Techniques for Analyzing Historical DB2 Application Performance Data

Group 5:
Lu Liu, Yumeng Wang
February 19th, 2013

[4] Oracle SQL Tuning – Tune individual SQL statements by Burleson Consulting
http://www.dba-oracle.com/art_sql_tune.htm
**Goal**

- DB2 Application Performance Tuning

**Strategy**

- SQL Tuning
- Resource Tuning
- Application Tuning
- Object Tuning
- Systems Tuning

from http://www.dba-oracle.com/art_sql_tune.htm
SQL Tuning: Goals

• Remove unnecessary large-table full-table scans
• Cache small-table full-table scans
• Verify optimal index usage
• Materialize aggregations and summaries for static tables

To Fetch the rows with minimum number of database "touches"
A Strategic Plan for SQL Tuning

1. Identify High-Impact SQL
   - Rank SQL Statements by #executions(Rows processed, Disk reads, etc)

2. Determine Execution Plan for SQL
   - Creates Plan-Table, use Explain Facility

3. Tune the SQL Statement
   - Adding SQL “hints”
   - Re-write SQL with Global Temporary Tables
   - Rewriting SQL in PL/SQL
Tools

- SMF(Service Management Facilities)/RMF(Resource Management Facilities) Reporting

Reports

- Accounting *Short* Report
- Historical Access Path
- Top “n” Report
- Explain Facility Report
- Health Check Information
- Accounting *Long* Report

Identify high-impact stmt.

Determine Execution Plan
Top “n” Reports

• Limited number of items on selected fields
• Usually categorized based on application type
• Gives an idea of the most expensive queries
• Example fields:
  ✓ Elapsed Time(Application; in DB2)
  ✓ CPU Time(Application; in DB2)
  ✓ Class 1 Wait Time(in-Application)
  ✓ Class 2 Wait Time(in-DB2)
  ✓ Class 3 Suspensions
  ✓ # DML Statements Executed
  ✓ # GetPages
**EXPLAIN in SQL**[2]

- **Function:** show the execution plan of a statement
- **Synopsis:** `EXPLAIN [ ANALYZE ] [ VERBOSE ]` query
- **Description:**
  - Displays the execution plan
    - How referenced tables are scanned (linear, index, etc.)
    - What join algorithms to use
    - Estimated query execution cost
  - `ANALYZE` option causes the query to be actually executed
  - `VERBOSE` option shows the internal plan tree

EXPLAIN Plan in SQL

Example[3]

```sql
SQL> EXPLAIN PLAN FOR
  2   SELECT *
  3   FROM   emp e, dept d
  4   WHERE  e.deptno = d.deptno
  5   AND    e.ename = 'SMITH';

Explained.
```

### EXPLAIN Plan in SQL

#### Example

```sql
SQL> @$ORACLE_HOME/rdbms/admin/utlxpls.sql

Plan Table

<table>
<thead>
<tr>
<th>Operation</th>
<th>Name</th>
<th>Rows</th>
<th>Bytes</th>
<th>Cost</th>
<th>Pstart</th>
<th>Pstop</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT STATEMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NESTED LOOPS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TABLE ACCESS FULL</td>
<td>EMP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TABLE ACCESS BY INDEX RO</td>
<td>DEPT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INDEX UNIQUE SCAN</td>
<td>PK_DEPT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8 rows selected.
```

Example of Execution Plan
1. Use Explain to output access path information[4]

```
EXPLAIN PLAN SET STATEMENT_ID = 'test1' FOR
   SET STATEMENT_ID = 'RUN1'
   INTO plan_table
   FOR
   SELECT  'T'||plansnet.terr_code, 'P'||detplan.pac1 || detplan.pac2 || detplan.pac3, 'P1', sum(plansnet.ytd_d_ly_tm),
             sum(plansnet.ytd_d_ty_tm),
             sum(plansnet.jan_d_ly),
             sum(plansnet.jan_d_ty),
   FROM plansnet, detplan
   WHERE
   plansnet.mgc = detplan.mktgpm
   AND
   detplan.pac1 in ('N33','192','195','201','BAI',
                    'P51','Q27','180','181','183','184','186','188',
                    '198','204','207','209','211')
   GROUP BY 'T'||plansnet.terr_code, 'P'||detplan.pac1 || detplan.pac2 || detplan.pac3;
```

Example of Execution Plan

2. Query on the plan table to get information [5]

```sql
plan.sql - displays contents of the explain plan table
SET PAGES 9999;
SELECT  lpad(' ',2^(level-1))||operation operation,
    options,
    object_name,
    position
FROM plan_table
START WITH id=0
AND
    statement_id = 'RUN1'
CONNECT BY prior id = parent_id
AND
    statement_id = 'RUN1';
```

Example of Execution Plan

2. Query on the plan table to get information [5]

```sql
SQL> @list_explain_plan

OPERATION
---------------------------------------------------------------
OPTIONS OBJECT_NAME POSITION
---------------------------------------------------------------
SELECT STATEMENT
SORT
GROUP BY
   CONCATENATION  1
   NESTED LOOPS   1
TABLE ACCESS FULL PLANSNET 1
TABLE ACCESS BY ROWID DETPLAN 2
   INDEX RANGE SCAN DETPLAN_INDEX5 1
NESTED LOOPS
```

Reference


[4] Oracle SQL Tuning – Tune individual SQL statements by Burleson Consulting
http://www.dba-oracle.com/art_sql_tune.htm