Data Science

hw10 . Hive

2017년 6월 8일

00 반
충남대학교 컴퓨터공학과
201202154
조윤재

❖ Contents.

1. Hive 설치 및 실행

1) 가장 먼저 wget 명령어를 통해 master node에 hive를 다운로드 하고 압축을 풀어줍니다.

sudo wget http://apache.mirror.cdnetworks.com/hive/hive-1.2.2/apache-hive-1.2.2-bin.tar.gz

tar -xvf apache-hive-1.2.2-bin.tar.gz

2) 압축이 해제된 디렉토리를 $HIVE_HOME 으로, 내부의 bin 디렉토리를 Path로 하여 bashrc 에 추가합니다.

```
129 export JAVA_HOME=/usr/lib/jvm/java-8-oracle
130 export HADOOP_HOME=/home/datascience/hadoop-2.7.3
131 export HADOOP_CONFIG_HOME=$HADOOP_HOME/etc/hadoop
132 export HIVE_HOME=/home/datascience/apache-hive-1.2.2-bin
133 export PATH=$PATH:$HADOOP_HOME/bin
134 export PATH=$PATH:$HADOOP_HOME/sbin
135 export PATH=$PATH:$HIVE_HOME/bin
```

3) Master node 에서 각 slave node로 hive 디렉토리를 복사합니다.

*sync -avz **apache-hive-1.2.2-bin** datascience@slave1:/home/datascience/*
*rsync -avz **apache-hive-1.2.2-bin** datascience@slave2:/home/datascience/*
*rsync -avz **apache-hive-1.2.2-bin** datascience@slave3:/home/datascience/*

4) Hadoop을 Master node와 sSlave node에서 실행 하고, hive를 실행합니다.

```
[datascience@master ~/hadoop-2.7.3 $ jps
25426 SecondaryNameNode
25139 NameNode
25255 DataNode
25611 NodeManager
25709 Jps
[datascience@master ~/hadoop-2.7.3 $ hive

Logging initialized using configuration in jar:file:/home/datascience/apache-hiv
e-1.2.2-bin/lib/hive-common-1.2.2.jar!/hive-log4j.properties
hive>
```

```
[datascience@slave1 ~ $ jps
2437 ResourceManager
2552 NodeManager
2316 DataNode
2591 Jps
```

```
[datascience@slave2 ~ $ jps
2531 Jps
2310 DataNode
2429 NodeManager
```

```
[datascience@slave3 ~ $ jps
2371 DataNode
2489 NodeManager
2591 Jps
```

< Hive가 Master에서 실행되고, slave에서 나머지 process가 실행되는 모습>

2. 타슈데이터 Hive에 업로드하기

로컬에 저장된 tashu.csv 파일을 rsync 명령어를 통해 각 node에 전달합니다.

rsync -avz tashu.csv datascience@192.168.99.100:/home/datascience/
rsync -avz tashu.csv datascience@192.168.99.101:/home/datascience/
rsync -avz tashu.csv datascience@192.168.99.102:/home/datascience/
rsync -avz tashu.csv datascience@192.168.99.103:/home/datascience/

3. 2) 에서 query 문을 통해 upload 합니다.

3. 타슈데이터 Hive를 통해 분석

Hive가 실행되면 SQL 처럼 Query를 실행 할 수 있는 상태입니다.

1) 먼저 tmp table을 만들어서 date에 대한 값을 string으로 저장합니다.

```
hive> create table tmp (
    > RENT_STATION int,
    > RENT_DATE string,
    > RETURN_STATION int,
    > RETURN_DATE string) row format delimited fields terminated by ',';
OK
Time taken: 0.082 seconds
```

2) tmp table 에 tashu.csv 데이터를 load 합니다.

```
hive> load data local inpath '/home/datascience/tashu.csv' overwrite into table
tmp;
Loading data to table default.tmp
Table default.tmp stats: [numFiles=1, numRows=0, totalSize=126400747, rawDataSiz
e=0]
OK
Time taken: 7.37 seconds
```

3) date format을 timestamp로 한 tashu table을 생성합니다.

```
hive> create table tashu(
    > RENT_STATION int,
    > RENT_DATE timestamp,
    > RETURN_STATION int,
    > RETURN_DATE timestamp)
    > row format delimited
    > fields terminated by ','
    > stored as orc;
OK
Time taken: 0.081 seconds
```

4) tashu table에 tmp table의 data를 insert 하는데, date format 을 conversion 합니다.

```
hive> insert into table tashu
    > select RENT_STATION,
    > from_unixtime(unix_timestamp(RENT_DATE, 'yyyyMMddHHmmss')),
    > RETURN_STATION,
    > from_unixtime(unix_timestamp(RETURN_DATE, 'yyyyMMddHHmmss'))
    > from tmp;
```

```
Table default.tashu stats: [numFiles=1, numRows=3404664, totalSize=157063263, ra
wDataSize=153658599]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 54.81 sec   HDFS Read: 126405010 HDFS Wr
ite: 157063344 SUCCESS
Total MapReduce CPU Time Spent: 54 seconds 810 msec
OK
Time taken: 86.332 seconds
```

5) select 명령어를 통해 data가 잘 들어 갔는지 확인합니다.

```
hive> select * from tashu limit 10;
OK
NULL    NULL    NULL    NULL
43      2013-01-01 05:56:03     34      2013-01-01 06:02:17
97      2013-01-01 06:04:00     NULL    2013-01-01 10:20:37
2       2013-01-01 06:04:06     10      2013-01-01 06:18:59
106     2013-01-01 10:53:05     105     2013-01-01 10:57:43
4       2013-01-01 11:22:23     4       2013-01-01 12:17:53
21      2013-01-01 11:39:53     105     2013-01-01 11:49:43
90      2013-01-01 12:08:33     91      2013-01-01 12:51:36
13      2013-01-01 13:14:29     30      2013-01-01 13:30:39
1       2013-01-01 13:37:42     1       2013-01-01 13:38:15
Time taken: 0.117 seconds, Fetched: 10 row(s)
```

A. 연도별 대여량 (Rent station)

```
hive> select year(RENT_DATE), count(*) as cnt from tashu group by year(RENT_DATE
);
Query ID = datascience_20170601204847_d4975fce-3afc-477e-8fae-5cf587e0de77
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1496316396394_0006, Tracking URL = http://slave1:8088/proxy/a
pplication_1496316396394_0006/
```

```
Total MapReduce CPU Time Spent: 6 seconds 550 msec
OK
NULL    1
2013    1036614
2014    1200187
2015    1167862
Time taken: 39.067 seconds, Fetched: 4 row(s)
```

B. 월별 대여량 (Rent station)

```
hive> select month(RENT_DATE), count(*) as cnt from tashu group by month(RENT_DA
TE);
Query ID = datascience_20170601215908_cb5a5ea0-87bc-4009-b2bb-920f4a3dd2b6
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
```

```
Total MapReduce CPU Time Spent: 9 seconds 90 msec
OK
NULL    1
1       99693
2       116987
3       232712
4       290519
5       414934
6       480429
7       358507
8       320058
9       366859
10      360240
11      214771
12      148954
```

C. 일별 대여량 (Rent station)

```
hive> select day(RENT_DATE), count(*) as cnt from tashu group by day(RENT_DATE);

Query ID = datascience_20170601204940_c6e3621c-6103-4cb1-815b-6718908c65b2
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
```

```
Total MapReduce CPU Time Spent: 7 seconds 380 msec
OK
NULL    1
1       116298
2       104499
3       106474
4       115649
5       115198
6       110271
7       105940
8       107095
9       117166
10      110197
11      105053
12      104176
13      107960
14      110600
15      120347
16      120099
17      114330
18      105077
19      112020
20      106850
21      113472
22      120796
23      111161
24      110613
25      104908
26      120407
27      106867
28      119053
29      102871
30      114542
31      64674
Time taken: 38.47 seconds, Fetched: 32 row(s)
```

D. 시간대별 대여량 (Rent station)

```
hive> select hour(RENT_DATE), count(*) as cnt from tashu group by hour(RENT_DATE
);
Query ID = datascience_20170601205045_d99787e9-b448-4fef-ac5f-cf74c348bf87
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1496316396394_0008, Tracking URL = http://slave1:8088/proxy/a
pplication_1496316396394_0008/
```

```
Total MapReduce CPU Time Spent: 7 seconds 300 msec
OK
NULL    1
0       63022
1       15205
2       199
3       3
4       11
5       16591
6       22152
7       129110
8       186421
9       142126
10      111652
11      96250
12      118768
13      157488
14      173437
15      167018
16      194413
17      250842
18      292905
19      241023
20      259816
21      273561
22      251385
23      241265
Time taken: 37.311 seconds, Fetched: 25 row(s)
```

추가적으로 Top 10 rent station과 Top 10 trace 를 구해보았습니다.

E. Top 10 Rent station

```
hive>
    > select RENT_STATION, count(*) as cnt from tashu group by RENT_STATION orde
r by cnt desc limit 10;
```

```
Total MapReduce CPU Time Spent: 9 seconds 530 msec
OK
3       174801
56      91111
31      83551
17      82973
32      73681
33      71191
14      57505
21      56384
105     56306
55      55200
Time taken: 61.21 seconds, Fetched: 10 row(s)
```

F. Top 10 trace

```
hive> select RENT_STATION, RETURN_STATION, count(*) as cnt from tashu group by R
ENT_STATION, RETURN_STATION order by cnt desc limit 10;
```

```
Total MapReduce CPU Time Spent: 13 seconds 340 msec
OK
3       3       84496
31      31      21749
56      56      18343
21      105     17220
1       1       14489
32      32      12177
105     21      12154
33      33      11973
17      17      11966
56      32      11868
Time taken: 455.792 seconds, Fetched: 10 row(s)
```