Hadoop-compatible Input-Output Format for Hive

Overview

This is a proposal for adding API to Hive which allows reading and writing using a Hadoop compatible API. Specifically, the interfaces being implemented are:

- InputFormat: http://hadoop.apache.org/docs/mapreduce/r0.21.0/api/org/apache/hadoop/mapreduce/InputFormat.html
- OutputFormat: http://hadoop.apache.org/docs/mapreduce/r0.21.0/api/org/apache/hadoop/mapreduce/OutputFormat.html

The classes will be named HiveApiInputFormat and HiveApiOutputFormat.

See HIVE-3752 for discussion of this proposal.

InputFormat (reading from Hive)

Usage:

1. Create a HiveInputDescription object.
2. Fill it with information about the table to read from (with database, partition, columns).
3. Initialize HiveApiInputFormat with the information.
4. Go to town using HiveApiInputFormat with your Hadoop-compatible reading system.

More detailed information:

- The HiveInputDescription describes the database, table and columns to select. It also has a partition filter property that can be used to read from only the partitions that match the filter statement.
- HiveApiInputFormat supports reading from multiple tables by having a concept of profiles. Each profile stores its input description in a separate section, and the HiveApiInputFormat has a member which tells it which profile to read from. When initializing the input data in HiveApiInputFormat you can pair it with a profile. If no profile is selected then a default profile is used.

Future plans:

- Lots of performance work. Expose more direct byte[] sort of semantics.
- Filtering of rows returned.

OutputFormat (writing to Hive)

Usage:

1. Create a HiveOutputDescription object.
2. Fill it with information about the table to write to (with database and partition).
3. Initialize HiveApiOutputFormat with the information.
4. Go to town using HiveApiOutputFormat with your Hadoop-compatible writing system.