

# 빅 데이터 처리 시스템 개발\_최현진

2021-06-25

### 3.Hdfs에 저장된 keyword 검색어 파일을 키워드를 집계하는 mapReduce 프로그램을 개발하시오

```
[root@Server101 ~]# start-all.sh
```

In operation

Show  entries

Search:

Node	Http Address	Last contact	Last Block Report	Capacity	Blocks	Block pool used	Version
✓ Server101:50010 (192.168.100.101:50010)	<a href="http://Server101:50075">http://Server101:50075</a>	0s	1m	16.99 GB <div><div></div></div>	20761	661.67 MB (3.8%)	2.10.1
✓ Server102:50010 (192.168.100.102:50010)	<a href="http://Server102:50075">http://Server102:50075</a>	2s	1m	16.99 GB <div><div></div></div>	20737	661.18 MB (3.8%)	2.10.1
✓ Server103:50010 (192.168.100.103:50010)	<a href="http://Server103:50075">http://Server103:50075</a>	1s	1m	16.99 GB <div><div></div></div>	20737	661.18 MB (3.8%)	2.10.1

Showing 1 to 3 of 3 entries

Previous **1** Next

```
[root@Server101 ~]# spark-shell
```

```
scala> var keywordRDD = sc.textFile("/bigdata/keyword/2021-06-20/*")
keywordRDD: org.apache.spark.rdd.RDD[String] = /bigdata/keyword/2021-06-20/* MapPartitionsRDD[1] at textFile at <console>:24
```

```
scala> var wordRDD = keywordRDD.map(_.split(",")(1))
wordRDD: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[2] at map at <console>:25
```

```
scala> var wordRDD = keywordRDD.map(_.split(",")(1))
wordRDD: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[2] at map at <console>:25
```

### 3.Hdfs에 저장된 keyword 검색어 파일을 키워드를 집계하는 mapReduce 프로그램을 개발하시오

```
scala> var kvRDD = wordRDD.map((_, 1))
kvRDD: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[3] at map at <console>:25
scala> var result = kvRDD.reduceByKey(_+_).sortBy(_. _2, false)
result: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[9] at sortBy at <console>:25
scala> print(result.collect.mkString("\n"))
scala> print(result.collect.mkString("\n"))
(알 고 있 지 만 ,1436)
(그 것 이 알 고 싶 다 ,1434)
(오 케 이 광 자 매 ,1429)
(장 성 철 ,1425)
(임 영 웅 ,1423)
(아 이 유 ,1418)
(비 트 코 인 ,1394)
(나 얼 ,1327)
(db,960)
(정 찬 성 ,898)
(슬 기 로 운 의 사 생 활 시 즌 2,490)
(유 로 2020,381)
(보 이 스 4,245)
(한 예 슬 ,113)
(보 씬 운 명 을 흠 치 다 ,25)
(마 인 ,2)
scala> 
scala> result.saveAsTextFile("/sample/keyword_result")
```

4.Hive를 설치하고 keyword 검색어 파일의 키워드를 집계하는 쿼리를 수행하시오.(설치과정)

```
[root@Practicel48 ~]# wget https://mirror.navercorp.com/apache/hive/
```

```
[root@Practicel48 ~]# vi ~/.bashrc
```

```
5 alias rm='rm -i'
6 alias cp='cp -i'
7 alias mv='mv -i'
8
9 # Source global definitions
10 if [ -f /etc/bashrc ]; then
11     . /etc/bashrc
12 fi
13 export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
14 export PATH=$PATH:$JAVA_HOME/bin
15 export HADOOP_HOME=/home/bigdata/hadoop
16 export HADOOP_MAPRED_HOME=$HADOOP_HOME
17 export HADOOP_COMMON_HOME=$HADOOP_HOME
18 export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
19 export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
20 export YARN_HOME=$HADOOP_HOME
21 export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
22 export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
23 export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
24 export CLASS_PATH=$JAVA_HOME/lib:$CLASS_PATH
25 export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
26 export HIVE_HOME=/home/bigdata/hive
27 export PATH=$PATH:$HIVE_HOME/bin
```

4.Hive를 설치하고 keyword 검색어 파일의 키워드를 집계하는 쿼리를 수행하시오.(설치과정)

```
[root@Practice148 ~]# source ~/.bashrc
```

```
[root@Server101 ~]# vi /home/bigdata/hive/conf/hive-env.sh
```

```
36 # fi
37
38 # The heap size of the jvm started by hive shell script can be controlled via
39 #
40 # export HADOOP_HEAPSIZE=1024
41 #
42 # Larger heap size may be required when running queries over large number of
43 # files or partitions.
44 # By default hive shell scripts use a heap size of 256 (MB). Larger heap si
45 ze would also be
46 # appropriate for hive server.
47
48 # Set HADOOP_HOME to point to a specific hadoop install directory
49 HADOOP_HOME=/home/bigdata/hadoop
50
51 # Hive Configuration Directory can be controlled by:
52 # export HIVE_CONF_DIR=
53
54 # Folder containing extra libraries required for hive compilation/execution
55 can be controlled by:
56 # export HIVE_AUX_JARS_PATH=
57
58 "/home/bigdata/hive/conf/hive-env.sh" 54L, 2365C 48,33 바 닷
```

4.Hive를 설치하고 keyword 검색어 파일의 키워드를 집계하는 쿼리를 수행하시오.(설치과정)

```
1 <configuration>
2   <property>
3     <name>hive.metastore.warehouse.dir</name>
4     <value>/hive/warehouse</value>
5   </property>
6   <property>
7     <name>javax.jdo.option.ConnectionURL</name>
8     <value>jdbc:mysql://localhost:3306/hive_metastore_db?
9         createDatabaseIfNotExist=true</value>
10  </property>
11  <property>
12    <name>javax.jdo.option.ConnectionDriverName</name>
13    <value>com.mysql.jdbc.Driver</value>
14  </property>
15  <property>
16    <name>javax.jdo.option.ConnectionUserName</name>
17    <value>hive</value>
18  </property>
19  <property>
20    <name>javax.jdo.option.ConnectionPassword</name>
21    <value>1234</value>
22  </property>
23 </configuration>
```

```
[root@Server101 ~]# start-all.sh
```

```
[root@Server101 ~]# hive
```

4.Hive를 설치하고 keyword 검색어 파일의 키워드를 집계하는 쿼리를 수행하시오.(쿼리 수행)

```
hive> CREATE EXTERNAL TABLE `test3` (  
  > `rank` Int,  
  > `keyword` String,  
  > `rdate` String  
  > )  
  > row format delimited  
  > fields terminated by ","  
  > location "/bigdata/2021-06-20/"  
  > tblproperties("skip.header.line.count"="1");  
OK  
Time taken: 0.463 seconds
```