International Journal of Grid and Distributed Computing
Vol. 9, No. 7 (2016), pp.169-176
http://dx.doi.org/10.14257/ijgdc.2016.9.7.18

Parallel Collaborative Filtering Recommendation Algorithm based on Cloud Computing

Guohua Zhang, Feng Bao*, Sheng Bai

Northeast Petroleum University, No.199, Fazhan RD.,
High-tech development zone, Daqing, China
sicaiyitang@163.com; baofengqhd@sohu.com

Abstract

The paper Proposed Item parallel collaborative filtering recommendation algorithm (IP-CF). Through designing efficient parallel algorithm, compute-extensive procedures are distributed to different processing nodes in Hadoop platform. Taking advantage of parallel computing, we accelerate the response of recommendation. The experimental results show that our proposed algorithm IP-CF is more efficient and scalable than current parallel algorithms.

Keywords*: Recommendation system, Collaborative filtering , Cloud computing*

1. Introduction

The recommendation based on collaborative filtering [1-5] has become one of the most successful recommendation methods, more attentions were paid to the recommendation based on item collaborative filtering by more e-commerce websites [6-10]. However, with increasing number of users and commodities, in the calculation of similarity and prediction, single centralized calculation can't meet the requirements of system instantaneity and scalability [11-14]. To improve the real-time performance of recommendation algorithm, [15] implemented concurrently on Hadoop platform the recommendation algorithm based on item collaborative filtering. But in the algorithm designed by [15], the item pair on average bought by user is calculated through looking up all buying records of the two items as to find all users who bought them [16-20]. Such kind of method has time complexity $O(m^2 * n^2)$. Moreover, by regarding user as center, the method computes in two MapReduce procedures the score of predicting that user does not buy item: send item bought by same client to the same node [21-24]; when the prediction is calculated, it's required to read similarity matrix to the memory on each node, read similarity of identical items to each item of the user; it reads too many redundant data, and in the case of enormous data, the quantity of both users and commodities is too huge to be read in due to limited scale of similarity matrix, which will affects the scalability of parallel collaborative filtering algorithm [25-26]. [1]

Here we propose item parallel collaborative filtering recommendation algorithm (IP-CF). We'll implement concurrently item-based collaborative filtering algorithm with MapReduce framework and HDFS on Hadoop cluster. Regarding problems of the method designed in [15], IP-CF method calculating user buying item pair is realized by sending any item pair as key to REDUCE nodes, sending items with same item pair to the same node. The method has time complexity $O(m^2 * n^2)$, decreasing greatly the searching time. What's more, an efficient parallel strategy is formulated, a MapReduce procedure which realizes rating prediction function achieved by two MAPREDUCE in [15]. In that way, calculation results are kept in memory, lessening communication amount, improving

Feng Bao is the corresponding author.

ISSN: 2005-4262 IJGDC

algorithm implementation efficiency and obtaining better real-time performance and scalability. Experiments prove IP-CF algorithm is effective.

2. Collaborative Filtering Recommendation Algorithm based on item

Item-based collaborative filtering means matching user bought or evaluated article to similar one; then put similar commodities to recommendation list. In a sense of calculation, it refers to using all users' comments about one purchased commodity as one vector; each user's rating of it as component of the vector; then, calculating similarity between commodities by means of vector value; after alike commodity of it is acquired, predicting current user's evaluation of non-purchased item based on user buying records to get a goods list based on predicted value as recommendation. The algorithm includes four major parts.

(1) Preprocess rating data and calculate item's average score;
(2) Calculate rating similarity between items by certain measuring method;
(3) Calculate predicted score of unrated item by certain prediction rating model;
(4) Measure time effectiveness and accuracy of the algorithm

3. Parallelization of Item-based Collaborative Filtering Recommendation Algorithm

In the item-based collaborative filtering recommendation algorithm, the most time-consuming stage is to calculate similarity matrix and predict user preference. If user number is M and commodity item number N, then the time complexity of calculating similarity is $O(n^2m)$ and predicted time complexity is $O(nm)$. Obviously, in calculating similarity matrix, calculating the similarity of one item pair is independent from that of other pair. Hence, the similarity matrix can be computed in a parallel way. Secondly, when user preference is predicted, the calculation of different users' preference is mutually independent. Thus, the prediction of user preference can be calculated concurrently. However, the calculation of similarity must complete before prediction of user preference. In other words, they're of serial execution.

MAPREDUCE is so far a popular parallel programming model. User needs only to specify the calculation process of MAP and REDUCE function. The system will automatically calculate concurrently on large-scale cluster. Google and Hadoop both realized MAPREDUCE.

With MAPREDUCE model, IP-CF algorithm can be implemented, through calculation in three procedures of both MAP and REDUCE. MAP1 calculates each user bought item ID and its scoring; REDUCE1 calculates mean value of item ratings; MAP2 calculates all item pairs in a same user's buying records; REDUCE2 calculates similarity between items; MAP3 calculates all other items identical to item Item_i; REDUCE3 calculates predicted rating of non-evaluated items. After calculation, MAP1 and REDUCE1 get items and their mean values, which are saved in AV-FILE. The calculation process is too simple that we won't introduce more.

User rating data file USER-FILE read by MAP2 function is stored in HDFS, where each line represents item buying records of one user, which are read to MAP nodes in the form of key value pair <Key, Value>. Each key value pair stands for a piece of data record; KEY is offset of current record against the initial point of input data file; Value is item ID and its score in current records; MAP2< Key, Value> calculates all item pairs by current customer and their relative ratings. The pseudo code of the MAP2 function is in Table1.

Table 1. Pseudo Code of the MAP2 Function

```
Input:
    Key value-Key, the value of the key corresponds to value
Output:
    Key value- key' , the value of the key corresponds to value'
Pseudo code flow:
MAP2(Key,Value)
{
1:GenItemsandRatings( value,item,rating,len);
2:  For i=0 to len-1 do
3:    For  j=I to len do {
4:    item_a=Items[i]; rating_a=Ratings[i];
5:    item_b=Items[j]; rating_b=Ratings[j];
6:      key' =CombineItems(item_a,item_b);
7:      value' = CombineItems(rating_a,rating_b);
8:      Output < key' , value' >;
9:    }
10: }
```

After MAP2 outputs, the system will send items with same KEY to the same REDUCE node. REDUCE2< Key, Value> is responsible for calculating similarity between two items to which item pair KEY corresponds. The pseudo code of the REDUCE2 function is in Table2.

Table 2. Pseudo Code of the REDUCE2 Function

```
Input:
    Item label -Key, Item correspondence rating- value
Output:
    Item key' correspondence similarity
Pseudo code flow:
REDUCE2(Key,Value)
{
1:AssignValue(value,ratings_i,ratings_j);
2:  ReadFileAveRat(AV-FILE,items,aveRatings);
3:    For each val in value{
4:    ReadRatings(val,rating_i[k],ratings_j[k]);
5:    k++;
6:      }
7:      value' = Pearson();
8:      key' = key ;
9:    Output < key' , value' >;
10: }
```

After REDUCE2 executes, the system gets item similarity matrix. It will predict scores of items not evaluated by each user.

< Key, Value> output by REDUCE2 used as input of MAP3, KEY as item pair and VALUE as similarity of item pair. MAP3< Key, Value> is responsible for sending all item pairs which contain one item in them to the same REDUCE node. The pseudo code of the MAP3 function is in Table3.

Table 3. Pseudo Code of the MAP3 Function

```
Input:
    Item Encoding -Key,
Output:
    Item key' correspondence to Item Encoding, List of similarity- value'
Pseudo code flow:
MAP3(Key,Value)
{
1: ReadItem(key,item1,item2);
2:  key' =item1;
3:   value' =Combine(item2,.value);
4:   Output < key' , value' >;
5:   key' =item2;
6:     value' =Combine(item1,.value);
7:   Output < key' , value' >;
8: }
```

After MAP3 executes, the system will send items similar to item KEY and similarity between them to the same REDUCE node. REDUCE3< Key, Value> calculates predicted score of non-evaluated items. The pseudo code of the REDUCE3 function is in Table4.

Table 4. Pseudo Code of the REDUCE3 Function

```
Input:
    Item Encoding -Key, List of similarity- value' , User rating file-AV-FILE, Item average
rating file-AV-FILE.
Output:
    Project forecast rating - value'
Pseudo code flow:
REDUCE3(Key,Value)
{
1: ReadFileAveRat(AV-FILE,allItems,aveRating);
2:  RaedFileRat(user-file,userid,items,rating);
3:    For each val in value;
4:    ReadItemSimi(val,item[k],simi[k]);
5:     k++;
6:      }
7:     If not items k in the array Item
8:     value' =Prediction();
9:     key' =Combine(userid,key);
10:}
11:  Output < key' , value' >;
12:  }
```


4. Experiment Design and Discussion

4.1. Experimental Platform and Dataset

The experimental dataset used here is chosen from GROUPLENS provided by MOVIELENS. The dataset includes comments on 20111 movies by 81454 users. Each user evaluated at least 20 movies, at rating level 1-5. The movie rating sparsity is 0.867. The experiment is executed on 21 servers (Octa-core CPU) which constitute the parallel computing environment. The proposed parallel collaborative filtering recommendation algorithm is implemented through Hadoop.

User rating data in the recommendation system includes user collection $U = \{u_1, u_2, ...u_m\}$ and item collection $I = \{i_1, i_2, ...i_n\}$. User rating matrix $M = (r_{i,j})_{m \times n}$ is expressed by a m*n order; each element in the matrix represents user rating of item, suggesting user preference for it. The rating can be expressed by 1/0, meaning like or hate, or by number, meaning degree of fondness. It is shown in Table5.

Table 5. User Item Rating Matrix M

	I_1	...	I_j	...	I_n
U_1	$r_{1,1}$	...	$r_{1,j}$	...	$r_{1,n}$
...	...	...	...	...	...
U_i	$r_{i,1}$	....	$r_{i,j}$	...	$r_{i,n}$
...	...	....	...	...	...
U_m	$r_{m,1}$	....	$r_{m,j}$	...	$r_{m,n}$

4.2. Measurement Method

Use U to represent the collection of users who evaluate both item i and j; then the rating similarity between item i and j are $sim(i, j)$, which can be acquired through Pearson correlation coefficient [27]. It is shown in Formula1.

$$sim(i, j) = \frac{\sum_{u \subset U} (R_{u,i} - \overline{R}_i)(R_{u,j} - \overline{R}_j)}{\sqrt{\sum_{u \subset U} (R_{u,i} - \overline{R}_i)^2} \sqrt{(R_{u,j} - \overline{R}_j)^2}} \quad (1)$$

To make prediction more accurate, we applied an optimized rating prediction strategy: regard the average score of items requiring evaluation by target users as benchmark; find out the neighboring collection of target users' target items; calculate predicted rating of target items with similarity of items in neighboring collection [28]; user u's rating value of item i is predicted through calculation formula2 as follows:

$$P_{u,i} = \overline{R}_i + \frac{\sum_{j \in S(i)} sim(i, j)^* (R_{u,i} - \overline{R}_j)}{\sum_{j \in S(i)} | sim(i, j) |} \quad (2)$$

4.3. Experimental Results and Analysis

To validate scalability of the system, we designed two groups of tests. In the first group, we define calculated node number and the size of testing dataset is variable; in the second group, we define the size of dataset and calculated node number is changeable. In item-based collaborative filtering recommendation, the calculation of similarity can be

operated offline. It requires real-time calculation only to predict user rating. So the time in the testing is what's used for predicting item rating. It is shown in Figure1-2.

From Fig. 1 and Fig. 2, we see item-based collaborative filtering recommendation algorithm reduced considerably online prediction time after parallel implementation; also with growing data volume, it demonstrated favorable scalability by adding the number of calculated nodes.

To compare time effectiveness of IP-CF approach and that in [15], we designed a group of test. The dataset is 10M, which runs respectively on 1, 10, 20, 30, 40, 50, 60 node. The execution time of them is compared in Figure3. The proposed method performs better than that in [15] as experiments suggest.

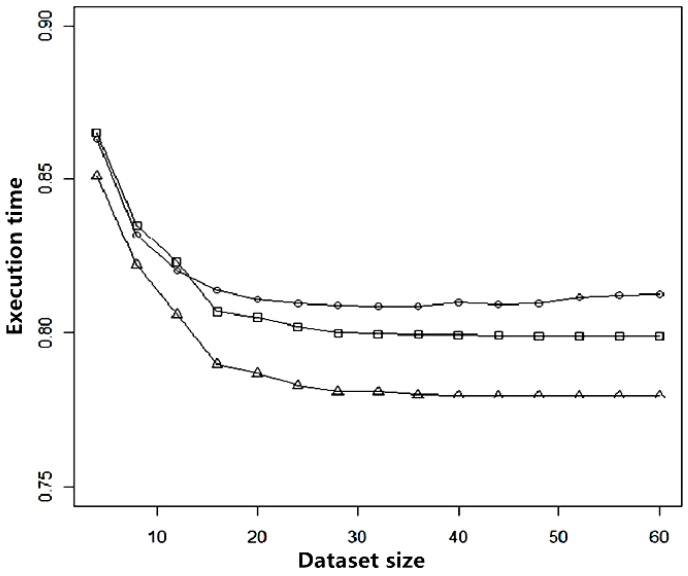

Figure 1. The Relationship of Execution Time and Dataset Size

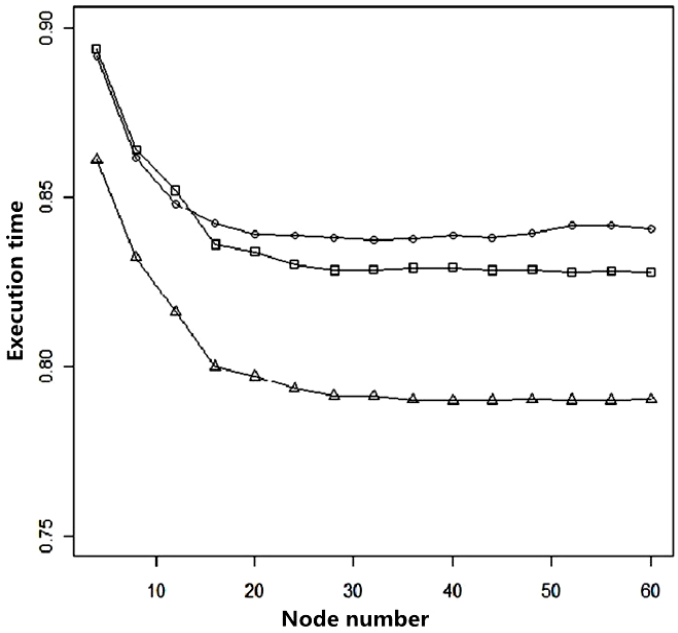

Figure 2. The Relationship of Execution Time and Node Number

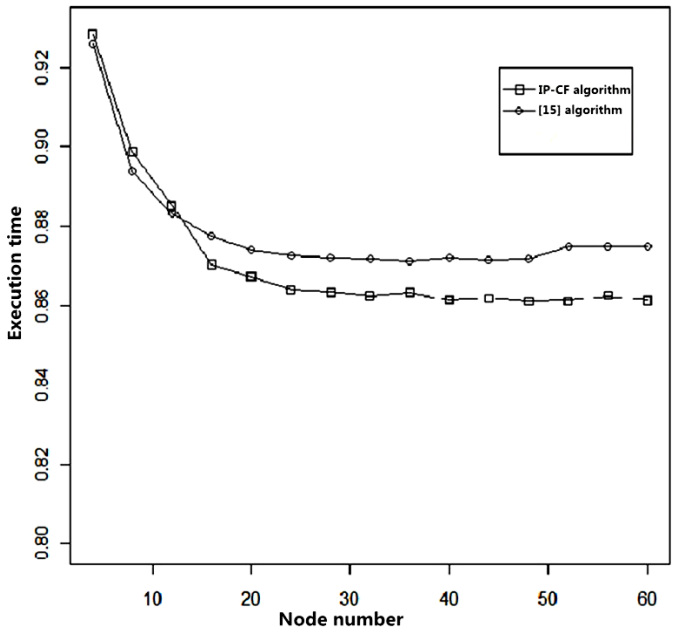

Figure 3. Comparison of the Execution time of IP-CF Algorithm and Literature[1] Algorithm

5. Conclusion

The paper introduced item-based collaborative filtering recommendation algorithm and then it proposed a new parallel recommendation method IP-CF, which was implemented on Hadoop cluster. By assigning reasonably the process of calculating similarity which involves too much calculation and item prediction rating procedure to each processing node, the new method reduced data traffic and execution time. IP-CF algorithm proved its effectiveness in the experiments.

References

[1] Sun Caiqi. Research on context aware recommendation algorithm based on trust network. Donghua University, 2015
[2] Zhang Chunli. Research and implementation of collaborative filtering algorithm based on Hadoop, Donghua University, 2015
[3] Li Fanglin. Research and implementation of parallel recommendation algorithm. Beijing University of Posts and Telecommunications, 2015
[4] Wu Chengchao. Research on the dynamic performance of collaborative filtering recommendation algorithm. University of Science & Technology China, 2015
[5] Luo jie. Collaborative filtering recommendation algorithm based on item-user. Kunming University of Science and Technology, 2014
[6] Miao Yu. Research and implementation of collaborative filtering algorithm based on matrix decomposition. Beijing University of Technology, 2014
[7] Wang Baoying. Collaborative filtering algorithm based on the comprehensive similarity of projects. Hebei University of Technology, 2014
[8] Li Shantao. Research and application of information recommendation technology based on cross domain. Beijing University of Posts and Telecommunications, 2014
[9] Zheng Jian. Research and implementation of collaborative filtering recommendation system based on Hadoop. Beijing University of Posts and Telecommunications, 2014
[10] Shi Qingxia. Collaborative filtering and personalized recommendation algorithm based on user's interest. Tianjin Normal University, 2013
[11] Fang Weihua. Research on an improved collaborative filtering recommendation algorithm for recommendation system. Xi'an Electronic and Science University, 2013

[12] Wen Junho, Shu Shan. An improved collaborative filtering recommendation algorithm of similarity measure. Computer science, 2014,05:68-71.
[13] Fan Bo, Cheng Jiujun. Among multiple users similarity collaborative filtering recommendation algorithm. Computer science, 2012,01:23-26.
[14] Ma Hongwei, Zhang Guangwei, Li Peng,. Review of collaborative filtering recommendation algorithm. Small and micro computer system, 2009,07:1282-1288.
[15] Jiang J, Lu J, et al. Scaling-up item-based on collaborative filtering recommendation algorithm based on Hadoop. Proceedings of 2011 IEEE world Congress on Services. Washington Marriott, DC, USA, IEEE, 201:490-497
[16] Xin Juqin, Jiang Yan, Shu Shaolong. Personalized recommendation of integrated user preference model and BP neural network. Computer engineering and application, 2013,02:57-60.
[17] Han Lu. Collaborative filtering recommendation algorithm research. Wireless Internet technology, 2013,02:160.
[18] Zhang Yao, Chen Weibin, Fu Shunkai. Review of collaborative filtering recommendation research. Microcomputer and application, 2013,06:4-6.
[19] Yang Zhiwen, Paul Liu. Collaborative filtering recommendation algorithm based on Hadoop platform. Application of computer system, 2013,07:108-112.
[20] Zhao Xiangyu. Top-N collaborative filtering recommendation technology research. Beijing Institute of Technology, 2014
[21] Gao Jianhuang. Technology and application of personalized recommendation system. University of Science & Technology China, 2010
[22] Jinbo Zhang. Research and implementation of collaborative filtering recommendation system. Beijing University of Posts and Telecommunications, 2010
[23] Xiong Yuran recommendation algorithm based on user behavior trajectory. University of Electronic Science and technology, 2013
[24] Liu Xin. Research and implementation of Web Recommendation System Based on hybrid recommendation. Beijing University of Technology, 2013
[25] Deng Xiongjie. Design and implementation of a recommendation system based on Hadoop. South China University of Technology, 2013
[26] Yu Xue. Research on the recommendation method based on collaborative filtering technology. Tianjin University, 2009
[27] Ahn HJ. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Information Science,2008,178(1):37-51
[28] Herlocker J L, Konstan J A, et al .Evaluating collaborative filtering recommender system. ACM Transactions on Information Systems,2004,22(1):5-53

