CSCI317 Database Performance Tuning

Simple Performance Measurement Tools

Dr Janusz R. Getta

School of Computing and Information Technology -
University of Wollongong

Simple Performance Measurement Tools

Outline

Simple measurement of response time

Using `TIMING` option of SQLcl

Using `V$` dynamic performance views

Simple measurement of response time

Find the total number of different values in `L_TAX` column

Declaration of SQLcl variables

```
VARIABLE START_TIME VARCHAR2(20);
VARIABLE STOP_TIME VARCHAR2(20);
VARIABLE TOTAL_TIME NUMBER;
```

Anonymous PL/SQL block

```
BEGIN
 SELECT TO_CHAR(SYSTIMESTAMP, 'DD-MON-YYYY:HH24:MI:SS:FF')
 INTO :START_TIME
 FROM DUAL;
END;
/
```

Processing of SQL statement

```
SELECT COUNT(DISTINCT L_TAX)
FROM LINEITEM;
```

Anonymous PL/SQL block

```
BEGIN
 SELECT TO_CHAR(SYSTIMESTAMP, 'DD-MON-YYYY:HH24:MI:SS:FF')
 INTO :STOP_TIME
 FROM DUAL;
END;
/
```

Simple measurement of response time

Find the total number of different values in L_TAX column

```sql
BEGIN
 SELECT EXTRACT(SECOND FROM
                 (TO_TIMESTAMP(:STOP_TIME,'DD-MON-YYYY:HH24:MI:SS:FF') -
                  TO_TIMESTAMP(:START_TIME,'DD-MON-YYYY:HH24:MI:SS:FF')))
 INTO :TOTAL_TIME
 FROM DUAL;
END;
/
```

```sql
PRINT :START_TIME;
PRINT :STOP_TIME;
PRINT :TOTAL_TIME;
```

Simple Performance Measurement Tools

Outline

Simple measurement of response time

Using `TIMING` option of SQLcl

Using `V$` dynamic performance views

Using TIMING option of SQLcl

Find the total number of different values in L_TAX column

```
SET TIMING ON
```
SQL

```
SELECT COUNT(DISTINCT L_TAX)
FROM LINEITEM;
```
SQL

```
COUNT(DISTINCTL_TAX)
--------------------
9
Elapsed: 00:00:02.69
```
SQL

```
SET TIMING OFF
```
SQL

Simple Performance Measurement Tools

Outline

Simple measurement of response time

Using `TIMING` option of SQLcl

Using `V$` dynamic performance views

Using `v$` dynamic performance views

`V$` data dictionary views are called as "dynamic performance views" and "global dynamic performance views"

Dynamic performance views contain information about the most current database activities

For example a view `V$INSTANCE` includes instance name, host name, start-up time, etc.

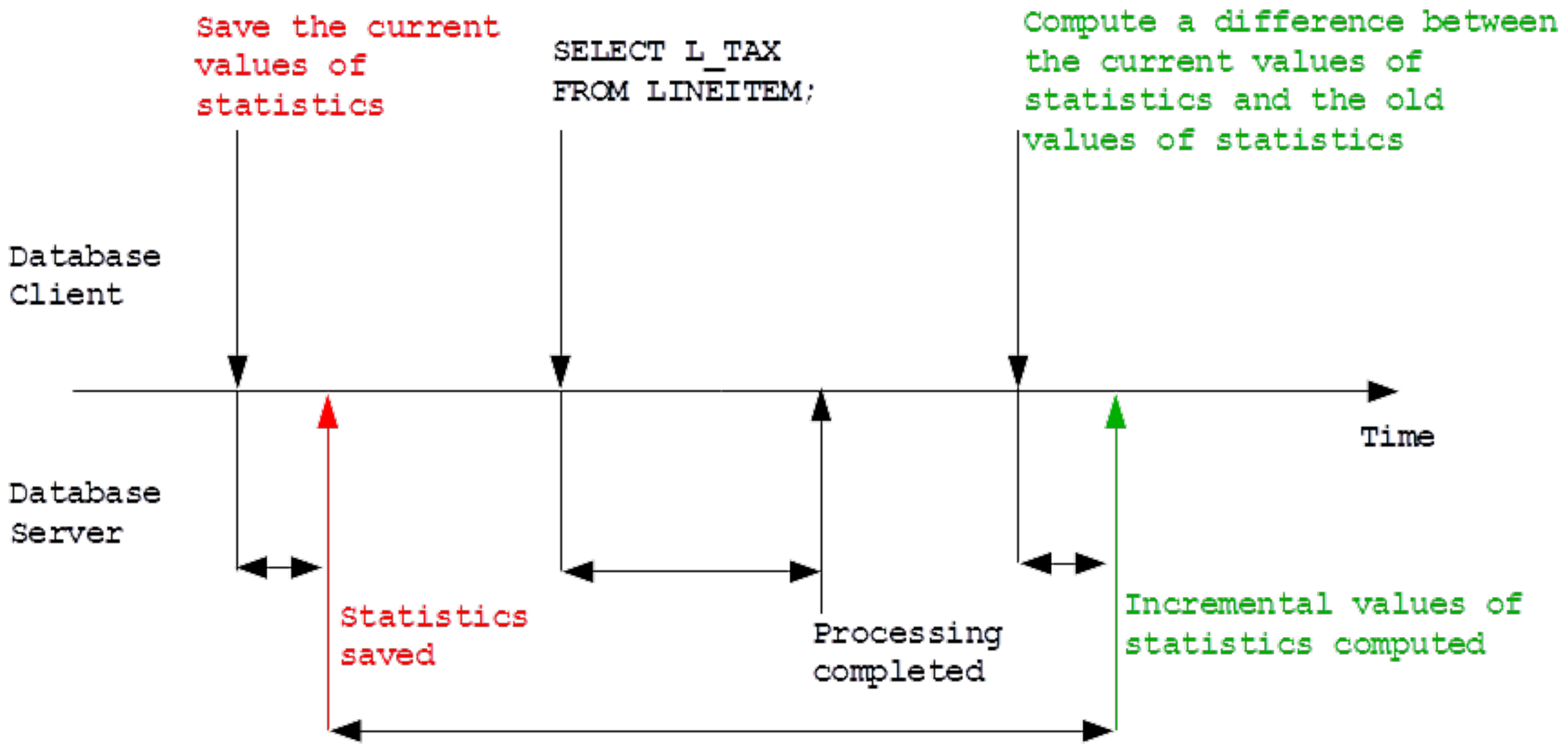
A view `V$DATABASE` includes database name, date when created, current operating system, etc.

A view `V$SESSION` includes active connections, login times, current state of transaction, etc.

A view `V$SEGEMENT_STATITICS` includes total number of read/write operations performed on all segments, etc.

Using v$ dynamic performance views

Find the total number of read block operations needed to compute a query that finds the values in L_TAX column


Save the current values of statistics
SELECT L_TAX FROM LINEITEM;
Compute a difference between the current values of statistics and the old values of statistics
Database Client
Time
Database Server
Statistics saved
Processing completed
Incremental values of statistics computed

Using v$ dynamic performance views

Find the total number of read block operations needed to compute a query that finds the values in `L_TAX` column

Connecting as a user SYSTEM

```
CONNECT SYSTEM
```

Creating a relational table for statistics

```
CREATE TABLE OPSTAT_TABLE(
OWNER            VARCHAR(30) NOT NULL,
OBJECT_NAME      VARCHAR(30) NOT NULL,
STATISTIC_NAME   VARCHAR(64) NOT NULL,
VALUE            NUMBER NOT NULL,
TS               TIMESTAMP NOT NULL,
  CONSTRAINT OPSTAT_TABLE_PKEY PRIMARY KEY(TS,OWNER,OBJECT_NAME,STATISTIC_NAME) );
```

Saving the current values of statistics

```
INSERT INTO OPSTAT_TABLE (
        SELECT OWNER,
               OBJECT_NAME,
               STATISTIC_NAME,
               VALUE,
               SYSTIMESTAMP
        FROM   V$SEGMENT_STATISTICS
        WHERE  OWNER = 'CSCI317' AND
               STATISTIC_NAME IN ('physical reads', 'logical reads') );
COMMIT;
```

Using v$ dynamic performance views

Find the total number of read block operations needed to compute a query that finds the values in `L_TAX` column

Connecting as a user CSCI317

```
CONNECT CSCI317
```

Processing SELECT statement

```
SELECT L_TAX
FROM LINEITEM;
```

Connecting as a user SYSTEM

```
CONNECT SYSTEM
```

Finding the differences between the old and current statistics

```
SELECT V$SEGMENT_STATISTICS.OWNER,
       V$SEGMENT_STATISTICS.OBJECT_NAME,
       V$SEGMENT_STATISTICS.STATISTIC_NAME,
       V$SEGMENT_STATISTICS.VALUE - NVL(OPSTAT_TABLE.VALUE,0) INCR
FROM   V$SEGMENT_STATISTICS LEFT OUTER JOIN OPSTAT_TABLE
                            ON V$SEGMENT_STATISTICS.OWNER = OPSTAT_TABLE.OWNER AND
                               V$SEGMENT_STATISTICS.OBJECT_NAME = OPSTAT_TABLE.OBJECT_NAME AND
                               V$SEGMENT_STATISTICS.STATISTIC_NAME = OPSTAT_TABLE.STATISTIC_NAME
WHERE  V$SEGMENT_STATISTICS.OWNER = 'CSCI317' AND
      ( V$SEGMENT_STATISTICS.VALUE - NVL(OPSTAT_TABLE.VALUE,0) ) <> 0 AND
      V$SEGMENT_STATISTICS.OBJECT_NAME <> 'PLAN_TABLE' AND
      V$SEGMENT_STATISTICS.STATISTIC_NAME IN ('physical reads', 'logical reads')
ORDER BY V$SEGMENT_STATISTICS.OBJECT_NAME, V$SEGMENT_STATISTICS.STATISTIC_NAME;
```

References

Cookbook, How to use TIMING, AUTOTRACE, and dynamic performance (V$) views

G. Harrison Oracle Performance Survival Guide, Prentice Hall, 2010

S. R. Alapati, Oracle 12c Performance Tuning Recipes, A Problem-Solution Approach, Apress, 2014