Performance Tuning Of Oracle Forms

For

Media Group
CASE STUDY

Performance Tuning Of Oracle Forms

CLIENT COMPANY PROFILE:

- The Client is leading global media investment management operation.
- Industry leading levels of investment in research and development.
- 400 offices in 81 countries.
- Part of WPP Group.
- Total worldwide market share: 32.7% (nearest competitor is Publicis with 23.1%)

TECHNOLOGIES USED:

- Language: SQL, PLSQL
- Transport: HTTP
- Development Tool: Oracle Forms Builder
- App Server: Oracle Weblogic Server 10.3.5
- Database: Oracle 11g
- Platform: Windows Server 2008

OVERVIEW:

The company is WPP’s consolidated media investment management operation, serving as the parent company to many other agencies. It is the global number one media investment management group (RECMA 2010).

The Company is engaged in the business of creating, financing and distributing intellectual property to create advantages for its clients and generate new sources of revenue for its agencies.

This case study describes how Simple Logic has tuned Oracle Forms to achieve efficient performance.

BUSINESS NEEDS AND CHALLENGES:

1. Reduce the load and run times for the form.
2. Optimize database access from the form.
3. Reduce memory requirements on the client.
4. Minimize network traffic.
5. Maximize the efficiency of packets sent over the network.
6. Render application displays efficiently on the client.

SOLUTION:

PERFORMANCE TUNING OF ORACLE FORMS:

STEP 1: Simple Logic Team first started with system-level SQL tuning. The Team performed optimization of the following things:

- **Optimization of the server kernel:**
  The Simple Logic Team tuned the disk and network I/O subsystem (RAID, DASD bandwidth, network) to optimize the I/O time, network packet size and dispatching frequency.

- **Adjusted the optimizer statistics:**
  You must always collect and store optimizer statistics to allow the optimizer to learn more about the distribution of your data to take more intelligent execution plans.

- **Made changes in the optimizer parameters:**
  Like optimizer_mode, optimizer_index_caching, optimizer_index_cost_adj.

- **Optimized the instance:**

© SIMPLE LOGIC IT PVT.LTD, All rights reserved.
Changed the `db_block_size`, `db_cache_size`, and OS parameters (`db_file_multiblock_read_count`, `cpu_count`) to influence SQL performance.

**Step 2:** Simple Logic Team after performing system-level SQL tuning started database level SQL query tuning using the following steps:

- **Identified high-impact SQL:**
  The DBA Team identified high impact sql queries considering following points:
  - Queries processing a large number of rows.
  - Queries having high buffer gets.
  - Queries having disk reads
  - The amount of memory allocated to a SQL statement.
  - SQL statements utilizing more CPU.
  - Number of sorts performed by a Query.
  - Most frequently executed query.

- **Determined the execution plan for the SQL statements:**
  Different execution plans were studied to understand the performance of the queries.

**Step 3:** Based on all the above findings, the Simple Logic DBA Team tuned the SQL statements using the following steps:

- Rewrite the complex sub queries with temporary tables.
- Used minus instead of EXISTS sub queries.
- The Team used analytic functions so that it can do multiple aggregations with a single pass through the tables, making them very fast for reporting SQL.
- Replaced NOT EXISTS and NOT EXISTS sub queries with outer joins.
- Created Index on NULL values for SQL that frequently tests for NULL.
- Remove the LIKE predicate wherever not required.
- Checked if there are any mixed data types.
- The Team used ‘decode’ and case functions for performing complex aggregations.
- The team avoided full table scans using indexes.

**RESULTS:**

The Client is happy with the performance gains after tuning the Forms. Users are happy as they don’t have to wait for very long time to get their forms processed. Overall performance of the application is working very smoothly.