Oracle 12c Optimizer

Lisa Garczynski

November 13, 2015

Agenda
- What is new?
- Behavior Observations based on testing
- Recommendations

PAYCHEX

Overview

- **Oracle 12c Optimizer Changes Whitepaper**
  - http://www.oracle.com/technetwork/database/bi-datawarehousing/twp-optimizer-with-oracledb-12c-1963236.pdf

- **Environment**
  - Testing Oracle 11.2.0.4 → 12.1.0.2 upgrade using database replay
  - Highly normalized schema – Over 1000 tables, 4900 indexes
  - Most indexes are single column, to support foreign keys (prevent locking)

Adaptive Plans

Adaptive Query Optimization

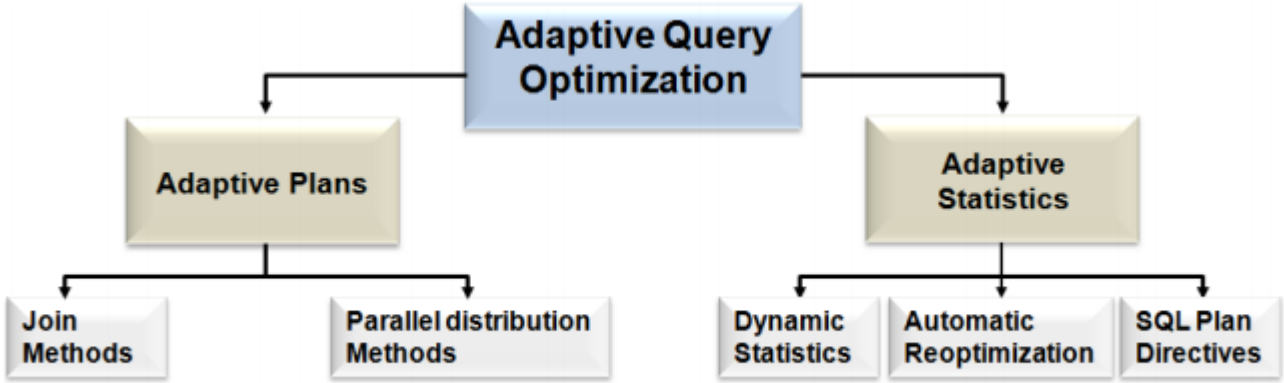


Figure 1. The components that make up the new Adaptive Query Optimization functionality

Runtime Adjustments to Query Plans

- Defer final plan decision for a statement until runtime
- Default plan has statistic collectors so if cardinality estimates differ greatly from actual number of rows, that plan or a portion of it can be adapted.

Adaptive Join Methods

- Pre-determines multiple sub-plans – Hash Join vs. Nested Loop via index
- Based on number actual number of rows, Oracle can choose

PAYCHEX

Our Experience

- **Adaptive Plans Experience**
  - Complex SQL Statements, many plans
  - Run the same statement, same statistics, 4 or 5 plans before “Final” plan is chosen
  - Difficult to tune/predict behavior
- **Recommendation**
  - We chose to turn it off to make it easier to get predicable results between executions

Parameter	Default	Change to
optimizer_adaptive_features	TRUE	FALSE
optimizer_adaptive_reporting_only	FALSE	TRUE

Scalar Subquery Unnesting

A scalar subquery is a subquery that appears in the SELECT clause of a SQL statement. Scalar subqueries are not unnested, so a correlated scalar subquery (one that references a column outside the subquery) needs to be evaluated for each row produced by the outer query.

Consider the following query,

```
SELECT  wpck_id,
      (SELECT SUM (wpccd_tax_empr_adj_amt)
         FROM wrkr_perd_chk_calc_detl
        WHERE wpccd_chkw_id = this_chkw.chkw_id)
         empr_adj_amt
  FROM wrkr_perd_chk
      JOIN wrkr
        ON wrkr_id = :b1
      JOIN chk_whld this_chkw
        ON chkw_wpck_id = wpck_id
      LEFT JOIN wptd_detl
        ON (wptdd_wrkr_id = wpck_wrkr_id)
      LEFT JOIN wrkr_perd_chk_comp_per_tax
        ON wpccpt_chkw_id = chkw_id
      LEFT JOIN wrkr_perd_chk_calc_detl
        ON wpccd_chkw_id = chkw_id
 WHERE wptdd_wrkr_id = :b1
```

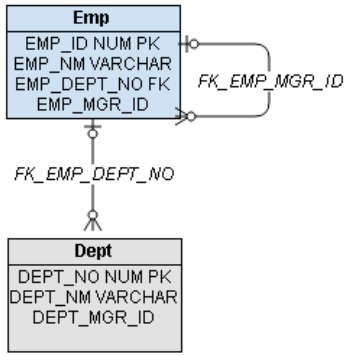
Our Experience

- **Majority of our queries returning 0 to 1 row**
  - Join order changed, most expensive tables doing multiple full scans
  - Queries that were running 38 ms ran 40 minutes
- **Recommendation**
  - This is likely a bug. We're working with Oracle to determine a patch.
  - Test in your environment
  - Workaround,
    - alter session set "_optimizer_unnest_scalar_sq"=FALSE;

Foreign Key Constraint Overview

- Foreign key constraint enforces business rules and describes relationships to optimizer

Entity Relationship Diagram (ERD)



- A manager is also an employee, so the ***fk_emp_mgr_id*** defines that recursive relationship in the emp table data.
- By default, Oracle enforces the constraints immediately during an insert. If you were to insert employee data for a department that didn't exist, the insert would fail the ***fk_emp_dept_no*** constraint. If you insert an employee before the manager, it would fail the ***fk_emp_mgr_id*** constraint.

PAYCHEX

Constraint Status

- **Enabled/Disabled**
  - Is the constraint on/off
- **Validated/Not Validated**
  - Has data been checked by Oracle
- **Not Deferrable**
  - Default, constraint checked on insert
- **Deferrable**
  - Constraint is checked on commit
- **Rely**
  - Yes – Used in conjunction with QUERY_REWRITE_INTEGRITY to tell optimizer eventhough I haven't validated my constraints, data is good.
  - Null – Default

Summary Statuses

- **Typical OLTP**
  - Enabled, Validated, Not Deferred
  - QUERY_REWRITE_INTEGRITY = ENFORCED (default)
- **Typical Data Warehouse**
  - Disabled, Not Validated, Not Deferred, Rely
  - QUERY_REWRITE_INTEGRITY=TRUSTED (trust data is good)
- **Paychex OLTP for payroll application**
  - Constraints are DEFERRED
  - Bulk transfer of data between systems, done by client with a single commit. The recursive relationships in the data make coding to ensure order more difficult.

Why is this important?

- **Since the optimizer knows the relationship, it can perform join elimination (department table not needed)**

```
SELECT emp_id, emp_nm
FROM   emp, dept
WHERE  dept_no = emp_dept_no
AND    emp_no = 10;
```

  - Department table really isn't needed, so Oracle eliminates it
- **Optimizer Behavior Change in 12c with DEFERRED constraints**
  - There was a bug in 11g (12628042) when the constraints are deferred, that join elimination would cause wrong results. Oracle fixed the wrong results bug in 12c, but this meant join elimination would no longer work unless we changed the status of the constraints to be not deferred.
  - Large report queries couldn't be tuned because additional tables needed to be joined. Modifying the code required 1600 hours of effort
  - 12c Patch – 21424812 plus session parameter
    - ALTER SESSION SET ALL CONSTRAINTS IMMEDIATE will permit join elimination!!

Under Investigation

- **I/O Costing appears to be different**
  - Setting db_multiblock_read_count from 16 to 128 changed query to Hash Join to Nested Loop/Index (desired plan)
- **Converting to ANSI sql prior to optimization**
  - 10053 trace files are HUGE