LanguageManual LZO

LZO Compression

- LZO Compression
  - General LZO Concepts
  - Prerequisites
    - Lzo/Lzop Installations
    - core-site.xml
    - Table Definition
  - Hive Queries
    - Option 1: Directly Create LZO Files
    - Option 2: Write Custom Java to Create LZO Files

General LZO Concepts

LZO is a lossless data compression library that favors speed over compression ratio. See http://www.oberhumer.com/opensource/lzo and http://www.lzop.org for general information about LZO and see Compressed Data Storage for information about compression in Hive.

Imagine a simple data file that has three columns

- id
- first name
- last name

Let's populate a data file containing 4 records:

```
19630001    john        lennon
19630002    paul        mccartney
19630003    george      harrison
19630004    ringo       starr
```

Let's call the data file `/path/to/dir/names.txt`.

In order to make it into an LZO file, we can use the lzop utility and it will create a `names.txt.lzo` file.

Now copy the file `names.txt.lzo` to HDFS.

Prerequisites

Lzo/Lzop Installations

`lzo` and `lzop` need to be installed on every node in the Hadoop cluster. The details of these installations are beyond the scope of this document.

`core-site.xml`

Add the following to your `core-site.xml`:

- `com.hadoop.compression.lzo.LzoCodec`
- `com.hadoop.compression.lzo.LzopCodec`

For example:

```
<property>
<name>io.compression.codecs</name>
<value>org.apache.hadoop.io.compress.GzipCodec,org.apache.hadoop.io.compress.DefaultCodec,org.apache.hadoop.io.
compress.BZip2Codec,com.hadoop.compression.lzo.LzoCodec,com.hadoop.compression.lzo.LzopCodec</value>
</property>

<property>
<name>io.compression.codec.lzo.class</name>
<value>com.hadoop.compression.lzo.LzoCodec</value>
</property>
```

Next we run the command to create an LZO index file:

```
hadoop jar /path/to/jar/hadoop-lzo-cdh4-0.4.15-gplextras.jar com.hadoop.compression.lzo.LzoIndexer  /path/to
/HDFS/dir/containing/lzo/files
```

This creates `names.txt.lzo` on HDFS.

Table Definition

The following `hive -e` command creates an LZO-compressed external table:

```
hive -e "CREATE EXTERNAL TABLE IF NOT EXISTS hive_table_name (column_1  datatype_1.....column_N datatype_N)
         PARTITIONED BY (partition_col_1 datatype_1 ....col_P  datatype_P)
         ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
         STORED AS INPUTFORMAT  \"com.hadoop.mapred.DeprecatedLzoTextInputFormat\"
                   OUTPUTFORMAT \"org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat\";
```

Note: The double quotes have to be escaped so that the '`hive -e`' command works correctly.

See CREATE TABLE and Hive CLI for information about command syntax.

Hive Queries

Option 1: Directly Create LZO Files

1. Directly create LZO files as the output of the Hive query.
2. Use `lzop` command utility or your custom Java to generate `.lzo.index` for the `.lzo` files.

Hive Query Parameters

```
SET mapreduce.output.fileoutputformat.compress.codec=com.hadoop.compression.lzo.LzoCodec
SET hive.exec.compress.output=true
SET mapreduce.output.fileoutputformat.compress=true
```

For example:

```
hive -e "SET mapreduce.output.fileoutputformat.compress.codec=com.hadoop.compression.lzo.LzoCodec; SET hive.
exec.compress.output=true;SET mapreduce.output.fileoutputformat.compress=true; <query-string>"
```

Note: If the data sets are large or number of output files are large , then this option does not work.

Option 2: Write Custom Java to Create LZO Files

1. Create text files as the output of the Hive query.
2. Write custom Java code to
    a. convert Hive query generated text files to `.lzo` files
    b. generate `.lzo.index` files for the `.lzo` files generated above

Hive Query Parameters

Prefix the query string with these parameters:

```
SET hive.exec.compress.output=false
SET mapreduce.output.fileoutputformat.compress=false
```

For example:

```
hive -e "SET hive.exec.compress.output=false;SET mapreduce.output.fileoutputformat.compress=false;<query-
string>"
```