Advances in Engineering Research (AER), volume 116
International Conference on Communication and Electronic Information Engineering (CEIE 2016)

Implement of network user virtual identity merging based on spark

Gang He and Sha Chen
Beijing Key Laboratory of Network System Architecture and Convergence, Beijing University of Posts and Telecommunications
Email: happystephanie@bupt.edu.cn
www.bupt.ed.cn

With the rapid development of Internet, people show growing dependence on network communications. Each user has a lot of network virtual identities because of the explosive increase of network applications. Compared with previous work, our study proposes an algorithm to find virtual identities which belong to the same natural person. On the other hand, in view of massive data from users' access records, our study is based on a quick, general engine with ease of use for large-scale data processing platform and clustering system called Spark.

Keywords: Network Virtual Identity; Identities Merging; Algorithm; Spark.

1. Introduction

According to the 38th report of China Internet Network Information Center (CNNIC) [1], by the June 2016, China has over 7 million network users. People do almost everything on internet. Registering website is always the first step to network activities, which generates millions and billions network user virtual identities called NUVIs in this paper.

The massive data reaching PB even TB level contains huge information including users' accounts and behaviors, messages. In the previous study, they just do some simple and analysis on NUVIs [7] or users behaviors [8] or not accurate enough [5]. Our research aims to make full use of NUVIs, such as QQ, WeChat, E-mail, forum account etc. Obviously, natural personal has more than one NUVI. Our target is to find correlations between NUVIs and merge them to a whole. We find an efficient method to verify whether two or more NUVIs belong to the same user. So it's convenient to confirm a user's real identity when knowing a virtual identity. Our research will make a significance of network information security.

2. Data Acquisition

In our research, the massive data is captured from the devices called Traffic Monitor located in the network gates of the enterprises. Every PC and routing equipment must through monitoring devices when surfing internet. A great number of devices constitute a Network Monitoring Platform collecting traffic data of network users. Captured data includes key information such as customer_id , mac address, NUVI_type, NUVI_account etc.

Raw data files are in term of individual days. In our study, we take data for half a year(from October 2015 to April 2016) as total resource reaching hundreds of GB. Proved by experiments, our study need six useful fields from a lot of fields. The Table 1 lists six related fields in our study.

Tab. 1 Description of fields

Name	Description
Customer_id	The number of monitoring device
NUVI_Type	The type of network user virtual identity
NUVI	Network user virtual identity
Time	The occurrence time of in reporting period
MAC	Inventory control
IP	IP address of PC

3. Data Processing Platform

Facing such huge data, Spark is a quick, general and ease of use engine for large-scale data processing. Spark revolves around the data model of a resilient distributes dataset(RDD) [10]. RDD provide a rich set of operations to manipulate data, which makes Spark a basic consistent way to deal with different data processing scenarios. RDD's transformation operation generate many new RDDs, the dependencies between these RDDs form a DAG(directed acyclic graph), supporting high fault-tolerance and data-recovering through data caching mechanism. Details about computing on Spark refers to [3][6] .Consequently, we choose Spark rather than Hadoop MapReduce [2] as data processing and computing platform in this research [9].

4. Algorithm Implement

Our algorithm help solve how to verify which NUVIs belong to the same user. Two points should be clear. Firstly, every network devices has a unique MAC address, so Mac address is the key fields to distinguish different NUVIs

belonging to different users. Secondly, every single user has multiple NUVIs, so multiple NUVIs appear on a MAC address of a device during the same time slot.

4.1. *Data preprocessing*

In this part, we mainly describe two points including how data noise generates and how to get rid of data noise to avoid data confusion and result deviation. Inevitably, our Network Monitoring Platform occur runtime errors such as running programs failed or unexpected surprises. Data noise embodies in the following points.

Certain records show with incomplete fields or incorrect data format.

Monitoring platform locates in the network gates of enterprises. There is occasion where a router connected with monitoring devices and many PC connect with the router, but monitoring devices only capture the routers' MAC address.

There is another occasion where some NUVIs will appear on more than a MAC address, which leads to those invalid NUVIs mixed together valid NUVIs.

We have three methods to get rid of data noise.

The first round of data preprocessing aims to uniform data format including extracting six useful fields from dozens of fields and eliminating data records which has incomplete fields or incorrect format. For examples, if NUVI is a QQ account, then its' length have to be 5 to 11.

Our research focuses on how to prove two or more NUVIs belong to the same user, so single NUVIs which never appear together with other NUVIs are not included in our study. After practical tests, single NUVIs occupy about 28.4% of the total NUVIs. The second round is to filter single NUVIs from all NUVIs.

The third round of data preprocessing aims to find out invalid MAC address and NUVIs called error records. We calculate numbers of NUVIs shows at every MAC address in a day at first. Then, we find the inflection point as the threshold N to determine whether a MAC address is invalid. Mac addresses which has more than N NUVIs is invalid MAC. We can filter error records with invalid NUVI from origin data in a similar way.

The fourth issue is about dealing with the time field. We can boldly guess that NUVIs appearing on the same mac and ip at the same time are very likely to belong to the same user. So the time field is formatted to the day.

4.2. *Finding out frequent virtual identities pairs with high relevance*

The above data preprocessing remove lots of error records from raw data roughly. In order to further find out NUVIs with high relevance more accurately,

our study proposed a method to find out frequent NUVIs pairs. We hold that those two final NUVIs thought as frequent NUVIs pairs belong to the same user [11][12]. Figure 2 is a flow chart of data processing. We get a group of NUVIs in the same key(key=customer_id, ip, mac, time) and count the number of every single NUVI and the number of every NUVIs pair.

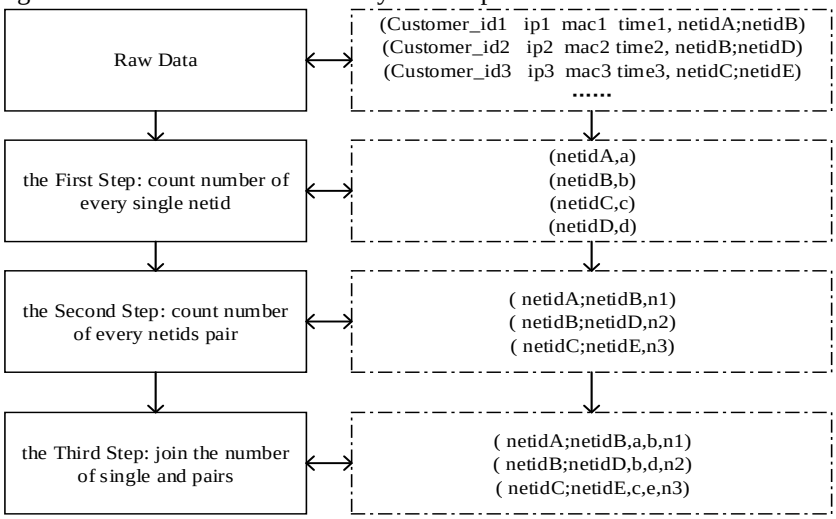

Fig. 2 Flow chart of data processing

Then calculating the probability that two NUVIs appear at the same key to judge whether the two are highly relevant or not. Therefore, we design two judge rules to find out frequent NUVIs pairs with high relevance in a more accurate way. Rules No.1 is the first step to filter NUVIs with low relevance. Only those data record meeting the requirements of rules No.1 can be kept. Figure 3 describes the detail rules. Input data's format is like (NUVIA; NUVIB, numA, numB, numAB).We assuming that a=numA&& b=numB&&n=numAB.

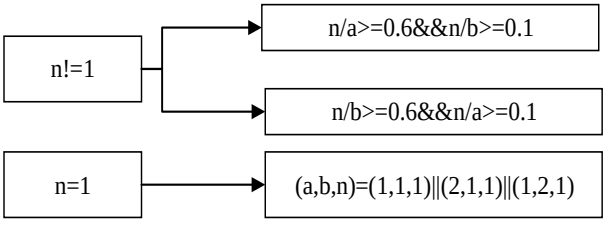

Fig. 3 Rule No.1

After the above processing, there are still a plenty of NUVIs related with more than 5 other NUVIs checking the intermediate result. Obviously, this result

is divorced from reality because few users' total QQ and email accounts will over 5. Judge rule No.2 is proposed to find out just two or three NUVIs with strongest relevance by sorting and grading for every NUVIs pair. Here has an simple example. There are two pairs of NUVIs like (id1;id2,a,b1,c1) and (id1;id3,a,b2,c2). We can see that id1 relate with id2 and id3, we grade the two pairs' relevance level according to the grading rules and find out (id1;id2) graded AA while (id1;id3) graded AB. So, we think (id1;id2) has stronger relevance.

Figure 4 describes how to grade according to the probability value. We take 0.4, 0.6 and 0.8 as cut-off points and define four grade A/B/C/D. A is the highest level and D is the lowest level. To improve the readability of program, we use double grades and scores corresponding to final level. The two grades AA/AB/BB/AC/AD/BC/BD is one to one correspondence with scores 7/6/5/4/3/2/1. Finally, scores stands the level of NUVIs pair. A higher score stands a stronger correlation.

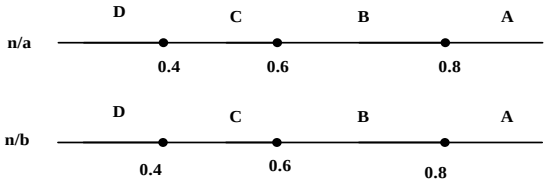

Fig. 4 Grading rules

Then we sort those pairs related with the same NUVI, such as (id1;id2 G1,id1;id3 G2,id1;id4,G3) by scores and G1>G2>G3.Rule No.2 defines which pairs should be kept and which should be abandoned.. The details of rule No.2 is as follows:

If G1=7 or G1=6, meanwhile G1-G2<=2, then we keep (id1;id2) and (id1;id3). Otherwise, we just keep (id1; id2).

If G1=5, meanwhile G1-G2<=1, then we keep (id1;id2) and (id1;id3). Otherwise, we just keep (id1;id2).

If G1=4, meanwhile G1=G2, then we keep (id1;id2) and (id1;id3). Otherwise, we just keep (id1;id2)

If G1=3 or G1=2 or G1=1, we just keep (id1; id2).

4.3. *Network virtual identities merging*

After finding frequent NUVIs pairs with high relevance, we further merge NUVIs into a whole system. The final result should be lots of group and every

group of NUVIs belongs to a user. In this part, we'll discuss an algorithm Connected Components in graph theory.

This merging algorithm is based on Spark GraphX theory. GraphX is a component in Spark for graph and graph-parallel computation. Details about GraphX and ConnectedComponents refer to official docs [4]. In graph theory, we find out all vertexes (here means NUVIs) in the same connected graph belongs to a user. The key step is to find the smallest vertex called "cc" in every graph, vertex with the same cc constitute a connected graph. Figure 5 is a simple example to describe detail process implementing Connected Components.

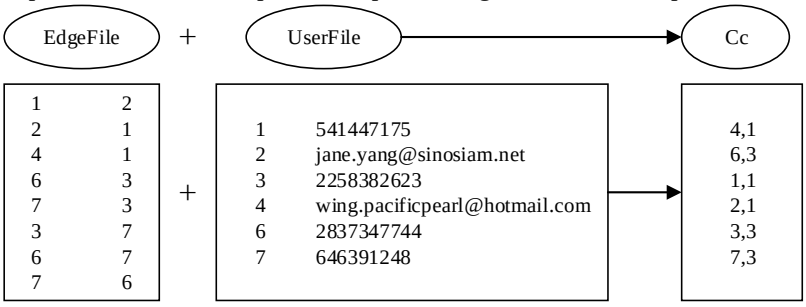

Fig. 5 Process Implementing of Connected Components

In final result, NUVIs in the same group of belong to the same user, while different group belongs to different user.

5. Conclusion

In this paper, we propose a new method to analyze network virtual identities based on Spark. We get millions and billions of NUVIs of internet users through the Network Monitoring Platform. By data preprocessing, data processing and merging processing, our research finally makes the merging correctness rate up to 90%. Our achievement can help to recognize actual person's network virtual identities. At a certain aspect, the result can be used to find out cybercriminals and protect network security.

References

1. http://www.cnnic.cn/gywm/xwzx/rdxw/2016/201608/t20160803_54389.htm.
2. SDEAN J, GHEMAWAT S. MapReduce: Simplified data processing on large clusters[J]. Communications of the ACM, ACM, 2008, 51(1):107-113.
3. http://spark.apache.org/docs/latest/.

4. http://spark.apache.org/docs/latest/graphx-programming-guide.html.
5. Gang He, Wenlong Qiu, Decheng Yu, Xiaochun Wu, “Analysis of Network User Viutual Identity based on Hadoop”, Beijing University of Posts and Telecommunications ,Beijing,China,2014.
6. ZHANRIA M, CHOWDHURY M, FRANKLIN M J,ET AL.Spark: cluster computing with working sets[C]. Proceedings of the 2nd USENIX conference on Hot topics in cloud computing, 2010: 10.
7. R.C.H.Lin, H.J.Liao, K.Y.Tung, Y.C.Lin, S.L.Wu, “Network Traffic Analysis with Cloud Platform”, Journal of Internet Technology,vol.13, no.6, pp. 953-962,November 2012.
8. Jia Li “Research of Analysis of User Behavior Based on Web Log” Computing center Anshan Normal University Anshan, China, 2013.
9. Ganju S, Wildish T, Kuznetsov V, et al. Evaluation of Apache Spark as Analytics as framework for CERN's Big Data Analytics[J]. 2010.
10. M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. Mocauley and I. Stoica, Resilient distributed datasets: A faulte-tolerant abstraction for in-memory cluster computing. In Proceedings of the 9th USENIX conference on Networked System Design and Implementation, pp. 2-2.
11. J M. Ester, H.P.Kriegel, J. Sander and X. Xu, “A density-based algorithm for discovering clusters in large spatial database with noise”, Kdd, vol. 96, no.34, pp. 226-231, 1996.
12. Zhang F, Liu M, Gui F, et al. A distributed frequent itemset mining algorithm using Spark for Big Data analytics[J]. Cluster Computing, 2015, 18(4):1493-1501.