

Auto upload hive jars

Dev Design

Some environments cannot use HADOOP commands and will not perform any checks.

Before starting the KE script execution, add the JAVA code to upload `$/KYLIN_HOME/spark/hive_1_2_2` to HDFS.

Execute Java code using `runToolInternal`

The logic for judging whether to upload is as follows: (The following conditions are also met)

1. Add a configuration item `kylin.engine.hive-client-jar-upload.enable`, which is used to enable automatic uploading of hive jars, the default is false (indicating that it is not enabled by default)
2. `Kylin.engine.spark-conf.spark.sql.hive.metastore.jars.path` is not set, then do not upload
3. Add `_upload_hive_jar_by_pass` file to the parent directory set by `kylin.engine.spark-conf.spark.sql.hive.metastore.jars.path`,
 - a. If it exists, it means that it has been uploaded successfully and there is no need to upload it again.
 - b. If not, upload, upload hive_1_2_2 success, add `_upload_hive_jar_by_pass` to `kylin.engine.spark-conf.spark.sql.hive.metastore.jars.path` set the parent directory
4. `Kylin.engine.spark-conf.spark.sql.hive.metastore.jars.path` set the parent directory to upload hive jar
 - If it does not exist, create the parent directory first and then upload it.
 - Exist will not upload, print log `HDFS dir [] exist, not upload hive client`
5. `$/KYLIN_HOME/spark/hive_1_2_2`
 - a. Upload hive jars when it exists and is not a file;
 - b. Does not exist or is a file, do not upload, print log `$/KYLIN_HOME/spark/hive_1_2_2 needs to be an existing directory`