL Layer**
- The second approach is to bypass all these checks and constraints and place the data directly into the preformatted blocks.
- The third approach is that while loading the data into the table that already contains the table, we can maintain indexes.
- The fourth approach says that to load the data in tables that already contain data, **drop the indexes & recreate them** when the data load is complete.

Integrity Checks

- Integrity checking highly affects the performance of the load. Following are the points to remember −
- Integrity checks need to be limited because they require heavy processing power.
- Integrity checks should be applied on the source system to avoid performance degrade of data load.

Tuning Queries

- We have two kinds of queries in data warehouse −
  - Fixed queries
  - Ad hoc queries

- Fixed queries are well defined. Following are the examples of fixed queries −
  - regular reports
  - Canned queries
  - Common aggregations
- Tuning the fixed queries in a data warehouse is same as in a relational database system.

Ad hoc Queries

- To understand ad hoc queries, it is important to know the ad hoc users of the data warehouse.
- For each user or group of users, you need to know the following −
  - The number of users in the group
  - Whether they use ad hoc queries at regular intervals of time
  - Whether they use ad hoc queries frequently
  - Whether they use ad hoc queries occasionally at unknown intervals.
  - The maximum size of query they tend to run
  - The average size of query they tend to run
  - Whether they require drill-down access to the base data
  - The elapsed login time per day
  - The peak time of daily usage
  - The number of queries they run per peak hour

Points to Note

- It is important to track the user's profiles and identify the queries that are run on a regular basis.
- It is also important that the tuning performed does not affect the performance.
- Identify similar and ad hoc queries that are frequently run.
- If these queries are identified, then the database will change and new indexes can be added for those queries.
- If these queries are identified, then new aggregations can be created specifically for those queries that would result in their efficient execution.

Data Warehousing - Testing

- Testing is very important for data warehouse systems to make them work correctly and efficiently.
- There are three basic levels of testing performed on a data warehouse −
  - Unit testing
  - Integration testing
  - System testing

Unit Testing

- In unit testing, each component is separately tested.
- Each module, i.e., procedure, program, SQL Script, Unix shell is tested.
- This test is performed by the developer.

Integration Testing

- In integration testing, the various modules of the application are brought together and then tested against the number of inputs.
- It is performed to test whether the various components do well after integration.

System Testing

- In system testing, the whole data warehouse application is tested together.
- The purpose of system testing is to check whether the entire system works correctly together or not.
- System testing is performed by the testing team.
- Since the size of the whole data warehouse is very large, it is usually possible to perform minimal system testing before the test plan can be enacted.

System Testing

- In system testing, the whole data warehouse application is tested together.
- The purpose of system testing is to check whether the entire system works correctly together or not.
- System testing is performed by the testing team.
- Since the size of the whole data warehouse is very large, it is usually possible to perform minimal system testing before the test plan can be enacted.

Testing Backup Recovery

- Testing the backup recovery strategy is extremely important. Here is the list of scenarios for which this testing is needed –
    - Media failure
    - Loss or damage of table space or data file
    - Loss or damage of redo log file
    - Loss or damage of control file
    - Instance failure
    - Loss or damage of archive file
    - Loss or damage of table
    - Failure during data failure

Testing Operational Environment

- **Security** − A separate security document is required for security testing. This document contains a list of disallowed operations and devising tests for each.
- **Scheduler** − Scheduling software is required to control the daily operations of a data warehouse. It needs to be tested during system testing. The scheduling software requires an interface with the data warehouse, which will need the scheduler to control overnight processing and the management of aggregations.
- **Disk Configuration.** − Disk configuration also needs to be tested to identify I/O bottlenecks. The test should be performed with multiple times with different settings.
- **Management Tools.** − It is required to test all the management tools during system testing. Here is the list of tools that need to be tested.
  - Event manager
  - System manager
  - Database manager
  - Configuration manager
  - Backup recovery manager

Testing the Database

- **Testing the database manager and monitoring tools** − To test the database manager and the monitoring tools, they should be used in the creation, running, and management of test database.
- **Testing database features** − Here is the list of features that we have to test −
  - Querying in parallel
  - Create index in parallel
  - Data load in parallel
- **Testing database performance** − Query execution plays a very important role in data warehouse performance measures. There are sets of fixed queries that need to be run regularly and they should be tested.

Testing the Application

- All the managers should be integrated correctly and work in order to ensure that the end-to-end load, index, aggregate and queries work as per the expectations.
    - Each function of each manager should work correctly
    - It is also necessary to test the application over a period of time.
    - Week end and month-end tasks should also be tested.
    - Overnight processing
    - Query performance

Logistic of the Test

- The aim of system test is to test all of the following areas −
  - Scheduling software
  - Day-to-day operational procedures
  - Backup recovery strategy
  - Management and scheduling tools
  - Overnight processing
  - Query performance