Hive UDF

http://hortonworks.com/wp-content/uploads/downloads/2013/09/HWX.Qubole.Hive_.UDF_.Guide_.1.0.pdf

UT Dallas

UDF: User-Defined Functions

- UDF is a Great tool for extending HiveQL.
- Written in Jave and then integrated to Hive as built-in functions.
- Can be called from a Hive query.

- Hive Built-in functions:
- *hive> SHOW FUNCTIONS;*
- *hive> DESCRIBE FUNCTION concat;*
- *hive> DESCRIBE FUNCTION EXTENDED concat;*

  concat(str1, str2, ... strN) - returns the concatenation of str1, str2, ... strN Returns NULL if any argument is NULL.

  Example: > SELECT concat('abc', 'def') FROM src LIMIT 1; 'abcdef'

UDF cont'd

- SELECT concat(column1,column2) AS x FROM table;

- **Standard Functions**
- round(), floor(), abs()
- ucase(), reverse()

- **Aggregate Functions**
- sum(), avg(), count(), min() and max()

UDF cont’d

- UDTFs: User Defined Table Generating Functions
- hive> select split(bday, '-') as bd_func from littlebigdata;
- ["2","12","1981"]
- ["10","10","2004"]
- ["4","5","1974"]

- hive> select explode(split(bday, '-')) as bd_func from littlebigdata;

```
2
12
1981
10
10
2004
4
5
1974
```

Custom UDF Example

- my_to_upper function
- We will use the following:
- *File name: littlebigdata.txt* with the following content:

```
edward capriolo,edward@media6degrees.com,2-12-1981,209.191.139.200,M,10
bob,bob@test.net,10-10-2004,10.10.10.1,M,50
sara connor,sara@sky.net,4-5-1974,64.64.5.1,F,2
```

```
hive > CREATE TABLE IF NOT EXISTS littlebigdata(  name STRING,  email STRING,  bday STRING,  ip STRING,  gender STRING,  anum INT)
 ROW FORMAT DELIMITED FIELDS TERMINATED BY ',';
hive> LOAD DATA LOCAL INPATH ‘unix/path/to/littlebigdata.txt'  INTO TABLE littlebigdata;
```

Java code

```
import org.apache.hadoop.hive.ql.exec.UDF;
import org.apache.hadoop.hive.ql.exec.Description;
import org.apache.hadoop.io.Text;

@Description(name = "my_to_upper",
value = "_FUNC_(str) - Converts a string to uppercase",
extended = "Example:\n"
+ "  > SELECT my_to_upper(author_name) FROM authors a;")

public class ToUpper extends UDF {
    public Text evaluate(Text s) {
        Text to_value = new Text("");
        if (s != null) {
            try {
                to_value.set(s.toString().toUpperCase());
            } catch (Exception e) { // Should never happen
                to_value = new Text(s);
            }
        }
        return to_value;
    }
}
```

Java code

- Extend UDF class and write the evaluate() function.
- evalute() methods can be overloaded.
- @Description(...) is an optional Java *annotation for* DESCRIBE FUNCTION ... command.
  - _FUNC_ strings will be replaced with the function name.

- Arguments and return types are what Hive can serialze (e.g., for numbers, use int, Integer wrapper object, or IntWritable which Hadoop wrapper for integers).
- In previous example we used Text

Compile, JAR and Create func.

In the Unix shell:

```
$ mkdir udf_classes_toUpper;
$ javac -classpath /usr/local/hive-0.9.0/lib/hive-exec-0.9.0.jar:/usr/local/hadoop-1.2.1/hadoop-core-1.2.1.jar -d udf_classes_toUpper ToUpper.java
$  jar -cvf toupper.jar -C udf_classes_toUpper/ .
```

In the Hive shell:

```
hive>  add jar /people/cs/l/lkhan/toupper.jar;
hive>  CREATE TEMPORARY FUNCTION my_to_upper as 'ToUpper'; -- ToUpper is the Jave class name
```

Function use

```
hive> desc function extended my_to_upper;
my_to_upper(str) - Converts a string to uppercase
Example:
  > SELECT my_to_upper(author_name) FROM authors a;


hive> select name, my_to_upper(name) from littlebigdata;
edward capriolo EDWARD CAPRIOLO
bob     BOB
sara connor     SARA CONNOR
```

Dropping a temp UDF

- hive> DROP TEMPORARY FUNCTION IF EXISTS my_to_upper;

- To make a function permanent:
  - Code should be added to Hive source code (FunctionRegistry class)
  - Rebuild Hive and redeploy.

Thank you

- Programming Hive book
- http://snowplowanalytics.com/blog/2013/02/08/writing-hive-udfs-and-serdes/