The Big Data Analysis of Two-degree Contacts on Hadoop

Hao Wu
Department of Computer Science, JLUZH
Zhuhai College of Jilin University
Zhuhai, China
haowu_mouse@hotmail.com

Xiongfei Li
Department of Computer Science, JLU
Jilin University
Changchun, China
lxf@jlu.edu.cn

***Abstract*—The social network has undergone a rapid development along of the popularity of the Internet. In face of the huge social data increasingly, the common processing methods or algorithm is difficult to deal with it. With its technology maturing gradually, Hadoop has been popularized and used to the different categories. For its characteristics of the lower cost and the higher efficiency to deal with the huge amounts of data, Hadoop is able to analyze two-degree contacts quickly, so as to obtain the information of contacts faster and to build better contacts. It designs the configuration on Hadoop, one is the master host, and the others are slavers. Through the configuration in the network, these parts are able to communicate with each other and ensure that the master host can control the slaves. This system adopts mainly the core frameworks which are the Hadoop distributed file system (HDFS) and the MapReduce algorithm. According to the statistical support and combining with the theory of two-degree contacts and Hadoop, it shows the degree of recommending contacts and realizes the analysis of two-degree contacts on Hadoop finally. Through the big data analysis technology on Hadoop and the process optimized of two-degree contacts, the efficiency of the system can be improved significantly.**

Keywords—MapReduce; two-degree contacts; Hadoop

I. INTRODUCTION

With the network popularization, more and more people go into it, so it sets off a new age of Internet. At the same time the social network has also experienced the rapid development, from BBS to chatting rooms and from the space or the blog to twitter or wechat. Accounted for a large proportion on Internet, the social network is not only to provide a convenient and entertainment communication platform for people, but also to promote the development of relevant industries. People are participants in the social network. The interpersonal relation is an important part of the social contacts. A popular and active social platform is bound to build the interpersonal relationships network. Therefore, such as Tencent and twitter, the social platform has a function of the friend recommendation.

Many users appreciate Hadoop for its advantages of the open source code, the easier secondary amendment, the lower cost and more efficient process the massive data. In a social network, the more users register, the more complex interpersonal relations. Whether it can recommend friends to users quickly and accurately, and expand users' social relationship, that is a crucial point of reserving the users. Hadoop has an outstanding performance in analyzing the big data. The analysis of two-degree contacts on Hadoop can improve the efficient that of user' social relationships.

Hadoop configuration is deployed by three parts, one is the master host and the others are slaver sub-systems. These three parts are able to communicate with each other through the network, and ensure that the master host can control the sub-systems. This system adopts mainly two core frameworks, one is the Hadoop distributed file system (HDFS), other is MapReduce algorithm. Combining with the theory of two-degree contacts and the Hadoop technology, through the statistical support, it shows the degree of the recommendable connection and realizes the analysis of two-degree contacts on Hadoop finally. Through the big data analysis technology on Hadoop and the superior process of two-degree contacts, it can improve the efficiency of the system significantly.

II. MAIN METHODOLOGY

A. Hadoop

Hadoop is a distributed computing framework, which can run programs on the cluster composed of a large amount of cheap hardware and provide a set of stable and reliable interface for them. That is as shown in Fig1.

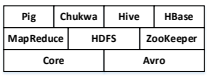

Fig. 1. Hadoop conformations

HDFS and MapReduce are the most important members on Hadoop, which provide a complementary or higher level of service on the core layer. HDFS is a master/slave structure, which runs only a NameNode on the master and a DataNode on each slave sub-system. MapReduce is a program model and used to calculate a large amount of data, which core operation is Map and Reduce. Map is to map a set of data to another group of it one by one. Reduce is to reduce a set of data.

The purpose of the computing framework on Hadoop is to build a distributed operating system with a high reliability and a good expansibility. As the cloud computing being popular increasingly, this technology is applied to individuals and enterprises more and more.

B. *MapReduce*

MapReduce is a computing architecture. It uses the parallel computation to deal with terabytes or petabytes of data. Its framework consists of the slave TaskTracker on cluster nodes of a single master JobTracker. The master is responsible for scheduling all the tasks in a job, and to distribute these tasks on different sub-systems. Slave nodes monitor the implementation of these tasks, and re-execute failure tasks. Thus, the slave is responsible for the implementation of the tasks assigned by the master.

Every stage of MapReduce is in from of <key,value> to transmit data. WordCount is one of the most classic program instances in Hadoop, which is to count the number of the occurrence of each word in an English article. During Map it is a series of <key,value>, such as <Red,1>, <Yellow,1>, <Red,1>, <Yellow,1>, <Yellow,1>,<Blue,1> etc. During Reduce it is to classify and to count the phase of <key,value>, and to output the final result, such as<Red,2>, <Yellow,3>, <Blue,1> etc.

By the way of the instance of assorting the triangle (△), the circle (○) and the square (□) in graphs classification, it explains the operating principle of MapReduce. That is as shown in Fig 2.

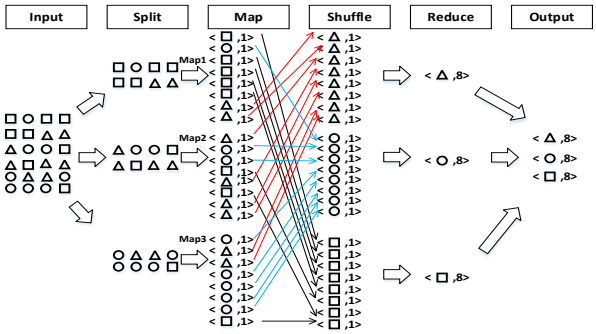

Fig. 2. MapReduce operating principle

The first step is Input that is to put various kinds of graphics in the graph set to upload to HDFS.

The second step is Split that NameNode receive the state information of DataNode to know which DataNode is idle, and then JobTracker to distribute tasks to these idle TaskTrackers. It shows that various kinds of graphic are collected and assigned to these three "workers" of Map1, Map2, Map3, and let them assort at the same time.

According to the rules of the <key,value>, the third step is Map is that Map1, Map2 and Map3 record the number of each type of graphics initially, such as that Map1 record is <□,1>, <○,1>, <□,1>, <□,1>, <□,1>, <□,1>, <△,1>, <△,1>.

The forth step is Shuffle is that before Reduce, three "workers" deal with these three graphic records preliminarily, and count the records of the triangle, the circle and the square respectively, in order to improve the efficiency of the graph statistics.

The fifth step is Reduce is that the "workers" report work results to JobTracker, then JobTracker notifies "statistician". After receiving records, "statistician" counts on each record and calculates the statistical result of <△,8>, <○,8>, <□,8>.

The sixth step is Output is that "statistician" makes records and submits work results. At this moment, the task of sorting the triangle (△), the circle (○) and the square (□) is completed.

III. ANALYSIS AND DESIGN

A. *Designing contacts structure*

The relation is to depict the contact between objects. The relationship among people is marvelous, because it seems compact, yet aloof, however, sometimes it is converse. The different relationship or identity, such as relatives, friends, teachers, students, alumni, colleagues, often hides some mysterious and interesting information.

In order to research and analysis two-degree contacts on Hadoop, it designs a simple relationship. That is as shown in Fig3.

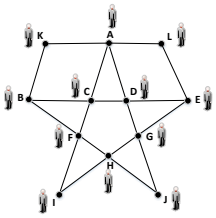

Fig. 3. The contact structure

The shape of this contact structure is like a star, which is consist of twelve people by the letters from A to L, and each solid dot is corresponds to the position of that in the network. That there is a black solid line between two solid dots represents the two people knowing each other.

1) *One-degree contacts*

In accordance with the order of the letter, it enumerates one-degree contacts of each one, which indicates that they pay close attention on each other in the way of <self,one-degree contacts>. Part of results is as following:

- NO. A:<A,C>, <A,D>, <A,K>, <A,L>
- NO. C: <C,A>, <C,B>, <C,D>, <C,F>

2) *Two-degree contacts*

In accordance with the order of the letter, it enumerates two-degree contacts of each one, which indicates that they know each other through introduced by the mutual friends in the way of<self,two-degree contacts>, excluding knowing each other directly. Part of results is as following:

- NO. A: <A,B>, <A,E>, <A,F>, <A,G>
- NO. C: <C,E>, <C,G>, <C,H>, <C,I>, <C,K >, <C,L>

3) *The support of two-degree contacts*

Because two-degree contacts can know each other through being introduced by the different one-degree ones, it is to take different routes in the contacts network. According to the number of routes in two-degree contacts, it externalizes that the

more support, the greater likelihood will become two-degree contacts, so it is worthy of being paid attention.

In accordance with the letter order, it enumerates the routes of the one-degree and two-degree contacts in the way of <self, one-degree friend, two-degree friend>. Part of results is as following:

- NO. A <A,B>: <A,C,B>, <A,K,B> → Support: 2
- NO. B <B,A>: <B,C,A>, <B,K,A> → Support: 2
- NO. C <C,K >: <C,A,K >, <C,B,K > → Support: 2

  <C,L>: <C,A,L> → Support: 1

It indicates that A and B are two-degree contacts, and the support is 2. That is to say that the route is the same, but only exchanges the starting and the end point. By computing the routes, <self,two-degree contacts,Support> can be expressed as <A,B,2> or <B,A,2>, which is to select one merely.

According to the alternative criteria of the encoding sequence, it need to select <A,B,2> from <A,B,2> to <B,A,2>. The part of results of the analysis process is as follows.

- NO. A: <A,B,2>, <A,E,2>, <A,F,1>, <A,G,1>
- NO. C: <C,E,1>, <C,G,1>, <C,H,1>, <C,I,1>, <C,K,2>, <C,L,1>

B. Data pre-processing

Getting one-degree contacts in the course of analyzing the contacts and the selection criteria of the encoding sequence, it is further to submit the data to HDFS, such as(A,C), (A,D), (A,K), (A,L), (B,C), (B,F), (B,K), (C,D), (C,F), (D,E), (D,G), (E,G), (E,L), (F,H), (F,I), (G,H), (G,J), (H,I), (H,J).

(A,C) means that A and C are one-degree contacts. It writes the data into the file of Friend-in.txt. That is as shown in TABLE I.

TABLE I. ONE-DEGREE CONTACTS

NO.	*One-degree contacts*	
1	A	C

In the light of the operating principle of MapReduce, it submits the file of Friend-in.txt to HDFS.

C. Performing the 1st time of MapReduce

1) Performing Mapper1

Mapper1 is a mapping method on the 1st time of MapReduce. The contacts information is separated by ",", the key named as their names, and the value considered as the two individual relationships. And then, it gets the <key,value>, which there is a tab between key and value. For example, it inputs "A,C". After passing Mapper1, it will produce two results which are key=A, value="A C" and key=C, value="A C". That is as shown in TABLE II.

TABLE II. MAPPER1 RESULTS

NO.	*map1_key*	*map1_value*	
1	A	A	C
2	C	A	C

Note: key=A, value=(A C)

2) Performing Reducer1

Reduce1 is a simplified method on the 1st time of MapReduce. Mapper1 submits results to Reducer1, which has two goals. The 1st goal is that it regards map1_value that stores the mutual concern of contacts as the key, and marks one-degree contacts as "friend1". For example, in Mapper1 it receives the input that is map1_key=A is as shown in TABLE III.

TABLE III. THE INPUT OF ONE-DEGREE CONTACTS (MAP1_KEY=A)

NO.	*reducer1_key*	*reducer1_value*	
1	A	A	C

After Reducer1, the relationship between the key and "friend1" is marked as the value, which output is as shown in TABLE IV.

TABLE IV. THE OUTPUT OF TWO-DEGREE CONTACTS(MAP1_KEY=A)

NO.	*reducer1_key*		*reducer1_value*
1	A	C	friend1

The 2nd goal is based on the map1_key as a collection, which considers one-degree contacts as the elements in that. Through combining, it can construct a new contacts relationship, which is identified preliminary with two-degree contacts marked as "friend2". For example, map1_key=A. <key, value> passed by Mapper1, it can know that the letter of A which one-degree contacts is C, D, K and L. It can get new contacts that are <C,D>,<C,K>,<C,L>,<D,K>,<D,L> and <K,L>, considered as key and marked "friend2" as value.

Then outputting results in Reducer1 are as shown in TABLE V.

TABLE V. THE OUTPUT OF TWO-DEGREE CONTACTS(MAP1_KEY=A)

NO.	*reducer1_key*		*reducer1_value*
1	C	D	friend2

D. Performing the 2nd time of MapReduce

1) Performing Mapper2

Mapper2 is a mapping method on the 2nd time of MapReduce, which main effect is to link. The results of Reducer1 are considered as a new text, and that is as MapReduce input on the 2nd time. Because the two results are <val, reduce1_value> and <reduce1_key, reduce1_value>, it can know that each line is divided with "\t" after being read by Reducer1, and can get three fields, such as following.

"A C friend1" can be divided into "A", "C" and "friend1". "C D friend2" can be divided into "C", "D" and "friend2".

The final contact is regard as map2_key and labeled as map2_value.That is as shown in TABLE VI.

TABLE VI. THE RESULTS OF MAPPER2

NO.	*map2_key*		*map2_value*
1	A	C	friend1
2	C	D	friend2

2) Performing Reducer2

Reducer2 is a simplified method on the 2nd time of MapReduce. It can know that the 1st goal of Reducer1 is just to put one-degree contacts into new combinations, defined as the two-degree contacts. It is not being considered as the true two-degree contacts, but also as the original one-degree ones, that cannot be defined as the two-degree contacts. The role of Reducer2 is to rule out that possibility, to point out the two-degree contacts explicitly and to do the statistics of the support.

In the light of map2_key it can put map2_value into a set by a loop, and then fetch the elements from the set to compare them one by one. If it is "friend1", then isFriend1 becomes true, else false. If it is "friend2", then isFriend2 becomes true, else false. The ultimate judgment is that if both are two-degree contacts, but not one-degree contacts, then being accepted and being recommended.

IV. RUNNING AND RESULTS

A. Setting the running path

In the file of Friend-in, Friend-in.txt is as the input file of MapReduce at the 1st time, and output of that at the 1st time is considered as the input route of MapReduce at the 2nd time.

```
FileInputFormat.addInputPath(job2, new Path(otherArgs[1]));
FileOutputFormat.setOutputPath(job2, new Path(otherArgs[2]));
```

B. Inspecting the running status

It is the status information on the execution of MapReduce at the 1st time and the 2nd time.

```
File System Counters
        FILE: Number of bytes read=722
        FILE: Number of bytes written=376270
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=198
        HDFS: Number of bytes written=1036
        HDFS: Number of read operations=15
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=4
Map-Reduce Framework
        Map input records=19
        Map output records=38
        Map output bytes=246
        Map output materialized bytes=328
        Input split bytes=125
        Combine input records=0
        Combine output records=0
        Reduce input groups=12
        Reduce shuffle bytes=0
        Reduce input records=38
        Reduce output records=84
        Spilled Records=76
        Shuffled Maps =0
        Failed Shuffles=0
        Merged Map outputs=0
        GC time elapsed (ms)=0
        CPU time spent (ms)=0
        Physical memory (bytes) snapshot=0
        Virtual memory (bytes) snapshot=0
        Total committed heap usage (bytes)=466616320
File Input Format Counters
        Bytes Read=99
File Output Format Counters
        Bytes Written=1036
```

```
File System Counters
        FILE: Number of bytes read=2680
        FILE: Number of bytes written=751430
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=2270
        HDFS: Number of bytes written=2246
        HDFS: Number of read operations=43
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=16
Map-Reduce Framework
        Map input records=84
        Map output records=84
        Map output bytes=1036
        Map output materialized bytes=1210
        Input split bytes=136
        Combine input records=0
        Combine output records=0
        Reduce input groups=46
        Reduce shuffle bytes=0
        Reduce input records=84
        Reduce output records=27
        Spilled Records=168
        Shuffled Maps =0
        Failed Shuffles=0
        Merged Map outputs=0
        GC time elapsed (ms)=0
        CPU time spent (ms)=0
        Physical memory (bytes) snapshot=0
        Virtual memory (bytes) snapshot=0
        Total committed heap usage (bytes)=677380096
File Input Format Counters
        Bytes Read=1036
File Output Format Counters
        Bytes Written=174
```

Fig. 4. The executional status of MapReduce at the 1st time and the 2nd time

C. Inspecting running results

After executing this project, in the file of the Friend-out in HDFS directory it adds two folders of MapReduce1 and MapReduce2, which are output paths of MapReduce at the 1st and the 2nd, consistent with the default path. The part result of MapReduce at the 1st time is as shown in TABLE VII.

TABLE VII. THE OUTPUT OF MAPREDUCE1

NO.	*MapReduce1_key*		*MapReduce1_value*
1	A	D	friend1
2	A	C	friend1
3	A	K	friend1
4	A	L	friend1

The part result of MapReduce at the 2nd time as shown in TABLE VIII .

TABLE VIII. THE OUTPUT OF MAPREDUCE2

NO.	*MapReduce2_key*		*MapReduce2_value*
1	A	B	2
2	K	L	1
3	K	C	2
4	L	C	1

D. Analysing running results

Observing the output of MapReduce at the 2nd time and comparing with the analysis results of the above two-degree contacts, it finds that the results on the artificial analysis and on Hadoop are both 27 records. Because Hadoop takes records on the dictionary sort order, the collation is formed in the alphabetical order or the digital size from small to large. For example, according to the dictionary sort order, {K,V,L,B,A,Z} becomes {A,B,K,L,Z,V}. Therefore, the records in the two results will change as TABLE IX.

TABLE IX. THE OUTPUT OF MAPREDUCE2 ON THE DICTIONARY SORT ORDER

No.	*MapReduce2 _key*		*MapReduce2 _value*	⇨	*MapReduce2 _key*		*MapReduce2 _value*
1	C	K	2		K	C	2
2	C	L	1		L	C	1

The record of "K L 1" is added to the above records on the dictionary sort order. The results get as TABLE X.

TABLE X. THE NEW OUTPUT OF MAPREDUCE2

NO.	*MapReduce2_key*		*MapReduce2_value*
5	K	L	1
6	K	C	2

The result is just accord with that of MapReduce output at the 2nd time, which proves that the data analysis of two-degree contacts on Hadoop is successful.

V. SYSTEM TESTING

A. Functional testing

This testing adopts the idea of the black box testing and uses the method of the equivalence class partition. The part of the testing case is as shown in TABLE XI.

TABLE XI. THE FUNCTIONAL TESTING CASES ON THE ANALYSIS OF TWO-DEGREE CONTACTS

Basic steps	1. Input data. 2. Running the program.		
Testing objects	***Testing data***	***Results expected***	***Testing results***
Data formats of the contacts	Normal input: "A,B"	A successful running	A successful running
	Contain punctuations, including commas in English: "Re,d, Blue"	Run successfully, but analysis results are error.	Run successfully, but analysis results are error.
	Contain English, Chinese and punctuations, excluding the commas in English: " 红 色 Re*d,蓝色 Blue"	Getting analysis results	A successful running

B. Analysis testing results

After testing, it proves that this program can run efficiently. That importing the text of ".txt" should appear no blanks, no space, no tabs and no blank line, otherwise, the program is running failure or gets incorrect operating results. The reason for the above problems is that there is no treatment for a variety of data combination and the judgment, so the algorithm should be designed more perfectly.

VI. CONCLUSION

This design adopts a more popular technology of Hadoop. After building a Hadoop platform, it takes the core technology of Hadoop which is HDFS and MapReduce framework. Combined with Linux and Java, it analyses two-degree contacts based on Hadoop. In the new era of Internet, not only on the social platform, but also on BBS, the user is always the core. The cardinal number of users determines the success or failure of the social networking platform. An outstanding function on the friend recommendation can promote the activity of users, which can make the social networking platform more popular.

This study is a good start, not only can bring the innovation about the big data analyses on Hadoop, but also may bring the innovation in the other areas and promote their developments. But there are some disadvantages in the program, such as, if the friends name contains blanks, English commas, tabs and so on, that can lead to executing the program abnormally or analyzing results unfaithfully. According to the format of the specified records, the data on the relationship network should be uploaded to HDFS, otherwise, it will lead to errors. As a result, the algorithm should be designed more delicately and precisely. In addition, this design of contacts is default mutual ones. It can extend this way technically, to increase the directed pattern that A pays attention to B, and C does it to B. Even it can extend two-degree contacts to three-degree or four-degree ones and so on, and can show the results of the analysis in the chart.

ACKNOWLEDGMENT

Thank Zhuhai College of Jilin University for providing the Funds on the training project of both the teaching and the research for young teachers. Thank Zhuhai Laboratory of Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education.

REFERENCES

[1] Yu, Le; Zheng, Jian; Shen, Wei Chong; Wu, Bin; Wang, Bai; Qian, Long; Zhang, Bo Ren. "BC-PDM: Data mining, social network analysis and text mining system based on cloud computing", Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012, p 1496-1499.

[2] Huang, Yin, "A scalable system for community discovery in twitter during hurricane sandy", Proceedings of the 14th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2014, 2014, p 893-899.

[3] http://ww.tdpress.com/zyzx/tsscflwj.

[4] http://www.rzchina.net.

[5] Ian H. Witten, Eibe Frank and Mark A. Hall:Data Mining: Practical Machine Learning Tools and Techniques, Third edition (ISBN 978-0-12-374856-0).

[6] Cramer; Jeannie; O'Donnell; Terrence. "Advancing big data and analytics capabilities", Proceedings -IBM Data Management Magazine, n 11, November 22, 2013

[7] Fernndez, Alberto; del Ro, Sara; Lpez, Victoria; Bawakid, Abdullah; del Jesus, Mara J.; Bentez, Jos M.; Herrera, Francisco . "Big Data with Cloud Computing: An insight on the computing environment, MapReduce, and programming frameworks", Proceedings of Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, v 4, n 5, p 380-409, September 1, 2014.

[8] Ayma, V.A. Ferreira, R.S.; Happ, P.; Oliveira, D.; Feitosa, R.; Costa, G.; Plaza, A.; Gamba, P. "Classification algorithms for big data analysis, a map reduce approach", Proceedings of International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, v 40, n 3W2, p 17-21, 2015.

[9] Triguero, Isaac; Peralta, Daniel; Bacardit, Jaume; Garca, Salvador; Herrera, Francisco. "MRPR: A MapReduce solution for prototype reduction in big data classification", Proceedings of Neurocomputing, v 150, n PA, p 331-345, February 20, 2015.

[10] Patel, Warish D. Vaghela, Dineshkumar B. "An efficient improved join algorithm using map reduce in Hadoop", Proceedings of 2014 International Conference on Signal Propagation and Computer Technology, ICSPCT 2014, p 263-272, 2014, 2014 International Conference on Signal Propagation and Computer Technology, ICSPCT 2014.