|  |  |  |
| --- | --- | --- |
| name | value | description |
| mapred.reduce.tasks | -1 | The default number of reduce tasks per job. Typically set to a prime close to the number of available hosts. Ignored when mapred.job.tracker is "local". Hadoop set this to 1 by default, whereas hive uses -1 as its default value. By setting this property to -1, Hive will automatically figure out what should be the number of reducers. |
| hive.exec.reducers.bytes.per.reducer | 1000000000 | size per reducer.The default is 1G, i.e if the input size is 10G, it will use 10 reducers. |
| hive.exec.reducers.max | 999 | max number of reducers will be used. If the one specified in the configuration parameter mapred.reduce.tasks is negative, hive will use this one as the max number of reducers when automatically determine number of reducers. |
| hive.exec.scratchdir | /tmp/hive-${user.name} | Scratch space for Hive jobs |
| hive.test.mode | false | whether hive is running in test mode. If yes, it turns on sampling and prefixes the output tablename |
| hive.test.mode.prefix | test\_ | if hive is running in test mode, prefixes the output table by this string |
| hive.test.mode.samplefreq | 32 | if hive is running in test mode and table is not bucketed, sampling frequency |
| hive.test.mode.nosamplelist |  | if hive is running in test mode, dont sample the above comma seperated list of tables |
| hive.metastore.local | true | controls whether to connect to remove metastore server or open a new metastore server in Hive Client JVM |
| javax.jdo.option.ConnectionURL | jdbc:derby:;databaseName=metastore\_db;create=true | JDBC connect string for a JDBC metastore |
| javax.jdo.option.ConnectionDriverName | org.apache.derby.jdbc.EmbeddedDriver | Driver class name for a JDBC metastore |
| javax.jdo.PersistenceManagerFactoryClass | org.datanucleus.jdo.JDOPersistenceManagerFactory | class implementing the jdo persistence |
| javax.jdo.option.DetachAllOnCommit | true | detaches all objects from session so that they can be used after transaction is committed |
| javax.jdo.option.NonTransactionalRead | true | reads outside of transactions |
| javax.jdo.option.ConnectionUserName | APP | username to use against metastore database |
| javax.jdo.option.ConnectionPassword | mine | password to use against metastore database |
| datanucleus.validateTables | false | validates existing schema against code. turn this on if you want to verify existing schema |
| datanucleus.validateColumns | false | validates existing schema against code. turn this on if you want to verify existing schema |
| datanucleus.validateConstraints | false | validates existing schema against code. turn this on if you want to verify existing schema |
| datanucleus.storeManagerType | rdbms | metadata store type |
| datanucleus.autoCreateSchema | true | creates necessary schema on a startup if one doesn't exist. set this to false, after creating it once |
| datanucleus.autoStartMechanismMode | checked | throw exception if metadata tables are incorrect |
| datancucleus.transactionIsolation | read-committed |  |
| datanuclues.cache.level2 | true | use a level 2 cache. turn this off if metadata is changed independently of hive metastore server |
| datanuclues.cache.level2.type | SOFT | SOFT=soft reference based cache, WEAK=weak reference based cache. |
| hive.metastore.warehouse.dir | /user/hive/warehouse | location of default database for the warehouse |
| hive.metastore.connect.retries | 5 | Number of retries while opening a connection to metastore |
| hive.metastore.rawstore.impl | org.apache.hadoop.hive.metastore.ObjectStore | Name of the class that implements org.apache.hadoop.hive.metastore.rawstore interface. This class is used to store and retrieval of raw metadata objects such as table, database |
| hive.default.fileformat | TextFile | Default file format for CREATE TABLE statement. Options are TextFile and SequenceFile. Users can explicitly say CREATE TABLE ... STORED AS <TEXTFILE|SEQUENCEFILE> to override |
| hive.fileformat.check | true | Whether to check file format or not when loading data files |
| hive.map.aggr | true | Whether to use map-side aggregation in Hive Group By queries |
| hive.groupby.skewindata | false | Whether there is skew in data to optimize group by queries |
| hive.groupby.mapaggr.checkinterval | 100000 | Number of rows after which size of the grouping keys/aggregation classes is performed |
| hive.mapred.local.mem | 0 | For local mode, memory of the mappers/reducers |
| hive.map.aggr.hash.percentmemory | 0.5 | Portion of total memory to be used by map-side grup aggregation hash table |
| hive.map.aggr.hash.min.reduction | 0.5 | Hash aggregation will be turned off if the ratio between hash table size and input rows is bigger than this number. Set to 1 to make sure hash aggregation is never turned off. |
| hive.optimize.cp | true | Whether to enable column pruner |
| hive.optimize.ppd | true | Whether to enable predicate pushdown |
| hive.optimize.pruner | true | Whether to enable the new partition pruner which depends on predicate pushdown. If this is disabled, the old partition pruner which is based on AST will be enabled. |
| hive.optimize.groupby | true | Whether to enable the bucketed group by from bucketed partitions/tables. |
| hive.join.emit.interval | 1000 | How many rows in the right-most join operand Hive should buffer before emitting the join result. |
| hive.join.cache.size | 25000 | How many rows in the joining tables (except the streaming table) should be cached in memory. |
| hive.mapjoin.bucket.cache.size | 100 | How many values in each keys in the map-joined table should be cached in memory. |
| hive.mapjoin.maxsize | 100000 | Maximum # of rows of the small table that can be handled by map-side join. If the size is reached and hive.task.progress is set, a fatal error counter is set and the job will be killed. |
| hive.mapjoin.cache.numrows | 25000 | How many rows should be cached by jdbm for map join. |
| hive.mapred.mode | nonstrict | The mode in which the hive operations are being performed. In strict mode, some risky queries are not allowed to run |
| hive.exec.script.maxerrsize | 100000 | Maximum number of bytes a script is allowed to emit to standard error (per map-reduce task). This prevents runaway scripts from filling logs partitions to capacity |
| hive.exec.script.allow.partial.consumption | false | When enabled, this option allows a user script to exit successfully without consuming all the data from the standard input. |
| hive.script.operator.id.env.var | HIVE\_SCRIPT\_OPERATOR\_ID | Name of the environment variable that holds the unique script operator ID in the user's transform function (the custom mapper/reducer that the user has specified in the query) |
| hive.exec.compress.output | false | This controls whether the final outputs of a query (to a local/hdfs file or a hive table) is compressed. The compression codec and other options are determined from hadoop config variables mapred.output.compress\* |
| hive.exec.compress.intermediate | false | This controls whether intermediate files produced by hive between multiple map-reduce jobs are compressed. The compression codec and other options are determined from hadoop config variables mapred.output.compress\* |
| hive.exec.parallel | false | Whether to execute jobs in parallel |
| hive.hwi.war.file | lib/hive-hwi-@VERSION@.war | This sets the path to the HWI war file, relative to ${HIVE\_HOME}. |
| hive.hwi.listen.host | 0.0.0.0 | This is the host address the Hive Web Interface will listen on |
| hive.hwi.listen.port | 9999 | This is the port the Hive Web Interface will listen on |
| hive.exec.pre.hooks |  | Pre Execute Hook for Tests |
| hive.merge.mapfiles | true | Merge small files at the end of a map-only job |
| hive.merge.mapredfiles | false | Merge small files at the end of any job(map only or map-reduce) |
| hive.heartbeat.interval | 1000 | Send a heartbeat after this interval - used by mapjoin and filter operators |
| hive.merge.size.per.task | 256000000 | Size of merged files at the end of the job |
| hive.script.auto.progress | false | Whether Hive Tranform/Map/Reduce Clause should automatically send progress information to TaskTracker to avoid the task getting killed because of inactivity. Hive sends progress information when the script is outputting to stderr. This option removes the need of periodically producing stderr messages, but users should be cautious because this may prevent infinite loops in the scripts to be killed by TaskTracker. |
| hive.script.serde | org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe | The default serde for trasmitting input data to and reading output data from the user scripts. |
| hive.script.recordreader | org.apache.hadoop.hive.ql.exec.TextRecordReader | The default record reader for reading data from the user scripts. |
| hive.script.recordwriter | org.apache.hadoop.hive.ql.exec.TextRecordWriter | The default record writer for writing data to the user scripts. |
| hive.input.format | org.apache.hadoop.hive.ql.io.HiveInputFormat | The default input format, if it is not specified, the system assigns it. It is set to HiveInputFormat for hadoop versions 17, 18 and 19, whereas it is set to CombinedHiveInputFormat for hadoop 20. The user can always overwrite it - if there is a bug in CombinedHiveInputFormat, it can always be manually set to HiveInputFormat. |
| hive.udtf.auto.progress | false | Whether Hive should automatically send progress information to TaskTracker when using UDTF's to prevent the task getting killed because of inactivity. Users should be cautious because this may prevent TaskTracker from killing tasks with infinte loops. |
| hive.mapred.reduce.tasks.speculative.execution | true | Whether speculative execution for reducers should be turned on. |