Hive 0.14.0 - Interning Strings in Thrift-generated Metastore API Classes

Goal:

To reduce memory usage in HiveServer2. This is the first of a series of changes to improve Hive memory consumption.

Proposed change:

Strings are usually a Java application's biggest problem in terms of memory consumption. We saw that by interning strings in Properties objects (HIVE-6262), we were able to save tremendous amounts of memory. It turns out that It's possible to save even more memory in the Thrift-generated Metastore API classes, which do not internalize strings by default.

By running one particular (rather complex) test query on a one-node test cluster, we were able to save about 8 MB, or 24%, of old gen memory at steady state. This test was run with a maximum heap size of 2 GB on a 500 GB dataset with about 1800 partitions, and the analysis was done with YourKit. Because the number of these Thrift-generated classes is proportional to the number of partitions in the data, there should be a tremendous amount of savings for users with tens of thousands of partitions.

We use the com.google.common.collect.Interner from the Google Guava libraries to internalize our strings.

Because we are interning strings in Thrift-generated files, it is difficult to expect that changes in these files will not be rolled back upon future commits. Therefore, we add a maven-replacer-plugin to perform these code changes after generating the source code. This is done in the process-sources phase of the Maven build, using the hive/metastore/thrift-replacements.txt file to specify the changes in the code. The new plugin is added to the build of hive/metastore/pom.xml.

Specifics:

There are four classes with String-based instance fields that we can internalize:

org.apache.hadoop.hive.metastore.api.FieldSchema
org.apache.hadoop.hive.metastore.api.Partition
org.apache.hadoop.hive.metastore.api.SerDeInfo
org.apache.hadoop.hive.metastore.api.StorageDescriptor

In this patch, we internalize the strings found in these four classes' constructors and setters; we even internalize the List<String>s and Map<String, String>s by going through their elements and internalizing them.

The interning is implemented using a static com.google.common.collect.Interner object, in the org.apache.hive.common.util.HiveStringUtils class. The new methods are:

public static String HiveStringUtils.intern(String)
public static List<String> HiveStringUtils.intern(List<String>)
public static Map<String, String> HiveStringUtils.intern(Map<String, String>)

Possible Drawbacks:

There might be a performance tradeoff with constructing or setting the Map objects in these classes as each key and value is internalized; there may also be smaller tradeoffs with the Strings and Lists. However, because the copy constructors for these classes are most frequently used, the performance hit is likely to be minimal.