Hive on Spark job statistic implementation design

Here is an example of hive tez job statistic information(MR job is very similar):

14/07/30 16:31:42 [main]: INFO exec.Task: org.apache.tez.common.counters.DAGCounter:

14/07/30 16:31:42 [main]: INFO exec.Task: TOTAL\_LAUNCHED\_TASKS: 2

14/07/30 16:31:42 [main]: INFO exec.Task: DATA\_LOCAL\_TASKS: 1

14/07/30 16:31:42 [main]: INFO exec.Task: File System Counters:

14/07/30 16:31:42 [main]: INFO exec.Task: FILE: BYTES\_READ: 89

14/07/30 16:31:42 [main]: INFO exec.Task: FILE: BYTES\_WRITTEN: 146

14/07/30 16:31:42 [main]: INFO exec.Task: FILE: READ\_OPS: 0

14/07/30 16:31:42 [main]: INFO exec.Task: FILE: LARGE\_READ\_OPS: 0

14/07/30 16:31:42 [main]: INFO exec.Task: FILE: WRITE\_OPS: 0

14/07/30 16:31:42 [main]: INFO exec.Task: HDFS: BYTES\_READ: 297

14/07/30 16:31:42 [main]: INFO exec.Task: HDFS: BYTES\_WRITTEN: 15

14/07/30 16:31:42 [main]: INFO exec.Task: HDFS: READ\_OPS: 13

14/07/30 16:31:42 [main]: INFO exec.Task: HDFS: LARGE\_READ\_OPS: 0

14/07/30 16:31:42 [main]: INFO exec.Task: HDFS: WRITE\_OPS: 2

14/07/30 16:31:42 [main]: INFO exec.Task: org.apache.tez.common.counters.TaskCounter:

14/07/30 16:31:42 [main]: INFO exec.Task: REDUCE\_INPUT\_GROUPS: 2

14/07/30 16:31:42 [main]: INFO exec.Task: REDUCE\_INPUT\_RECORDS: 3

14/07/30 16:31:42 [main]: INFO exec.Task: COMBINE\_INPUT\_RECORDS: 0

14/07/30 16:31:42 [main]: INFO exec.Task: SPILLED\_RECORDS: 6

14/07/30 16:31:42 [main]: INFO exec.Task: NUM\_SHUFFLED\_INPUTS: 1

14/07/30 16:31:42 [main]: INFO exec.Task: NUM\_SKIPPED\_INPUTS: 0

14/07/30 16:31:42 [main]: INFO exec.Task: NUM\_FAILED\_SHUFFLE\_INPUTS: 0

14/07/30 16:31:42 [main]: INFO exec.Task: MERGED\_MAP\_OUTPUTS: 1

14/07/30 16:31:42 [main]: INFO exec.Task: GC\_TIME\_MILLIS: 60

14/07/30 16:31:42 [main]: INFO exec.Task: CPU\_MILLISECONDS: -4860

14/07/30 16:31:42 [main]: INFO exec.Task: PHYSICAL\_MEMORY\_BYTES: 537821184

14/07/30 16:31:42 [main]: INFO exec.Task: VIRTUAL\_MEMORY\_BYTES: 2454896640

14/07/30 16:31:42 [main]: INFO exec.Task: COMMITTED\_HEAP\_BYTES: 408551424

14/07/30 16:31:42 [main]: INFO exec.Task: INPUT\_RECORDS\_PROCESSED: 5

14/07/30 16:31:42 [main]: INFO exec.Task: OUTPUT\_RECORDS: 3

14/07/30 16:31:42 [main]: INFO exec.Task: OUTPUT\_BYTES: 45

14/07/30 16:31:42 [main]: INFO exec.Task: OUTPUT\_BYTES\_WITH\_OVERHEAD: 53

14/07/30 16:31:42 [main]: INFO exec.Task: OUTPUT\_BYTES\_PHYSICAL: 57

14/07/30 16:31:42 [main]: INFO exec.Task: ADDITIONAL\_SPILLS\_BYTES\_WRITTEN: 57

14/07/30 16:31:42 [main]: INFO exec.Task: ADDITIONAL\_SPILLS\_BYTES\_READ: 57

14/07/30 16:31:42 [main]: INFO exec.Task: ADDITIONAL\_SPILL\_COUNT: 0

14/07/30 16:31:42 [main]: INFO exec.Task: SHUFFLE\_BYTES: 57

14/07/30 16:31:42 [main]: INFO exec.Task: SHUFFLE\_BYTES\_DECOMPRESSED: 53

14/07/30 16:31:42 [main]: INFO exec.Task: SHUFFLE\_BYTES\_TO\_MEM: 57

14/07/30 16:31:42 [main]: INFO exec.Task: SHUFFLE\_BYTES\_TO\_DISK: 0

14/07/30 16:31:42 [main]: INFO exec.Task: NUM\_MEM\_TO\_DISK\_MERGES: 0

14/07/30 16:31:42 [main]: INFO exec.Task: NUM\_DISK\_TO\_DISK\_MERGES: 0

14/07/30 16:31:42 [main]: INFO exec.Task: HIVE:

14/07/30 16:31:42 [main]: INFO exec.Task: CREATED\_FILES: 1

14/07/30 16:31:42 [main]: INFO exec.Task: Shuffle Errors:

14/07/30 16:31:42 [main]: INFO exec.Task: BAD\_ID: 0

14/07/30 16:31:42 [main]: INFO exec.Task: CONNECTION: 0

14/07/30 16:31:42 [main]: INFO exec.Task: IO\_ERROR: 0

14/07/30 16:31:42 [main]: INFO exec.Task: WRONG\_LENGTH: 0

14/07/30 16:31:42 [main]: INFO exec.Task: WRONG\_MAP: 0

14/07/30 16:31:42 [main]: INFO exec.Task: WRONG\_REDUCE: 0

14/07/30 16:31:42 [main]: INFO exec.Task: org.apache.hadoop.hive.ql.exec.MapOperator$Counter:

14/07/30 16:31:42 [main]: INFO exec.Task: DESERIALIZE\_ERRORS: 0

MR/Tez job collect statistic information through Counter, these counters could be divided into several different categories:

1. File system statistics(File System Counters in previous example).

2. Job level statistics(org.apache.tez.common.counters.DAGCounter/org.apache.tez.common.counters.TaskCounter/Shuffle Errors in previous example).

3. Hive level statistics(HIVE/org.apache.hadoop.hive.ql.exec.MapOperator$Counter in previous example).

In Hive on Spark, since spark does not support counters and has its own way to collect statistic information, we may implements Hive on Spark job statistic in a different way for each category counter:

1. File system statistic. Since these statistics are collected in file system level, we can get these information through file system API in hive driver side.

2. Job level statistic. Spark has a metrics system which only contains some simple spark job level information, it's not enough for Hive. Besides that, Spark has a listener framework which registered spark listener would get notified by different kinds of spark events, these events contains hive required statistic information, so we could create a specified spark listener, and calculate spark job statistic information through combining all these spark events.

3. Hive level statistic. Spark support accumulator which is similar as counter in MR/Tez, hive driver need to expand spark accumulator in hive driver side to support full counter features, like job/group/name concepts.

4. Combine previous 3 statistics and add a uniform API.