# Hive – Optimization

By [puneetha](http://blog.puneethabm.in/author/punbm/" \o "Posts by puneetha) | August 8, 2017

[0 Comment](http://blog.puneethabm.in/hive-optimization/#respond)

1. To set user timezone:
2. SET mapreduce.map.java.opts="-Duser.timezone=UTC";

SET mapreduce.reduce.java.opts="-Duser.timezone=UTC";

1. Compress results
2. -- Determines whether the output of the final map/reduce job in a query is compressed or not.
3. SET hive.exec.compress.output=true;
5. -- Determines whether the output of the intermediate map/reduce jobs in a query is compressed or not.

SET hive.exec.compress.intermediate=true;

1. Avro settings – Compression
2. -- Supported codecs are snappy and deflate.

SET avro.output.codec=snappy;

1. Parquet settings – Compression

SET parquet.compression=snappy;

1. Hive – Hint – STREAMTABLE
3. -- In every map/reduce stage of the join, the table to be streamed can be specified via a hint.

SELECT /\*+ STREAMTABLE(a) \*/ a.val, b.val, c.val FROM a JOIN b ON (a.key = b.key1) JOIN c ON (c.key = b.key1)

All the three tables are joined in a single map/reduce job and the values for a particular value of the key for tables b and c are buffered in the memory in the reducers. Then for each row retrieved from a, the join is computed with the buffered rows. If the STREAMTABLE hint is omitted, Hive streams the rightmost table in the join.  
Reference: https://cwiki.apache.org/confluence/display/Hive/LanguageManual+Joins

1. Joins – while select   
   Mapjoin – Hive enables the optimization about converting common join into mapjoin based on the input file size. Prevent mapjoins for large tables
2. SET hive.auto.convert.join=false;
3. (OR) By specifying the hint "MAPJOIN"
4. SELECT /\*+ MAPJOIN(s) \*/ idOne, idTwo, value
5. FROM bigTable b
6. JOIN smallTableOne s
7. ON (b.idOne = s.idTwo)
8. -- Optimize Chains of Map Joins
9. select /\*+ MAPJOIN(time\_dim, date\_dim) \*/ count(\*) from
10. store\_sales
11. join time\_dim on (ss\_sold\_time\_sk = t\_time\_sk)
12. join date\_dim on (ss\_sold\_date\_sk = d\_date\_sk)
13. where t\_hour = 8 and d\_year = 2002

;

Reference: https://cwiki.apache.org/confluence/display/Hive/LanguageManual+JoinOptimization

1. Sort Merge bucket map join
2. SET hive.input.format=org.apache.hadoop.hive.ql.io.BucketizedHiveInputFormat;
3. SET hive.optimize.bucketmapjoin=true;

SET hive.optimize.bucketmapjoin.sortedmerge=true;

1. Bucket mapjoin
2. -- Enforce bucketing - while inserting
3. SET hive.enforce.bucketing=true;
4. -- Enforce sorting in each bucket - while inserting

SET hive.enforce.sorting=true;

1. Sort memory – The total amount of buffer memory to use while sorting files, in megabytes. By default, gives each merge stream 1MB, which should minimize seeks.
2. SET io.sort.mb=800;

Note: io.sort.mb should be 10 \* io.sort.factor

1. Memory
2. -- Shuffle memory
3. SET mapreduce.reduce.shuffle.memory.limit.percent=0.65;
4. -- Map memory
5. SET mapreduce.map.java.opts=-Xmx8192m;
6. -- Reduce memory

SET mapreduce.reduce.java.opts=-Xmx12288m;

1. Execute jobs in parallel – — Whether to execute jobs in parallel. Applies to MapReduce jobs that can run in parallel, for example jobs processing different source tables before a join. As of Hive 0.14, also applies to move tasks that can run in parallel, for example moving files to insert targets during multi-insert.

SET hive.exec.parallel=true;

1. Fraction of the number of maps in the job which should be complete before reduces are scheduled for the job.

SET mapreduce.job.reduce.slowstart.completedmaps=0.7;

1. Data skew – Skew join Optimization – If your data highly uneven distribution, the data skew could happen in such a way that a small number of compute nodes must handle the bulk of the computation
2. SET hive.optimize.skewjoin=true;
3. -- Syntax: SET hive.skewjoin.key=<skew\_key\_threshold\_number>;

SET hive.skewjoin.key=100000;

-- Skew data could also happen in GROUP BY data too.

-- Hive will first trigger an additional MapReduce job whose map output will randomly distribute to the reducer to avoid data skew.

SET hive.groupby.skewindata=true;

1. Avoid shunting of results in Hue
2. -- File format to use for a query's intermediate results. This takes care of results shunting in Hue

SET hive.query.result.fileformat=SequenceFile;