

## 设计思路

仍然是两个Map和Reduce

自定义数据类型为Friends，其数据成员有person1/person2/commfriend，分别是两个不同用户以及他们的共同好友之一

假设输入为：

```
100, 200 300 400
200, 100 300 400
300, 100
400, 100
```

第一个Map及Reduce用于输出具有共同好友的用户对及其共同好友，其输出为：

```
[200,300]:100
[200,400]:100
[300,400]:100
[100,200]:300
[100,200]:400
```

第一个Map用于统计“被好友”的主体及对象，以输入的第一行为例，第一个Map产生的[key,value]有：  
[200,100] [300,100] [400,100]，可得到所有将某个key当做好友的人；再到Reduce中将这些人两两组合成用户对作为key，对应的value是二者的某一共同好友，自定义数据类型Friends就是用于这个阶段。

第二个Map及Reduce用于整合同一好友对下的共同好友们，其输出为：

```
[100,200]:[300,400]
[200,300]:100
[200,400]:100
[300,400]:100
```

## 代码结果

```
sheepxi@ubuntu:~/FBDP_2020Fall_171840013/cf_selfdefine/CommFriends$ hdfs dfs -cat output2/part-r-00000
2020-11-03 10:21:51,004 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2020-11-03 10:21:52,332 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
[100,200], [300,400]
[100,300], [200,400,500]
[100,400], [200,300]
[100,500], [300]
[200,300], [400,100]
[200,400], [300,100]
[200,500], [300,100]
[200,600], [100]
[300,400], [200,100]
[300,500], [100]
[300,600], [100]
[400,500], [100,300]
[400,600], [100]
[500,600], [100]
```

## 运行截图

```
2020-11-03 10:21:36,301 INFO mapred.LocalJobRunner: Finishing task: attempt_local714321574_0002_r_000000_0
2020-11-03 10:21:36,304 INFO mapred.LocalJobRunner: reduce task executor complete.
[job name:      all users
job id: myCtrl0
job state:      SUCCESS
job mapred id:  job local598873233_0001
job message:    just initialized
job has no depending job:
, job name:     common friends
job id: myCtrl1
job state:      SUCCESS
job mapred id:  job local714321574_0002
job message:    just initialized
job has 1 depending jobs:
    depending job 0:      all users
]
sheepxi@ubuntu:~/FBDP_2020Fall_171840013/cf_selfdefine/CommFriends$ hdfs dfs -cat output2/part-r-00000
```