Enable Compression in Hive

This entry was posted in Hive *on May 2, 2015 by Siva*

For data intensive workloads, I/O operation and network data transfer will take considerable time to complete. By Enabling Compression in Hive we can improve the performance Hive Queries and as well as save the storage space on HDFS cluster.

Table of Contents [hide]

Find Available Compression Codecs in Hive

To enable compression in Hive, first we need to find out the available compression codes on hadoop cluster, and we can use below **set** command to list down the available compression codecs.

```
hive> set io.compression.codecs;
io.compression.codecs=
                    oop.io.compress.GzipCodec,
                    oop.io.compress.DefaultCodec,
      org.apache.hadoop.io.compress.BZip2Codec,
      org.apache.hadoop.io.compress.SnappyCodec
hive>
```


Enable Compression on Intermediate Data

A complex Hive query is usually converted to a series of multi-stage MapReduce jobs after submission, and these jobs will be chained up by the Hive engine to complete the entire query. So “intermediate output” here refers to the output from the previous MapReduce job, which will be used to feed the next MapReduce job as input data.

We can enable compression on Hive Intermediate output by setting the property **hive.exec.compress.intermediate** either from Hive Shell using **set** command **or** at site level in **hive-site.xml** file.

```
  <property>
    <name>hive.exec.compress.intermediate</name>
    <value>true</value>
    <description>
      This controls whether intermediate files produced by Hive between multiple map-reduc
      The compression codec and other options are determined from Hadoop config variables r
    </description>
  </property>
  <property>
    <name>hive.intermediate.compression.codec</name>
    <value>org.apache.hadoop.io.compress.SnappyCodec</value>
    <description/>
  </property>
  <property>
    <name>hive.intermediate.compression.type</name>
    <value>BLOCK</value>
    <description/>
  </property>
```

Or we can set these properties in hive shell as shown below with set commands.

```
hive> set hive.exec.compress.intermediate=true;
hive> set hive.intermediate.compression.codec=org.apache.hadoop.io.compress.SnappyCodec;
hive> set hive.intermediate.compression.type=BLOCK;
hive>
```

Enable Compression on Final Output

We can enable compression on final output in hive shell by setting below properties.

```
  <property>
    <name>hive.exec.compress.output</name>
    <value>true</value>
    <description>
      This controls whether the final outputs of a query (to a local/HDFS file or a Hive t
      The compression codec and other options are determined from Hadoop config variables r
    </description>
  </property>
```

or

```
hive> set hive.exec.compress.output=true;
hive> set mapreduce.output.fileoutputformat.compress=true;
hive> set mapreduce.output.fileoutputformat.compress.codec=org.apache.hadoop.io.compress.G
hive> set mapreduce.output.fileoutputformat.compress.type=BLOCK;
hive>
```


Example Table Creation with Compression Enabled

In the below shell snippet we are creating a new table **compressed_emp** from existing **testemp** table in hive after setting the compression properties to true in the hive shell.

Source Table: testemp contents

```
hive> select * from testemp;
OK
123 Ram Team Lead
345 Siva    Member
678 Krishna Member
Time taken: 0.096 seconds, Fetched: 3 row(s)
hive>
```

Setting Compression properties in Hive Shell:

```
hive> set hive.exec.compress.output=true;
hive> set mapreduce.output.fileoutputformat.compress=true;
hive> set mapreduce.output.fileoutputformat.compress.codec=org.apache.hadoop.io.compress.G
hive> set mapreduce.output.fileoutputformat.compress.type=BLOCK;
hive> set hive.exec.compress.intermediate=true;
```

Target Table compressed_emp Creation:

```
hive> CREATE TABLE compressed_emp ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
    > AS SELECT * FROM testemp;
0150502214545_8eb6915e-8d0d-4109-b743-cb6505dfa26b
```

Thus we can create the output files in gzipped format and we can view the contents of this file with **dfs -text** command.

Share this:

Share 0 Tweet

About Siva

Senior Hadoop developer with 4 years of experience in designing and architecture solutions for the Big Data domain and has been involved with several complex engagements. Technical strengths include Hadoop, YARN, Mapreduce, Hive, Sqoop, Flume, Pig, HBase, Phoenix, Oozie, Falcon, Kafka, Storm, Spark, MySQL and Java.

View all posts by Siva →

Leave a comment

Your email address will not be published. Required fields are marked *

b i link b-quote del ins img ul ol li code more close tags crayon

Name *

Hadoop Online Tutorial
A learning center for hadoop Eco System

Website

Post Comment

One thought on “Enable Compression in Hive”

ravikiran

Reply ↓

July 21, 2020 at 10:31 am

Which one is the best compression codec for intermediate and final output compression in hive ?

Post navigation

← Hadoop Performance Tuning

Hive Performance Tuning →

Search

Search

Core Hadoop

- Big Data
- Hadoop
- Map Reduce

EcoSystem Tools

- Hive
- Pig
- HBase
- Impala

Contact Me

please reach out to us on siv535@gmail.com or +91-9704231873

Hadoop Online Tutorial
A learning center for hadoop Eco System

Recent Comments

- raviteja on Formula to Calculate HDFS nodes storage
- Durgesh Majeti on HDFS Web UI
- Brijesh Yadav on Run Example MapReduce Program
- ravikiran on Enable Compression in Hive
- Riya on Hive Performance Tuning

Hadooptutorial.info

Hadooptutorial.info
2,060 likes

Like Page A learning center for hadoop Eco System Contact Us

Contat Us

Call Us On : +91-9704231873
Mail Us On : siv535@gmail.com

Email ID

Your email address Send

Let's get Social :