Load and Store Interfaces

Table of contents

1 Set Up

The HCatLoader and HCatStorer interfaces are used with Pig scripts to read and write data in HCatalog managed tables. If you run your Pig script using the "pig" command (the bin/pig Perl script) no set up is required.

```
$ pig mypig.script
```

If you run your Pig script using the "java" command (java -cp pig.jar...), then the hcat jar needs to be included in the classpath of the java command line (using the -cp option). Additionally, the following properties are required in the command line:

- -Dhcat.metastore.uri=thrift://<hcatalog server hostname>:9080
- -Dhcat.metastore.principal=<hcatalog server kerberos principal>

```
$ java -cp pig.jar hcatalog.jar
     -Dhcat.metastore.uri=thrift://<hcatalog server hostname>:9080
     -Dhcat.metastore.principal=<hcatalog server kerberos principal> myscript.pig
```

Authentication

If a failure results in a message like "2010-11-03 16:17:28,225 WARN hive.metastore ... - Unable to connect metastore with URI thrift://..." in /tmp/<username>/hive.log, then make sure you have run "kinit <username>@FOO.COM" to get a kerberos ticket and to be able to authenticate to the HCatalog server.

2 HCatLoader

HCatLoader is used with Pig scripts to read data from HCatalog managed tables.

2.1 Usage

HCatLoader is accessed via a Pig load statement.

```
A = LOAD 'dbname.tablename' USING org.apache.hcatalog.pig.HCatLoader();
```

Assumptions

You must specify the database name and table name using this format: 'dbname.tablename'. Both the database and table must be created prior to running your Pig script. The Hive metastore lets you create tables without specifying a database; if you created tables this way, then the database name is 'default' and the string becomes 'default.tablename'.

If the table is partitioned, you can indicate which partitions to scan by immediately following the load statement with a partition filter statement (see Examples).

2.2 HCatalog Data Types

Restrictions apply to the types of columns HCatLoader can read.

HCatLoader can read **only** the data types listed in the table. The table shows how Pig will interpret the HCatalog data type.

(Note: HCatalog does not support type Boolean.)

HCatalog Data Type	**Pig Data Type**
primitives (int, long, float, double, string)	int, long, float, double string to chararray
map (key type should be string, valuetype can be a primitive listed above)	map
List<primitive> or List<map> where map is of the type noted above	bag, with the primitive or map type as the field in each tuple of the bag
struct<primitive fields>	tuple
List<struct<primitive fields>>	bag, where each tuple in the bag maps to struct <primitive fields>

2.3 Examples

This load statement will load all partitions of the specified table.

```
/* myscript.pig */
A = LOAD 'dbname.tablename' USING org.apache.hcatalog.pig.HCatLoader();
...
...
```

If only some partitions of the specified table are needed, include a partition filter statement **immediately** following the load statement. The filter statement can include conditions on partition as well as non-partition columns.

```
/* myscript.pig */
A = LOAD 'dbname.tablename' USING  org.apache.hcatalog.pig.HCatLoader();

B = filter A by date == '20100819' and by age < 30; -- datestamp is a partition column; age
 is not

C = filter A by date == '20100819' and by country == 'US'; -- datestamp and country are
 partition columns
...
...
```

Certain combinations of conditions on partition and non-partition columns are not allowed in filter statements. For example, the following script results in this error message:

```
ERROR 1112: Unsupported query: You have an partition column (datestamp ) in a construction like: (pcond and ...) or ( pcond and ...) where pcond is a condition on a partition column.
```

A workaround is to restructure the filter condition by splitting it into multiple filter conditions, with the first condition immediately following the load statement.

```
/* This script produces an ERROR */

A = LOAD 'default.search_austria' USING org.apache.hcatalog.pig.HCatLoader();
B = FILTER A BY
    (   (datestamp < '20091103' AND browser < 50)
     OR (action == 'click' and browser > 100)
    );
...
...
```

3 HCatStorer

HCatStorer is used with Pig scripts to write data to HCatalog managed tables.

3.1 Usage

HCatStorer is accessed via a Pig store statement.

```
A = LOAD ...
B = FOREACH A ...
...
...
my_processed_data = ...

STORE my_processed_data INTO 'dbname.tablename'
    USING
 org.apache.hcatalog.pig.HCatStorer('month=12,date=25,hour=0300','a:int,b:chararray,c:map[]');
```

Assumptions

You must specify the database name and table name using this format: 'dbname.tablename'. Both the database and table must be created prior to running your Pig script. The Hive metastore lets you create tables without specifying a database; if you created tables this way, then the database name is 'default' and string becomes 'default.tablename'.

For the USING clause, you can have two string arguments:

- The first string argument represents key/value pairs for partition. This is a mandatory argument. In the above example, month, date and hour are columns on which table is partitioned. The values for partition keys should NOT be quoted, even if the partition key is defined to be of string type.

- The second string argument is the Pig schema for the data that will be written. This argument is optional, and if no schema is specified, a schema will be computed by Pig. If a schema is provided, it must match with the schema computed by Pig. (See also: Partition Schema Semantics.)

3.2 HCatalog Data Types

Restrictions apply to the types of columns HCatStorer can write.

HCatStorer can write **only** the data types listed in the table. The table shows how Pig will interpret the HCatalog data type.

(Note: HCatalog does not support type Boolean.)

HCatalog Data Type	**Pig Data Type**
primitives (int, long, float, double, string)	int, long, float, double, string **Note:** HCatStorer does NOT support writing table columns of type smallint or tinyint. To be able to write form Pig using the HCatalog storer, table columns must by of type int or bigint.
map (key type should be string, valuetype can be a primitive listed above)	map
List<primitive> or List<map> where map is of the type noted above	bag, with the primitive or map type as the field in each tuple of the bag
struct<primitive fields>	tuple
List<struct<primitive fields>>	bag, where each tuple in the bag maps to struct <primitive fields>