Top K Stats

Column Level Top K Statistics

This document is an addition to Statistics in Hive. It describes the support of collecting column level top K values for Hive tables (see HIVE-3421).

Scope

In addition to the partition statistics, column level top K values can also be estimated for Hive tables.
The name and top K values of the most skewed column is stored in the partition or non-partitioned table's skewed information, if user did not specify skew.
This works for both newly created and existing tables.
The algorithm for computing top K is based on this paper: Efficient Computation of Frequent and Top-k Elements in Data Streams.

Implementation

Top K statistics are gathered along with partition level statistics. The interface IStatsAggregator needs to add a method aggregateStatsTopK() that reads multiple entries from the temporary storage:

```
...

public interface IStatsAggregator {

...

  /**
 * This method aggregates top K statistics.
   *
 * */
  public List<String> aggregateStatsTopK(String keyPrefix, String statType);

...

}
```

Usage

Top K statistics are disabled by default. The user can set the boolean variable **hive.stats.topk.collect** to be **true** to enable computing top K and putting top K into skewed information.

```
set hive.stats.topk.collect=true;
```

The user can also specify the number of K by setting the integer variable **hive.stats.topk.num**, and the minimal row percentage that a value needs to hold to be in the top K result, by setting the float variable **hive.stats.topk.minpercent**.

```
set hive.stats.topk.num=8;
set hive.stats.topk.minpercent=5.0;
```

Another integer variable, **hive.stats.topk.poolsize**, specifies the number of values to be monitored while computing top K. The accuracy of top K estimate increases as this number gets larger.

```
set hive.stats.topk.poolsize=200;
```

Computing top K for a large number of partitions simultaneously can be stressful to memory. The user can specify the integer variable **hive.stats.topk.maxpartnum** for the maximal number of partitions to collect Top K. When this number is exceeded, top K will be disabled for all the remaining partitions.

```
set hive.stats.topk.maxpartnum=10;
```

In case of JDBC implementation of temporary stored statistics (eg. Derby or MySQL), the user should also specify the column type for top K, by setting the variable **hive.stats.topk.column.type**. By default, **TEXT** is used for MySQL, and **LONG VARCHAR** is used for Derby.

```
set hive.stats.topk.column.type='LONG VARCHAR';
```

Example

The user may set top K related variables at the beginning:

```
set hive.stats.topk.collect=true;
set hive.stats.topk.num=4;
set hive.stats.topk.minpercent=0;
set hive.stats.topk.poolsize=100;
```

Newly Created Tables

Suppose a partitioned table is created without skew, and data is inserted to its partitions:

```
CREATE TABLE table1 (key STRING, value STRING) PARTITIONED BY (ds STRING);
INSERT OVERWRITE TABLE table1 PARTITION (ds='2012-09-07') SELECT * FROM table_src;
```

Top K was computed for the partition while data was inserted. If the user issues the command:

```
DESCRIBE FORMATTED table1 partition (ds='2012-09-07');
```

then among the output, the following will be displayed:

```
...
Skewed Columns:         [value]
Skewed Values:          [[val_348], [val_230], [val_401], [val_70]]
...
```

If the user issues the command:

```
DESCRIBE FORMATTED table1;
```

then among the output, there will not be skewed information, since table level top K is not available for partitioned tables.

For a non-partitioned table:

```
CREATE TABLE table1 (key STRING, value STRING);
INSERT OVERWRITE TABLE table1 SELECT * FROM table_src;
```

If the user issues the command:

```
DESCRIBE FORMATTED table1;
```

then among the output, the following will be displayed:

```
...
Skewed Columns:         [value]
Skewed Values:          [[val_348], [val_230], [val_401], [val_70]]
...
```

When a table is created with skew:

```
set hive.internal.ddl.list.bucketing.enable=true;
CREATE TABLE table1 (key STRING, value STRING) PARTITIONED BY (ds STRING) SKEWED BY (key) on ('38', '49');
INSERT OVERWRITE TABLE table1 PARTITION (ds='2012-09-07') SELECT * FROM table_src;
```

Top K will not be collected, and the user specified skewed information remains. If the user issues the command:

```
DESCRIBE FORMATTED table1 partition (ds='2012-09-07');
```

then among the output, the following will be displayed:

```
...
Skewed Columns:         [key]
Skewed Values:          [[38], [49]]
...
```

Existing Tables

Top K works the same way for ANALYZE commands as for INSERT commands.

Current Status (JIRA)

See HIVE-3421.