Impact of Data Statistics on the Execution Plan of the Oracle Database

Aiwu Li
Department of Computer Science, Guangdong Vocational College of Post and Telecom, Guangzhou, China

*Corresponding author e-mail: law_mail@126.com

Abstract. The data in the database is constantly changing. In order to make the statistical information truly reflect the current situation of the database, it is necessary to update the statistical information when needed. This paper uses column statistics as an example to analyze the impact of statistics on the execution plan of a single-table query.

1. Introduction

1.1. Contents of statistics
Data statistics are the data volume and data distribution details of the database and each object in the database, which are used to select the optimal execution plan for each SQL statement. The data in the database is constantly changing.

According to the object type, statistics are mainly divided into: table statistics, index statistics, column statistics, and system statistics.

1.2. Data dictionary related to statistics
Various statistical information can be obtained by querying the relevant data dictionary. The data dictionary related to the statistical information mainly includes the following content.

Table statistics can be obtained by querying DBA_TAB_STATISTICS
Index statistics can be obtained by querying DBA_IND_STATISTICS
Column statistics can be obtained by querying DBA_TAB_COL_STATISTICS

Another type of more refined statistical information is called histogram statistical information, which stores the distribution range of column values. The histogram statistics column value distribution is divided into two categories: Frequency and height balanced. When the number of distinct column values does not exceed 254, Oracle will create a distribution bucket with the same number of distinct values, and each bucket can have one the number of repetitions of the column value. This histogram is called Frequency Histogram. When the number of distribution buckets is greater than the number of non-repetitive values, this histogram is called height Balanced histogram. The system information about the histogram can be obtained by querying the data dictionary view DBA_TAB_HISTOGRAMS.
2. **Statistical information collection method**
There are three ways to collect statistical information, one is automatic collection by Oracle, the other is dynamic collection, and the third is manual collection when needed.

2.1. **Automatic collection of statistics**
Oracle uses the dictionary view `dba_tab_modifications` to monitor the modification of the data. The modification results are updated every 15 minutes.

If you want to update immediately, use `dbms_stats.flush_database_monitoring_info()`.

If the table data is modified more than 10%, Oracle considers the table statistics to expire. Oracle uses the automatic maintenance task `auto optimizer stats collection` to collect statistics on all objects in the database whose statistics are out of date. This automatic task starts at 10 pm from Monday to Friday and lasts for 4 hours. It starts at 6 am on Saturday and Sunday, and lasts 20 hours.

2.2. **Dynamic collection**
When there is no available statistical information for some or all of the objects involved in a query, Oracle uses partial data of each object as a sample as the statistical information for selecting the execution plan.

The way to perform dynamic collection of statistics is determined by the current value of the initialization parameter `OPTIMIZER_DYNAMIC_SAMPLING`, and its default value is 2, which collects the data of 64 blocks.

The main disadvantages of dynamic collection of statistical information: only collects the statistical information of part of the data in the table. When the amount of data in the table is large, it may not reflect the overall distribution of the data in the table and cause Oracle to choose the wrong execution plan.

2.3. **Collect statistics manually**
The database administrator can manually collect statistics through the `dbms_stats` package or the `analyze` table command.

3. **Research design and empirical analysis**

3.1. **Construct test data**
Edit a SQL script and execute it in Oracle SQL*Plus on Oracle database 11g to create a test table `big_table`, which contains 5 million rows of records. Among other columns is `id1` and `id2`. Each value in `id1` is unique and from 1 to 5,000,000, and only three values 0, 1, and 2 appear in column `id2`, so that the repetition rate of each column value is about 33%.

Create indexes on the `id1` and `id2` fields respectively. The following mainly focuses on the usage of the index on `id2` in the execution plan.

```
SQL> create index idx_id1 on big_table(id1) tablespace tbs_idx;
SQL> create index idx_id2 on big_table(id2) tablespace tbs_idx;
```

3.2. **Effect of dynamic collection of statistics**
Use `id2` as the query condition, execute the following query:

```
SQL> set autotrace on
SQL> select * from big_table where id2=2;
```

**Execution Plan**

```
Plan hash value: 3993303771
```
From the above results, it can be seen that there is no available statistical information at this time. Oracle first performs dynamic collection of statistical information. Because the repetition rate of the id2 field value 2 is too high, the index on id2 is not used, but uses full table scan. Execute the following command to prohibit Oracle from performing dynamic collection of statistics.

```
SQL> alter session set OPTIMIZER_DYNAMIC_SAMPLING = 0;
```

Re-execute the following query with id2=2 as the condition. From the execution plan, you can see that the same query command uses the B-tree index on id2 at this time.

```
SQL> select * from big_table where id2=2;
```

### 3.3. Effect of manually collecting statistics

If you execute the following command to manually collect statistics on the id2 column so that Oracle knows the data duplication on the id2 column, Oracle will not use the index on id2 when executing the query.

```
SQL> analyze table big_table compute statistics for columns id2;
SQL> select * from big_table where id2=2;
```
3.4. Impact of expiration of statistics on the execution plan

Execute the following command to modify the big_table table so that the id2 field value of row 49,999,990 in the big_table table is set to 1000:

```
SQL> update big_table set id2=1000 where id1>10;
```

After the above update operation is performed on the id2 column of the big_table table, the statistical information is not updated accordingly, which cannot reflect the actual situation of the current data. If you execute the following query with id2=1000 as the condition at this time, you can find that although the actual repetition rate of this value is very large, Oracle still selects the execution plan based on the old statistics, thinking that the row with id2=1000 does not exist. From the display results of the actual execution plan below, you can see that Oracle uses the index to execute this query task:

```
SQL> alter session set OPTIMIZER_DYNAMIC_SAMPLING=0;
SQL> select * from big_table where id2=1000;
```

```
| Id  | Operation       | Name            | Rows | Bytes | Cost (%CPU) | Time    |
|-----|-----------------|-----------------|------|-------|-------------|---------|
| 0   | SELECT STATEMENT|                 | 1    | 173   | 4           | 00:00:01|
| 1   | TABLE ACCESS BY INDEX ROWID | BIG_TABLE | 1    | 173   | 4           | 00:00:01|
|* 2  | INDEX RANGE SCAN| IDX_ID2         | 1    |       | 3           | 00:00:01|
```

Execute the following query with id2=2, Oracle still uses a full table scan:

```
SQL> select * from big_table where id2=2;
```

```
| Id  | Operation       | Name            | Rows | Bytes | Cost (%CPU) | Time    |
|-----|-----------------|-----------------|------|-------|-------------|---------|
| 0   | SELECT STATEMENT|                 | 2049K| 338M  | 20533       | 00:04:07|
|* 1  | TABLE ACCESS FULL| BIG_TABLE | 2049K| 338M  | 20533       | 00:04:07|
```

Obviously, Oracle made an execution plan based on outdated statistics, and Oracle used the wrong execution plan at this time.

Collect statistics again, and then execute the same query. When id2=1000 is the condition, Oracle uses a full table scan to complete the query, and when id2=2 is the condition, Oracle will use the index. That is, after the statistics are updated, Oracle can select the correct execution plan.

```
SQL> analyze table big_table compute statistics for columns id2;
SQL> select * from big_table where id2=1000;
```

```
| Id  | Operation       | Name            | Rows | Bytes | Cost (%CPU) | Time    |
|-----|-----------------|-----------------|------|-------|-------------|---------|
| 0   | SELECT STATEMENT|                 | 2049K| 338M  | 20533       | 00:04:07|
|* 1  | TABLE ACCESS FULL| BIG_TABLE | 2049K| 338M  | 20533       | 00:04:07|
```
Execution Plan

Plan hash value: 3993303771

| Id | Operation        | Name      | Rows  | Bytes | Cost (%CPU) | Time     |
|----|------------------|-----------|-------|-------|-------------|----------|
|    | SELECT STATEMENT |           | 6147K | 1014M | 20605       (2) | 00:04:08 |
|* 1 | TABLE ACCESS FULL| BIG_TABLE | 6147K | 1014M | 20605       (2) | 00:04:08 |

SQL> select * from big_table where id2=2;

Execution Plan

Plan hash value: 936110477

| Id | Operation                   | Name      | Rows  | Bytes | Cost (%CPU) | Time     |
|----|-----------------------------|-----------|-------|-------|-------------|----------|
|    | SELECT STATEMENT            |           | 4     | 692   | 4           (0) | 00:00:01 |
| 1  | TABLE ACCESS BY INDEX ROWID | BIG_TABLE | 4     | 692   | 4           (0) | 00:00:01 |
|* 2 | INDEX RANGE SCAN            | IDX_ID2   | 4     |       | 3           (0) | 00:00:01 |

4. Conclusion

From the above experimental process, the following conclusions can be obtained:

Manual collection of statistical information should be used, and automatic and dynamic statistical information collection should be avoided to avoid occupying too much resources and affecting the operating efficiency of the Oracle server, and also to avoid dynamically collected statistical information that does not accurately reflect the data in the table.

When the data volume or data distribution in the table changes on a large scale, you should immediately re-collect statistical information manually, so that Oracle can use the latest statistical information to rework a more appropriate execution plan.

References

[1] Jonathan Lewis, Cost-Based Oracle Fundamentals, Apress Inc., 2006
[2] Joe Celko, SQL for Smarties: Advanced SQL Programming, Morgan Kaufmann, 2005
[3] Oracle Database Performance and Scalability, Wiley, 2011
[4] Oracle Database 11g: SQL Tuning Workshop, Oracle
[5] Richard Niemiec, Oracle Database 12c Release 2 Performance Tuning Tips & Techniques, Oracle Press, 2017
[6] Darl Kuhn etc., Pro Oracle Database 18c Administration, 3rd Edition, Apress, 2019
[7] Database Performance Tuning Guide(19c), Oracle, 2020