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. No HCatalog-specific setup is required for these interfaces.

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 'tablename' USING org.apache.hcatalog.pig.HCatLoader();
```

Assumptions

You must specify the table name in single quotes: LOAD 'tablename'. If you are using a non-default database you must specify your input as 'dbname.tablename'. If you are using Pig 0.9.2 or earlier, you must create your database and table prior to running the Pig script. Beginning with Pig 0.10 you can issue these create commands in Pig using the SQL command.

The Hive metastore lets you create tables without specifying a database; if you created tables this way, then the database name is 'default' and is not required when specifying the table for HCatLoader.

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

2.2 HCatalog Data Types

Restrictions apply to the types of columns HCatLoader can read from HCatalog-managed tables.

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

Primitives	
HCatalog Data Type	**Pig Data Type**
int	int
long	long
float	float

Primitives	
double	double
string	chararray
boolean	boolean
binary	bytearray
Complex Types	
HCatalog Data Type	**Pig Data Type**
map (key type should be string)	map
List<*any type*>	bag
struct<*any type fields*>	tuple

2.3 Running Pig with HCatalog

Pig does not automatically pick up HCatalog jars. To bring in the necessary jars, you can either use a flag in the `pig` command or set the environment variables PIG_CLASSPATH and PIG_OPTS as described below.

The -useHCatalog Flag

To bring in the appropriate jars for working with HCatalog, simply include the following flag:

```
pig -useHCatalog
```

Jars and Configuration Files

For Pig commands that omit `-useHCatalog`, you need to tell Pig where to find your HCatalog jars and the Hive jars used by the HCatalog client. To do this, you must define the environment variable PIG_CLASSPATH with the appropriate jars.

HCatalog can tell you the jars it needs. In order to do this it needs to know where Hadoop and Hive are installed. Also, you need to tell Pig the URI for your metastore, in the PIG_OPTS variable.

In the case where you have installed Hadoop and HCatalog via tar, you can do this:

```
export HADOOP_HOME=<path_to_hadoop_install>

export HCAT_HOME=<path_to_hcat_install>
```

```
export HIVE_HOME=<path_to_hive_install>

export PIG_CLASSPATH=$HCAT_HOME/share/hcatalog/hcatalog-*.jar:\
$HIVE_HOME/lib/hive-metastore-*.jar:$HIVE_HOME/lib/libthrift-*.jar:\
$HIVE_HOME/lib/hive-exec-*.jar:$HIVE_HOME/lib/libfb303-*.jar:\
$HIVE_HOME/lib/jdo2-api-*-ec.jar:$HIVE_HOME/conf:$HADOOP_HOME/conf:\
$HIVE_HOME/lib/slf4j-api-*.jar

export PIG_OPTS=-Dhive.metastore.uris=thrift://<hostname>:<port>
```

Or you can pass the jars in your command line:

```
<path_to_pig_install>/bin/pig -Dpig.additional.jars=\
$HCAT_HOME/share/hcatalog/hcatalog-*.jar:\
$HIVE_HOME/lib/hive-metastore-*.jar:$HIVE_HOME/lib/libthrift-*.jar:\
$HIVE_HOME/lib/hive-exec-*.jar:$HIVE_HOME/lib/libfb303-*.jar:\
$HIVE_HOME/lib/jdo2-api-*-ec.jar:$HIVE_HOME/lib/slf4j-api-*.jar  <script.pig>
```

The version number found in each filepath will be substituted for `*`. For example, HCatalog release 0.5.0 uses these jars and conf files:

- `$HCAT_HOME/share/hcatalog/hcatalog-core-0.5.0.jar`
- `$HCAT_HOME/share/hcatalog/hcatalog-pig-adapter-0.5.0.jar`
- `$HIVE_HOME/lib/hive-metastore-0.10.0.jar`
- `$HIVE_HOME/lib/libthrift-0.7.0.jar`
- `$HIVE_HOME/lib/hive-exec-0.10.0.jar`
- `$HIVE_HOME/lib/libfb303-0.7.0.jar`
- `$HIVE_HOME/lib/jdo2-api-2.3-ec.jar`
- `$HIVE_HOME/conf`
- `$HADOOP_HOME/conf`
- `$HIVE_HOME/lib/slf4j-api-1.6.1.jar`

Authentication

If you are using a secure cluster and 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.

2.4 Load Examples

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

```
/* myscript.pig */
A = LOAD '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 in the data flow. (In the script, however, a filter statement might not immediately follow its load statement.) The filter statement can include conditions on partition as well as non-partition columns.

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

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

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

To scan a whole table, for example:

```
a = load 'student_data' using org.apache.hcatalog.pig.HCatLoader();
b = foreach a generate name, age;
```

Notice that the schema is automatically provided to Pig; there's no need to declare name and age as fields, as if you were loading from a file.

To scan a single partition of the table web_logs partitioned by the column datestamp, for example:

```
a = load 'web_logs' using org.apache.hcatalog.pig.HCatLoader();
b = filter a by datestamp == '20110924';
```

Pig will push the datestamp filter shown here to HCatalog, so that HCatalog knows to just scan the partition where datestamp = '20110924'. You can combine this filter with others via 'and':

```
a = load 'web_logs' using org.apache.hcatalog.pig.HCatLoader();
b = filter a by datestamp == '20110924' and user is not null;
```

Pig will split the above filter, pushing the datestamp portion to HCatalog and retaining the '`user is not null`' part to apply itself. You can also give a more complex filter to retrieve a set of partitions.

Filter Operators

A filter can contain the operators 'and', 'or', '()', '==', '!=', '<', '>', '<=' and '>='.

For example:

```
```

```
a = load 'web_logs' using org.apache.hcatalog.pig.HCatLoader();
b = filter a by datestamp > '20110924';
```

A complex filter can have various combinations of operators, such as:

```
a = load 'web_logs' using org.apache.hcatalog.pig.HCatLoader();
b = filter a by datestamp == '20110924' or datestamp == '20110925';
```

These two examples have the same effect:

```
a = load 'web_logs' using org.apache.hcatalog.pig.HCatLoader();
b = filter a by datestamp >= '20110924' and datestamp <= '20110925';


a = load 'web_logs' using org.apache.hcatalog.pig.HCatLoader();
b = filter a by datestamp <= '20110925' and datestamp >= '20110924';
```

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 'tablename'
   USING org.apache.hcatalog.pig.HCatStorer();
```

Assumptions

You must specify the table name in single quotes: LOAD 'tablename'. Both the database and table must be created prior to running your Pig script. If you are using a non-default database you must specify your input as 'dbname.tablename'. If you are using Pig 0.9.2 or earlier, you must create your database and table prior to running the Pig script. Beginning with Pig 0.10 you can issue these create commands in Pig using the SQL command.

The Hive metastore lets you create tables without specifying a database; if you created tables this way, then the database name is 'default' and you do not need to specify the database name in the store statement.

For the USING clause, you can have a string argument that represents key/value pairs for partition. This is a mandatory argument when you are writing to a partitioned table and the

partition column is not in the output column. The values for partition keys should *NOT* be quoted.

If partition columns are present in data they need not be specified as a STORE argument. Instead HCatalog will use these values to place records in the appropriate partition(s). It is valid to specify some partition keys in the STORE statement and have other partition keys in the data.

3.2 Store Examples

You can write to a non-partitioned table simply by using HCatStorer. The contents of the table will be overwritten:

```
store z into 'web_data' using org.apache.hcatalog.pig.HCatStorer();
```

To add one new partition to a partitioned table, specify the partition value in the store function. Pay careful attention to the quoting, as the whole string must be single quoted and separated with an equals sign:

```
store z into 'web_data' using org.apache.hcatalog.pig.HCatStorer('datestamp=20110924');
```

To write into multiple partitions at once, make sure that the partition column is present in your data, then call HCatStorer with no argument:

```
store z into 'web_data' using org.apache.hcatalog.pig.HCatStorer();
  -- datestamp must be a field in the relation z
```

3.3 HCatalog Data Types

Restrictions apply to the types of columns HCatStorer can write to HCatalog-managed tables.

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

Primitives	
Pig Data Type	**HCatalog Data Type**
int	int
long	long
float	float
double	double
chararray	string

Primitives	
boolean	boolean
bytearray	binary
Complex Types	
Pig Data Type	**HCatalog Data Type**
map	map (key type should be string)
bag	List<*any type*>
tuple	struct<*any type fields*>