Hive on Spark table statistics collection based on Spark Counter

Hive supports collect table statistics based on database, fs, hbase and counter, Hive on MR/Tez/Spark works the same way while collect table statistics based on database, fs or hbase. As we implement Spark Counter based on Spark Accumulator, it would be different for Hive to collect table statistic based on Spark Counter.

# MR Counter, Tez Counter and Spark Counter

Tez Counter is quite similar with MR Counter, so we mainly compare MR Counter with Spark Counter here. MR Counter is used to collect information across cluster, Counter has name and value, and CounterGroup is used to fold same kinds Counters into one group and identified by group name. User can fetch Counter in map/reduce tasks by group name and counter name, then add counter value, finally, user can get aggregated Counter value by group name and counter name in Hive driver side.

Spark Counter works the same way as MR Counter, except that user need to register Counter with group name and counter name at first on driver side before add value for it in spark tasks, which is due to the limitation of Spark Accumulator.

# Hive on MR collect table statistics based on Counter.

## Existed table statistic collection

Hive support syntax like “ANALYZE TABLE tablename COMPUTE STATISTICS”, ANALYZE query would be compiled into 2 tasks: MapredTask and StatsTask. MapredTask contains TableScanOperator which is used to gather table statistics by task, and publish statistic to Counters. StatsTask get all available table statistic Counter and fetch statistics in local driver.

## New created table statistic collection

While set “hive.stats.autogather” as true, Hive would collect table statistic automatically while load data into table with “INSERT OVERWRITE TABLE”. INSERT OVERWRITE would be compiled to at least 3 tasks: MapredTask, MoveTask and StatsTask. MapredTask should contains FileSinkOperator which would write query result into temporary files, MoveTask moves the files to target table source directory. FileSinkOperator would also gather table statistic information, and publish statistic to Counters, StatsTask get all available table statistic Counter and fetch statistics in local driver.

# Hive on Spark collect table statistics based on Spark Counter

As we described before, the main difference between Spark Counter and MR Counter is that, for Spark Counter, user has to register counter before fetch counter and add value in spark tasks. For table statistic counters, group name represents tableName or tableName/partition if it’s a partition table, so the challenge is that how to get table name (and partitions for partition table) in the SparkTask before execute it.

## Existed table statistic collection

ANALYZE query would be compiled into 2 tasks: SparkTask and StatsTask. SparkTask contains TableScanOperator which used to gather table statistics by task, and publish statistic to Counters. For existed table, table name and partition information is wrapped in tableSpec which is stored in StatsWork. As we can fetch StatsTask in SparkTask through the task chain, so we can get StatsWork in SparkTask, and fetch table name and partitions through StatsWork, then register counters based on table name and partitions before execute SparkTask.

## New created table statistic collection

INSERT OVERWRITE would be compiled to at least 3 tasks: SparkTask, MoveTask and StatsTask. SparkTask should contains FileSinkOperator which would gather table statistics and publish to Spark Counter.

While insert overwrite non-partition table or partition table on specified partition, table name and partition information is wrapped in LoadTableDesc which is stored in StatsWork. As we can fetch StatsTask in SparkTask through the task chain, so we can get StatsTask in SparkTask, and fetch table name and partitions through StatsWork, then register counters based on table name and partitions before execute SparkTask.

While insert overwrite table with dynamic partition, partitions information is pushed to StatsTask after MoveTask is executed, it’s impossible to get partition information before execute SparkTask, so I think we can’t support table statistic collection based on Spark Counter in this case.

# Collect table column statistics

Hive table column statistics collection does not depends on Counter, so it should not be related to this topic.