

Lab-4

Q-1 Mutual friend finding

We are using map reduce over here to find the common friend of any two friends. Below is the step by step process to achieve this.

Let say we stored friends as Person->[List of Friends]:

A -> B C D
B -> A C D E
C -> A B D E
D -> A B C E
E -> B C D

Here we will take key as a friend along with a person and value will be list of friend. The all mapper will be look like below, let say for A it will be:

For map(A -> B C D) :

(A B) -> B C D
(A C) -> B C D
(A D) -> B C D

The same way it will be look for all rows. We need to group them as per keys before sending them to reducer.

(A B) -> (A C D E) (B C D)
(A C) -> (A B D E) (B C D)
(A D) -> (A B C E) (B C D)
(B C) -> (A B D E) (A C D E)

Now we can pass this to reducer and reducer will take these lists as sets and will find out common values from it and the result will be as follow.

(A B) -> (C D)
(A C) -> (B D)
(A D) -> (B C)
(B C) -> (A D E)
(B D) -> (A C E)
(B E) -> (C D)
(C D) -> (A B E)
(C E) -> (B D)
(D E) -> (B C)

Below are the file for algorithm and output commands executed on cloudera.



Cloudera_Job_Exec MutualFriends.java
ution.txt



Input

```
[cloudera@quickstart ~]$ cat>input.txt
A B C D
B A C D E
C A B D E
D A B C E
E B C D
^C
[cloudera@quickstart ~]$ cat input.txt
A B C D
B A C D E
C A B D E
D A B C E
E B C D
```

Output

```
[cloudera@quickstart workspace]$ hadoop jar Mutualfriend.jar MutualFriends hiren/input.txt friendoutput
16/02/17 09:12:14 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
16/02/17 09:12:14 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interf
16/02/17 09:12:15 INFO input.FileInputFormat: Total input paths to process : 1
16/02/17 09:12:15 INFO mapreduce.JobSubmitter: number of splits:1
16/02/17 09:12:15 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1455727015015_00
16/02/17 09:12:15 INFO mapreduce.Job: Running job: job_1455727015015_0002
16/02/17 09:12:26 INFO mapreduce.Job: Job job_1455727015015_0002 running in uber mode : false
16/02/17 09:12:26 INFO mapreduce.Job: map 0% reduce 0%
16/02/17 09:12:33 INFO mapreduce.Job: map 100% reduce 0%
16/02/17 09:12:40 INFO mapreduce.Job: map 100% reduce 100%
16/02/17 09:12:40 INFO mapreduce.Job: Job job_1455727015015_0002 completed successfully
[cloudera@quickstart workspace]$ hadoop fs -ls
Found 4 items
drwxr-xr-x - cloudera cloudera 0 2016-02-17 09:12 friendoutput
drwxr-xr-x - cloudera cloudera 0 2016-02-17 09:07 hiren
drwxr-xr-x - cloudera cloudera 0 2016-02-13 19:44 ram
drwxr-xr-x - cloudera cloudera 0 2016-02-14 00:39 wordcountoutput
[cloudera@quickstart workspace]$ hadoop fs -cat friendoutput
cat: 'friendoutput': Is a directory
[cloudera@quickstart workspace]$ hadoop fs cd friendoutput
cd: Unknown command
[cloudera@quickstart workspace]$ hadoop fs -ls friendoutput
Found 2 items
-rw-r--r-- 1 cloudera cloudera 0 2016-02-17 09:12 friendoutput/_SUCCESS
-rw-r--r-- 1 cloudera cloudera 57 2016-02-17 09:12 friendoutput/part-r-00000
[cloudera@quickstart workspace]$ ^C
[cloudera@quickstart workspace]$ hadoop fs -cat friendoutput/part-r-00000
AB CD
AC BD
AD BC
BC ADE
BD ACE
BE CD
CD ABE
CE BD
CE BD
DE BC
```

Q-2 Smartphone Applicationspeech services.

I have used the face++ services for face detection of this application. Below is the snap shot for this.

User can select the image by clicking on Browse image and then it will locate the face. Source is available on Github.

