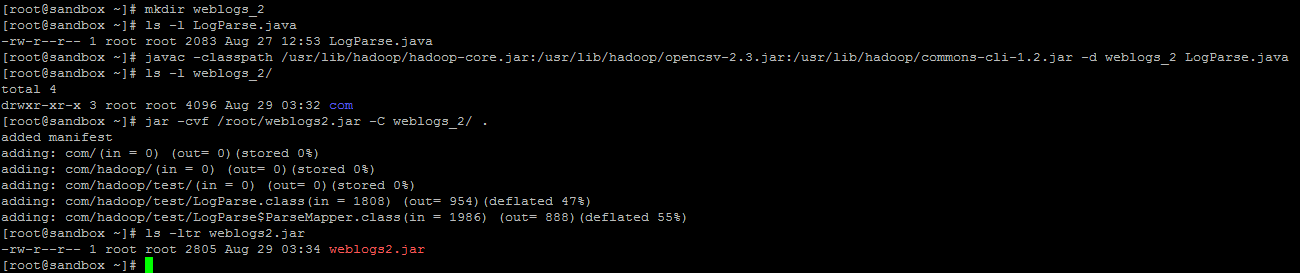
|  |  |
| --- | --- |
| Objective | To Process weblogs (click stream data) |
| Sample weblog |  |
| Expected fields | h = The remote requestor ID, in IP format (example: 165.32.124.3) if Directory Name Service (DNS) is not enabled  I = The remote username via identd, which is blank (indicated by a ‘-’) unless the client is Unix or Linux.  u = The HTTP authenticated username, which is likely blank unless your site requires user registration  r = The entire text of the HTTP request  s = The returned HTTP status code  b = The number of bytes in the returned page  {referer}i = most recent referring URL, taken from the ‘referer’ field of the HTTP request header  {user-agent}i = the requesting software type, usually a browser like Microsoft IE or Netscape, but it can also be things like search engine robots |
| Solutions | We have 2 solutions for this.   1. Using a java program compiled as jar. 2. Using a serde in hive. |

**Using a java program compiled as jar**

|  |  |
| --- | --- |
| Used Files: |  |

1. Place the jar files in hadoop home dir.( /usr/lib/hadoop)
2. Extract logparse.java file from the zip folder and place it in the current dir where you want to compile and create jar file.
3. Making javac and jar commands available to use.
4. PATH=/usr/lib64/qt-3.3/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin:/usr/jdk/jdk1.6.0\_31/bin
5. Export PATH
6. Compiling and creating a jar file.



1. Making a directory weblogs\_2 to keep my compiled java classes.
2. LogParse.java file extracted from zip attached is in current location.
3. Compling LogParse.java file with below command.

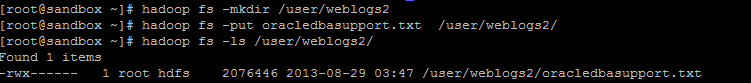
javac -classpath /usr/lib/hadoop/hadoop-core.jar:/usr/lib/hadoop/opencsv-2.3.jar:/usr/lib/hadoop/commons-cli-1.2.jar -d weblogs\_2 LogParse.java

1. A directory with “com” is created under weblogs\_2 dir.
2. Creating jar file of weblogs\_2 dir with a name weblogs2.jar.

jar -cvf /root/weblogs2.jar -C weblogs\_2/ .

1. Now we use this weblogs2.jar file on the weblogs sample file which we have uploaded onto the hdfs to make the data clean(removing “” and making fields tab separated and removing unnecessary spaces) for loading into hive tables.

Note: we can change LogParse.java file so as to meet our requirements and make a new jar file.



1. Making a weblogs2 directory on HDFS
2. Uploading our weblogs file onto HDFS
3. Use the below command to clean the data

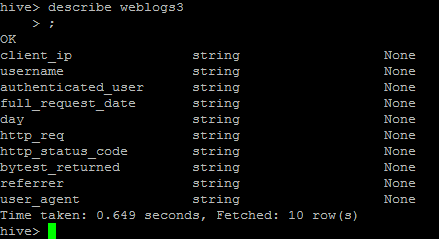
hadoop jar LogParse.jar com.hadoop.test.LogParse -libjars /usr/lib/hadoop/opencsv-2.3.jar /user/weblogs2/oracledbasupport.txt /user/weblogs2/output/

/user/weblogs2/output/ is the directory where the output is collected with name as part\*

1. Below is the output



1. Now we create a hive table to load this data from part-r-0000



Create table weblogs3 (

Client\_ip string,

Username string,

Authenticated\_user string,

Full\_request\_date string,

Day string,

http\_req string,

http\_status\_code string,

bytest\_returned string,

referrer string,

user\_agent string) row format delimited fields terminated by '\t';

1. Loading data into this table weblogs3.

Load data inpath ‘/user/weblogs3/output/part-r-0000’ into table weblogs3;

1. Now we have our data in table weblogs3.

**Using a serde in hive.**

Note: I am still working on the date part and the regex part for this method.

1. In hive shell.
2. add jar /usr/lib/hive/lib/hive-contrib-0.11.0.1.3.0.0-107.jar
3. create a tables based on the serde properties.

CREATE TABLE weblogs1(

host STRING,

identity STRING,

user STRING,

time STRING,

request STRING,

status STRING,

size STRING,

referer STRING,

agent STRING)

ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.RegexSerDe'

WITH SERDEPROPERTIES (

"input.regex" = "([^ ]\*) ([^ ]\*) ([^ ]\*) (-|\\[[^\\]]\*\\]) ([^ \"]\*|\"[^\"]\*\") (-|[0-9]\*) (-|[0-9]\*)(?: ([^ \"]\*|\".\*\") ([^ \"]\*|\".\*\"))?",

"output.format.string" = "%1$s %2$s %3$s %4$s %5$s %6$s %7$s %8$s %9$s"

);

1. Load data into this weblogs1 table.

Load data local inpath ‘/root/ oracledbasupport.txt’ into table weblogs1;

1. We have our data in table now.