International Journal of Grid and Distributed Computing
Vol. 9, No. 12 (2016), pp.245-254
http://dx.doi.org/10.14257/ijgdc.2016.9.12.22

Collaborative Filtering Algorithm based on User in Cloud Computing

Dan Zhang

*Institute of Technology, Mudanjiang normal university,
Mudanjiang 157000, china,
zhangdanwyc@163.com*

Abstract

The user-based collaborative filtering algorithm has been widely used in various kinds of personalized recommendation systems. But it has a serious shortcoming: with the increasing number of the users and commodities, its calculation work grows rapidly. To address the problem of vast time consumption by big dataset, we utilize MapReduce programming idea to do parallelized transformation of the algorithm; finally deploy it to be run in Hadoop cloud computing platform. Experiments have revealed that if computing data is reasonably distributed and the data volume is big, then the algorithm performance of the algorithm can realize favorable linearly speeding effect.

Keywords: *Cloud computing, MapReduce, Collaborative filtering, Hadoop*

1. Introduction

High-speed development of Internet provides abundant information to the public [1-3], meanwhile, more and more people are drowned in the network like snowballing. Hence to the masses, information explosion occurs, making them at a loose end before flooding information [4-7]. Network or business users grow exponentially. Thus, traditional data processing and service way can hardly handle that which leads to the cloud computing coming up. Cloud computing, an emerging business computing model, utilizes rapid transmission capacity of Internet to transfer data processing procedure from personal computers or servers to computer clusters over the Internet [8].

The rise of Web2.0 makes the network reach a new development peak. Many web sites visit the Web2.0 Era [9], far more than the traditional portal, the number of users as well as the degree of participation is high, it is unprecedented. How to provide convenient and efficient services for such a group of users becomes the urgent problem to solve by these websites [10-24]. And at the same time the Google has been very successful, who has built up the Google machine cluster with its file system and provides a fast search speed as along with the powerful processing capacity. How to effectively utilize these technologies to provide more powerful computing power and services for more enterprises or individuals has become a serious problem for Google such a huge resource and a large business enterprise to think deeply [25-35].It is precise because of the strong demand for massive data processing and massive computing power. On the other side it can provide this ability, so cloud computing comes into being. Of course, the Internet as the main feature of the information explosion era of cloud computing is adapt to the field is not just limited to the traditional sense of the narrow Internet. In accordance with the use of cloud computing to solve the problem of the occasion, it can use the cloud. Only did the Internet give birth to the emergence of cloud computing, improve it attention, and accelerate its promotion and development [36-40].

The emergence of cloud computing’s concept allows people to see the information explosion era to solve many problems such as massive data processing. For example, IBM, Google,Yahoo,Amazon,Sum and Microsoft and so on. IT vendors based on their respective strengths, positioning, core strategy and technology have launched their own cloud calculation plan. But most of them have their own cloud computing plans as private secrets, which makes cloud computing to a large extent appear to be just a big IT vendors patent, then ordinary users can only be served. Fortunately, open source for everyone to open a detailed solution of cloud computing window, provides a source of free download cloud computing platform [41-42]. This is still in the Apache open source organization developed by the Hadoop framework of the most classic, the most widely used.

The Internet will encounter the problem of massive data processing and computing, and the field of data mining will often encounter the same problem. This makes many existing data mining algorithms face a lot of limitations, unable to process the input data or the amount of computation. Cloud computing is good at dealing with this kind of problem, which is introduced into the cloud computing concept in many data mining algorithms. Cloud computing platform is to solve practical problems. In the field of data mining, it introduces cloud computing thinking [43-44]. The key is to realize the parallel data mining algorithm, so that it can be processed by the cloud computing platform to process massive data and the massive computation.

This paper is based on collaborative filtering algorithm, which is based on the detailed study of the process of collaborative filtering algorithm and its parallel process in the Hadoop cloud computing platform on the actual effect, convenient cloud platform services to help.

Only cloud computing platform is not enough. There must have suitable applications running in cloud platform so that the cloud computing can exert into full play [45]. But how to compile parallel program which can run in cloud platform is very particular. Different from previous single computer programming model, it has to be subject to the limit of cloud computing framework. More importantly, the algorithm needs reasonable parallelization. Whether the algorithm can be effectively parallelized is the key to it to run efficiently in cloud platform. Previous algorithm parallelization concerns only multiple threads and constrains to single computer. However the parallelization in cloud computing framework is unlike other patterns, which features' parallelization are among multiple computers or even computer cluster. So how to parallelize the algorithm in cloud computing platform is a question deserving studying. In the field of data mining, massive data is often a big headache. Cloud computing can handle this issue better. With Hadoop frame as benchmark, here we study and investigate the parallelization of four data mining algorithms, not only proving MapReduce parallelization of them but also obtaining good cluster speed-up ratio performance, solving the problems of large-scale input data and long computing time by the four methods.

2. Collaborative Filtering Algorithm

Collaborative filtering algorithm is a classical personalized recommendation algorithm [46]. It's extensively applied in lots of business recommendation systems. When dataset is too big, collaborative filtering algorithm has huge calculation amount. To address that, we deploy it after MapReduce parallelization to Hadoop platform to do computing. Since MapReduce frame in Hadoop platform allows users to decompose big problems to numerous small ones, which are then processed in a concurrent manner in the computer cluster. The operational speed is accordingly enhanced greatly.

The author [47] proposed new collaborative filtering algorithm, which is based on the combination of memory and model algorithm. The method is of small computational amount. However the author discussed algorithm process only from the perspective of traditional thinking, not considering the implementation of traditional collaborative filtering in cloud computing platform, let alone what rules to be satisfied by the speed-up ratio between running speed and computer quantity in Hadoop platform. In [48], it experimented that in a system whose internal memory is share, much better performance

of some algorithms can be achieved through simple parallel codes. In [49], it described how genetic algorithm was implemented in Hadoop platform. [50]introduced an effective and scalable MapReduce frame. The performance of CGL-MapReduce and Hadoop was compared in [51] proposed MRBench is an evaluation criteria for MapReduce system. Based on the above work, we have studied from the two points:
1 Calculate the recommendation process for each user and substitute that of traditional collaborative filtering algorithm; deploy it to run in Hadoop platform after MapReduce parallelization;
2 Performance difference of collaborative filtering algorithm implemented with cluster in Hadoop MapReduce frame and single computer; and the relationship between improvement of operation performance and DataNode number in clusters in Hadoop platform.

3. Description of Collaborative Filtering System

3.1. Hypothesis and Objective

The classical personalized recommendation algorithm, collaborative filtering algorithm, has following hypotheses:

1、 Similarities in preference and interest exist between people; people who are similar having great reference;

2、People’s preference for things is constant, not changing randomly or frequently;

3、People’s future is inheritable from the past; one’s choice in the future can be predicted through its past preference.

Based on the above hypotheses, through comparing the user’s behavior and others, the collaborative filtering algorithm firstly finds out its most similar neighbors; then predict the user’s interest and fondness according to those neighbors so as to help it make decision.

3.2. Specific Process of Collaborative Filtering Algorithm

Collaborative filtering algorithm process includes three steps: relation representation of users and items, find similar users and predict rating. Hereunder we introduce the process in details:

3.2.1 Representation of relation between user and item. The first step of collaborative filtering algorithm is to construct relation matrix between users and items, i.e. representing the relation between users and items in the form of a rating matrix; use row line for users and column for items, then matrix element M_{ij} means rating of users i for items j; if the matrix is extremely sparse, other way like storage of linking list can be utilized.

3.2.2 Search similar user. The second step is to calculate similarities between users as to find out the most similar neighboring user; similarity is generally calculated by cosine similarity computing algorithm.

$$\mathrm{si}\,m(x,y)=\frac{\sum_{s\in S_{xy}} r_{x,s}\bullet r_{y,s}}{\sqrt{\sum_{s\in S_{xy}} r_{x,s}^{2}\bullet\sqrt{\sum_{s\in S_{xy}} r_{y,s}^{2}}}} \qquad (1)$$

Where, $r_{x,s}$ represents the user's x score on the project s. $r_{y,s}$ represents the user's y score on the project s. S_{xy} A represents projects collection of users x and user review.

3.2.3 Forecast rating value. The final step of collaborative filtering algorithm is to calculate the user's evaluation of the project's value. It is shown in formu2.

$$r_{x,s} = \overline{r}_x + \frac{\sum_{y \in S_{xy}} (r_{y,s} - \overline{r}_x) \bullet sim(x,y)}{\sum_{y \in S_{xy}} sim(x,y)} \qquad (2)$$

When the score is calculated, it only needs to sort the forecast value, and it can be recommended to the users according to the predictive value.

3.3. Problems Facing the Traditional Collaborative Filtering Algorithm and Solutions

Looking back at the above analyses of the algorithm process, it’s not hard to find the computing amount of the algorithm is huge and it’s too complicated, especially when users and items are of big quantity, single computer consuming time will be a few or more days. That is hardly sufferable. If the quantity is bigger, single computer even can’t complete computation. For the reasons mentioned above, we decide to parallelize traditional collaborative filtering algorithm; then with the use of Hadoop cloud computing platform, problems like big calculation amount and longer time can be solved.

4. MapReduce Parallelization of Traditional Collaborative Filtering Algorithm

In this part we introduce the specific process of MapReduce parallelization of the algorithm. Through analyses we have noted that collaborative filtering algorithm is not only of big calculated quantity but also the calculation process is too complicated. If we parallelize MapReduce fully by following traditional procedure, it’s quite difficult to realize. Even so, the algorithm performance degrades as a result of frequent communications among various clustering nodes. Hence, we suggest user-based segmentation algorithm. This implementation algorithm encapsulates in Map procedure the process of calculation, predicated rating and recommendation of similarity between one user and the others. So Map input is a file including user ID. The implementation of our proposed algorithm has the following three stages.

4.1 Data division

At this stage, users’ ID are reasonably segmented into different files as input of Map process. Generally data division must comply with the two principles:

1.Effective computation time maximization principle: most of running time of the platform should be maximized to spend on the calculation process instead of communication for initializing mapper frequently; in the experiment in Figure1, 1000 groups of data are partitioned into 1000 portions to make mapper initialize frequently; then new generated data is divided into 40 portions and 50 portions; compare the three groups; as it is seen obviously from the picture, data division is greatly influential to computation time;

2.Task deadline consistency principle: each mapper terminates task at a generally consistent time; for instance, when there are 10 mappers, if first 9 mappers run over, the last one mapper needs another 100 seconds, we distribute evenly calculation amount done in 100s to all 10 mappers, then it requires only 10s to complete, 90s saved in this case.

4.2. Map Stage

At this stage, according to consumption of resources like internal memory of the algorithm, system assigns to each DataNode enough number of mapper for initialization. Corresponding to each file divided at the first step, system will judge whether one mapper can be initialized; if yes, initialize a new mapper, then perform operations as described in part II for each user in the file; use the result as medium value which is regarded as input of Reduce process. If there's no available resource in the cluster for initializing a new mapper, await till one mapper task completes to release resources; next initialize one mapper for the operation till all Map tasks finish.

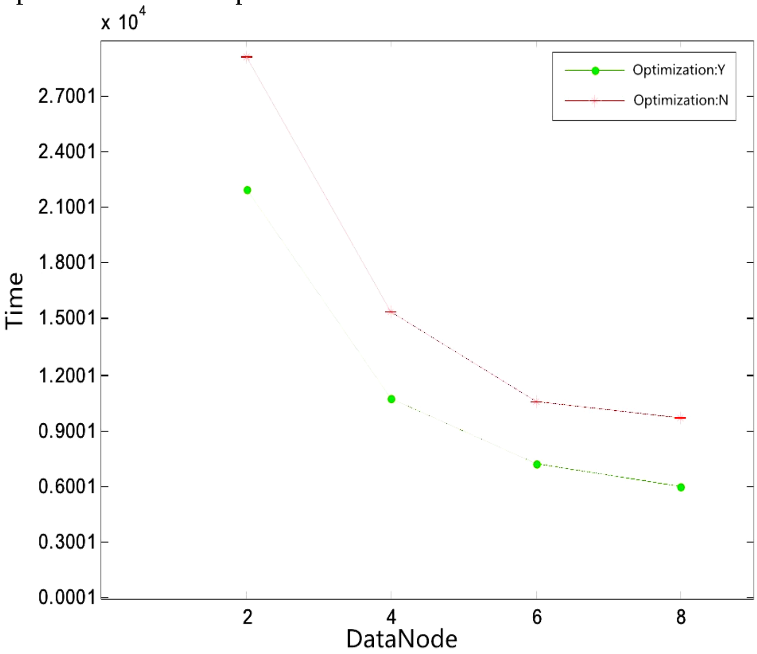

Figure 1. Performance Time Contrast of Collaborative Filtering Algorithm after MapReduce

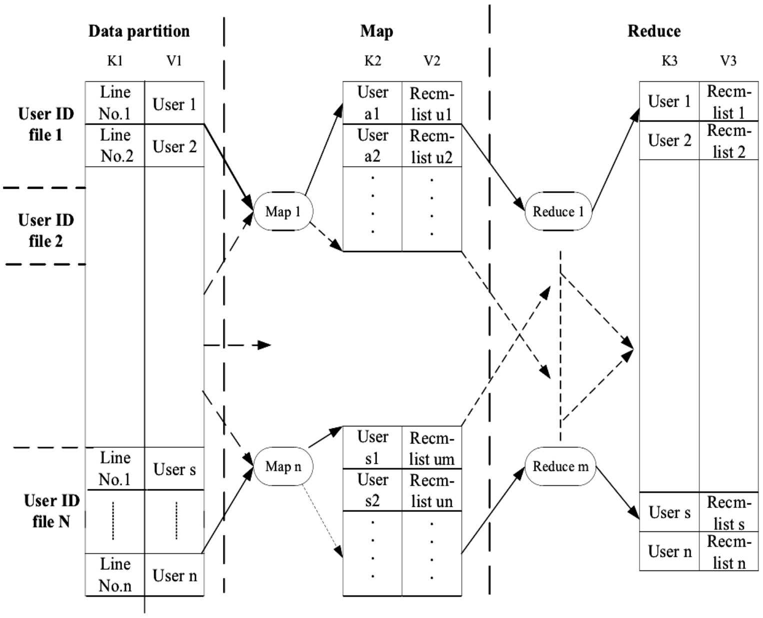

Figure 2. Collaborative Filtering of the MapReduce Process

4.3. Reduce Stage

In Reduce, system produces Reducer based on given Reduce number or self-judging the generated Reduce number. Collect users and relevant recommended items lists generated at Map stage, which are output in order as per user ID. It is shown in Figure2.

5. Experimental Analysis and Results

In the experiment, a collaborative filtering algorithm based on user and cosine similarity is used. The experimental data comes from Netflix (http://www.netflixprize.com/), and the experimental data are shown as follows:

```
1:
1488844,3,2015-08-06
822109,5,205-09-13
885013,4,2015-10-19
……
124105,4,2015-08-05
……
……
n:
1329923,2,2015-10-20
2472537,2,2015-10-23
403531,5,2015-11-19
……
1520914,3,2015-03-27
```

Arabia digital n is the film number, and the rest contains two comma lines. The first number represents the user number, and the second number represents the user's score on the front of the nearest movie, at last the third part of the date form represents the time of users' mark.

From Netflix dataset, we simultaneously choose 4 experimental dataset, consisting of 100 users, 200 users, 500 users and 1000 users, and the dataset includes more than 263,000 movies; each user evaluates differently from a few to tens of thousands of movies. Since in the paper, we discuss different running time between single computer and the cluster, so we don't mention user rating and accuracy. DataNode number is in the cluster includes 2 units, 4 units, 6 units and 8 units. During experimentation, to compare how cluster superior over single computer, with fixed DataNode number and testing dataset, we separate each type of dataset in different ways which can satisfy the data division principle. Then take the average value of consumed time in different division ways as the running time of cluster on the condition of current node number and experimental dataset. Next, compare it with running time of single computer spending on the dataset to assess advantages of cluster.

The calculation of collaborative filtering algorithm is divided by users; so when cluster is doing the test, it means different users are distributed to run in different computers. Based on theoretical analysis, speed-up ration should be 1.0; therefore, operation time is in inverse proportion to the number of cluster; speed-up rate is linearly bounding up with DataNode number in clusters. It is shown in Figure3.

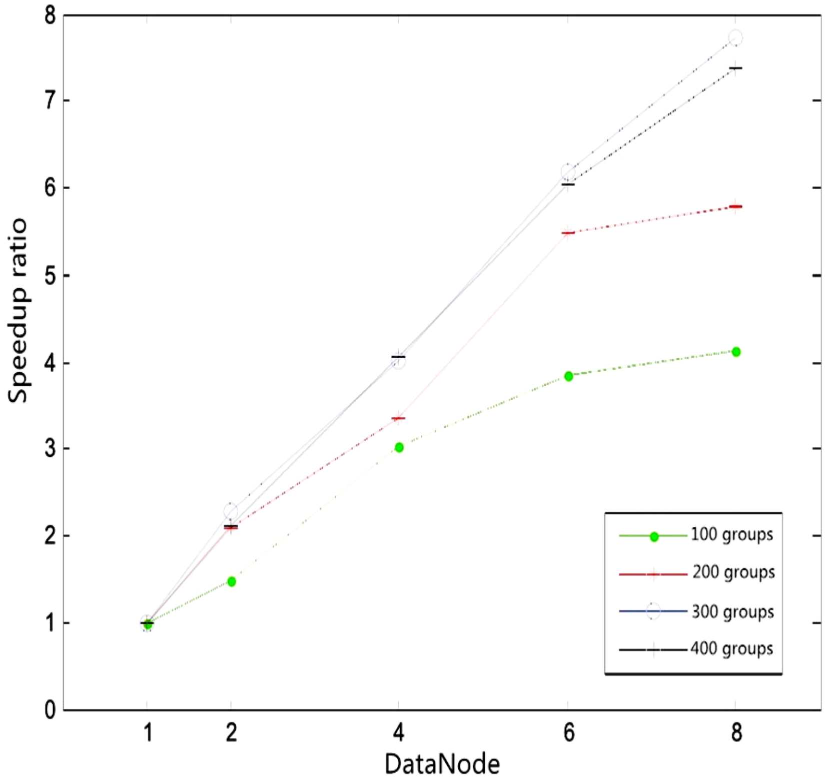

Figure 3. Performance Speedup of Collaborative Filtering Algorithm after MapReduce

As indicated from Figure3, with increasing number of DataNode in clusters, acceleration grows accordingly in times. During the testing, what's best is when DataNode in clusters has 8 computers, and cluster's acceleration being 7.72 times of single computer is a desirable value, which suggests that our algorithm can well solve the problem of longer computing time for huge data volume.

6. Conclusion

The paper has introduced the implementation of collaborative filtering algorithm in cloud computing platform. Through plenty of experiments, we prove the proposed algorithm work very effectively by reducing computation time for enormous data. Also through lots of experiments, data division principle is derived by the collaborative filtering algorithm running in Hadoop. Despite our method can better shorten running time for calculation by abundant users, if recommendation is made only to single user, its running time remains a constant at a certain scale, not improving its running speed. In the future, we will work on how to improve system's reaction speed with recommendation only to single user.

In this paper, the principle and implementation process of collaborative filtering algorithm based on users and the main defects of the algorithm are described. A method of MapReduce based on this algorithm is proposed, and the MapReduce of the algorithm is proved by the cluster experiment.

Acknowledgement

This work is supported by Mudanjiang Normal College science and technology research projects. No. QY2014002

References

[1] H. Ying, “Research on Collaborative Filtering Recommendation Algorithm Based on user characteristics and cloud model”, Jiangxi University of Science and Technology, **(2014)**.

[2] L.-F. Li, “Personalized recommendation of cloud services based on the relationship between service network and service”, Beijing University of Posts and Telecommunications, **(2015)**.

[3] Y.-H. Fang, “Ever grand platform of cloud based data mining”, Jilin University, **(2015)**.

[4] G. Sui, “Research on E-commerce Recommendation Algorithm Based on collaborative filtering”, Shandong Normal University, **(2014)**.

[5] Z. Yang, “Research and application of recommendation system based on Hadoop”, Hubei University of Technology, **(2014)**.

[6] C.-Y. Zhao, “Basic theory and key technology research of personalized tourism information service system”, Lanzhou University, **(2012)**.

[7] H.-F. Sun, “Personalized Web recommendation based on collaborative filtering”, Beijing University of Posts and Telecommunications, **(2012)**.

[8] Z. Nvsen, “Research and implementation of dynamic recommendation system algorithm based on user behavior”, University of Electronic Science and technology, **(2013)**.

[9] Q.-Z. Liang, “Multi objective task scheduling algorithm on hybrid cloud platform”, China University of Geosciences, **(2015)**.

[10] F.-F. Han, “Research on e-commerce personalized recommendation based on cloud computing”, North China Electric Power University, **(2014)**.

[11] X.-W. Liu, “Research on the analysis method of large data mining based on collaborative filtering”, Zhejiang University of Technology, **(2014)**.

[12] C. Chen, “Research on Web service QoS prediction technology in cloud computing mode”, Jiangxi University of Finance and Economics, **(2014)**.

[13] Chen, Promise, Design and implementation of recommendation system based on personalized recommendation engine recommendation system”, South China University of Technology, **(2012)**.

[14] H.-M. Li, “Research on collaborative filtering algorithm in e-commerce recommendation system”, Jilin University, **(2011)**.

[15] Q. Xiao, Q.-H. Zhu, H. Zheng and K.-W. Wu, “Design and implementation of distributed collaborative filtering algorithm under the environment of Hadoop”, Modern library and information technology, no. 22901, **(2013)**, pp. 83-89.

[16] L. Yi, J. Ya and C. Chen, “A personalized recommendation system based on cloud computing technology”, Computer engineering and applications, vol. 51, no. 83613, **(2015)**, pp. 111-117.

[17] T.-H. Sun, L.-M. Leann and Q.-S. Zhu, “Research on Collaborative Filtering Recommendation Algorithm Based on Hadoop distributed improved clustering”, Computer engineering and application, vol. 51, no. 83815, **(2015)**, pp. 124-128.

[18] X. Xu, “The king of Xufa, Collaborative filtering algorithm of similarity optimization method”, Computer Engineering, vol. 36, no. 35106, **(2010)**, pp. 52-54.

[19] Y.-Y. Zhao and S.-W. Gu, “A service recommendation algorithm for cloud computing environment”, Journal of Chaohu College, vol. 14, no. 11403, **(2012)**, pp. 42-47.

[20] K.-C. Li and Z.-G. Liang, “Adapting to changes in user interest is exponential forgetting collaborative filtering algorithm”, Computer engineering and applications, vol. 13, **(2011)**, pp. 154-156.

[21] H.-G. Fu and R. Li, “Collaborative filtering algorithm based on user feedback in real time”, Computer application, vol. 07, **(2011)**, pp. 1744-1747.

[22] W.-H. Huang, X.-W. Meng and L.-C. Wang, “Collaborative filtering algorithm based on user socialization in the mobile communication network”, Journal of electronics and information, vol. 12, **(2011)**, pp. 3002-3007.

[23] B. Hu and D.-W. Peng, “A collaborative filtering algorithm based on user characteristics and time”, Journal of Wuhan University of Technology, vol. 03, **(2009)**, pp. 24-28.

[24] C.-Q. Hou, L.-C. Jiao and W.-J. Zhang, “A collaborative filtering algorithm for sparse user rating matrix”, Journal of Xi'an Electronic and Science University, vol. 04, **(2009)**, pp. 614-618.

[25] Q. Wang, L.-Y. Yang and D.-L. Yang, “A distributed collaborative filtering algorithm for attribute value based on user preference”, Journal of systems engineering, vol. 04, **(2010)**, pp. 561-568.

[26] S. Y. Liang, Y.-D. Su, M. Kai and Z. Q. Yan, “Consider the user background information collaborative filtering algorithm”, Microcomputer information, vol. 36, **(2010)**, pp. 197-198.

[27] Y.-Q. Feng and Y. Zhang, “Collaborative filtering algorithm based on user's topic preference in data sparse environment”, Operation and management, vol. 02, **(2014)**, pp. 145-152.

[28] L. Zhang and T. Qin, “The use of key users of social networks to improve collaborative filtering algorithm performance”.

[29] P.-Q. Zhang, Q. Li and T. Tao, “An improved collaborative filtering algorithm based on user clustering”, Information science, vol. 10, **(2014)**, pp. 24-27.

[30] J. Wu and F. Feng, “Collaborative filtering algorithm for integrated user preference and priority new product recommendation”, Computer applications and software, vol. 10, **(2014)**, pp. 285-287.

[31] Y.-P. Chen and S. Wang, "Hybrid collaborative filtering algorithm based on user – project", Computer technology and development, vol. 12, **(2014)**, pp. 88-91.

[32] Z. Li and X.-X. Yan, "User recommendation ability of collaborative filtering algorithm affects the performance comparison analysis", Library and information work, vol. 2, **(2014)**, pp. 215-219.

[33] C. G. Wei, Y.-Z. Ding and Z.-Q. Wu, "A collaborative filtering algorithm based on user ratings and project tags", Computer technology and development, vol. 03, **(2015)**, pp. 71-75

[34] N.-J. Sun and L. Liu, "Collaborative filtering algorithm based on item classification and user group interest", Computer engineering and application, vol. 10, **(2015)**, pp. 128-131.

[35] W. Zhao, J.-F. Li, Y. Han and H.-T. Zhang, "Research on the user based collaborative filtering algorithm based on the collaborative filtering algorithm in Hadoop", Cloud platform computer measurement and control, vol. 06, **(2015)**, pp. 2082-2085.

[36] Jin, "Research on collaborative filtering algorithm based on improved user interest", Modern economic information, vol. 21, **(2015)**, p. 88.

[37] D.-W. Peng, D.-H. Liu and H. Zhang, "A collaborative filtering algorithm based on time weighted and user characteristics", Journal of Wuhan University of Technology, vol. 05, **(2012)**, pp. 144-148.

[38] X. Huang, S.-Y. Wei, N. Ye, J. Zhu and S. Zhang, "Collaborative filtering algorithm based on user attributes and item categories", Computer and digital engineering, **(2012)**, vol. 10, pp. 5-7.

[39] Y.-T. Wu, X.-M. Zhang, M. Xing and L.-H. Wang, "Based on fuzzy similarity of user collaborative filtering algorithm", Journal on communications, vol. 01, **(2016)**, pp. 198-206.

[40] W. Cheng, Z.-G. Zhu, Y.-X. Zhang and F.-F. Su, "The recommendation efficiency and personalized improvement of the collaborative filtering algorithm based on users".

[41] X.-F. Wu and Jia, "Collaborative filtering algorithm based on user characteristics and time effect", Modern computer (Professional Edition), vol. 10, **(2016)**, pp. 21-24.

[42] W. Wang and J. Zheng, "Collaborative filtering algorithm based on the similarity of users", Journal of East China Normal University (Natural Science Edition), vol. 03, **(2016)**, pp. 60-66.

[43] H.-M. Wang, "Nie planning, Collaborative filtering algorithm for the integration of user and project related information", Journal of Wuhan University of Technology, vol. 07, **(2007)**, pp. 160-163.

[44] S. Chen and Z.-J. Dong, "Collaborative filtering algorithm which reflects the change of user interest", Computer application and software, vol. 06, **(2013)**, pp. 295-297.

[45] Z.-W. Wang, "Collaborative filtering algorithm based on trust user association clustering", Computer and modernization, vol. 09, **(2013)**, pp. 50-53.

[46] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art andpossible extensions", IEEE Trans. on Knowledge and Data Engineering, vol. 17, no. 6, **(2005)**, pp. 734-749.

[47] A. S. Das, M. Datar and A. Garg, "Google news personalization: scalable online collaborative filtering", WWW: Proceedings of the 16th international conference on World Wide Web, **(2007)**; Banff, Alberta, Canada.

[48] J. Chao, C. Vecchiola, R. Buyya, "MRPGA, An Extension of MapReduce for Parallelizing Genetic Algorithms", eScience, eScience, IEEE Fourth International Conference, **(2008)**; Canada.

[49] A. W. McNabb, C. K. Monson and K. D. Seppi, "Parallel PSO using MapReduce", Evolutionary Computation, CEC, IEEE Congress, **(2007)**, pp. 7-14.

[50] J. Ekanayake, S. Pallickara and G. Fox, "MapReduce for Data Intensive Scientific Analyses", eScience, eScience, IEEE Fourth International Conference, **(2008)**.

[51] K. Kiyoung, J. Kyungho and H. Hyuck, "A Benchmark for MapReduce Framework.Parallel and Distributed Systems", ICPADS, 14th IEEE International Conference, **(2008)**.

Author

Dan Zhang. **S**he received her B.S degree from Harbin Normal University and received her M.S degree from University of Electronic Science and Technology of China. She is a lecturer at Institute of Engineering of Mudanjiang Normal University. Her research interests include software testing.