International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 04 | July-2015 www.irjet.net p-ISSN: 2395-0072

IMPLEMENTING COLLABORATIVE FILTERING ON LARGE SCALE DATA USING HADOOP AND MAHOUT

Swati Sharma[1], Manoj Sethi[2]

[1]*Mtech Student, Deparment of Computer science and Engineering, DTU, Delhi, India.*
[2]*A.P. , Deparment of Computer science and Engineering, DTU, Delhi, India.*

Abstract - ***We are living in an age of Data and Information. Online social networks are contributing in enlargement of this data on high scale and Recommendation systems are helping industries to make this data useful for business purposes. It is helping to enhance the opportunities in online social data. Online social network generate large quantity of data from its users and recommendation system use this data for suggesting right piece of information to the user. But in the time of Big Data, processing large volumes of data for generating suggestions is a difficult job. We are aiming to implement recommendation algorithm using Apache Mahout, a machine learning tool, on Hadoop platform to provide a scalable system for processing large data sets efficiently.***

Key Words: *Recommendation system, Large scale data, Hadoop, Apache Mahout, Collaborative filtering.*

1. INTRODUCTION

In the recent years WWW has grown on exponential rate and that resulted into huge amount of data. This has become an opportunity as well as a problem for user because finding the right information has become difficult. E-commerce has been gone through challenges of managing this huge explosion of information and utilized it in a smarter way by using recommendation systems. By the rise in online shopping, recommender systems has become basic need of every e-commerce portal. Recommendation systems comes with the capability of predicting the suggestions for its users on the bases of user's past behavior or the by the behavior of similar users. Amazon, Netflix and other such portal use recommender systems extensively for suggesting content to their users. Collaborative filtering is based upon the idea of predicting user's choice of purchasing by considering its past behavior and choices or behavior and choices of similar user. It utilizes the prior knowledge of item and user.

Recent years have faced a vast increase in online data and user which leads to rise of Big data. Big data has made recommendation system more important for the users as it predict right piece of information out of huge amount of information. But it also leads to problem of scalability of system and algorithm. In this paper, we are going to implement and analyze collaborative filtering for making recommendations on the top of Hadoop[1] platform using Apache Mahout[2] and MovieLens dataset[3] to see the performance on the base of scalability and speedup.

2. RELATED WORK

Area of recommendation has been became interest of researcher from past two decades. Because of its close relation to e-commerce business it has fascinated researchers as well as industries to work in the field of recommendation system. Research is going in the field of content recommendation and friend recommendation. With the popularity of online social networks, potential friend recommendation has turned into area of interest for many researchers. Here is some related work listed. Deuk Hee Park et.al. have presented a classification and literature review in their paper and gives insight about past work and future scope of area[4].

X. Yang et.al. have compared recommender systems based on collaborative filtering [5]. Ruzhi Xu et.al. implemented CF and made predictions using singular ratings for large scale recommendation using Hadoop MapReduce distributed framework for improving efficiency [7].

J. Bobadilla et.al. provided an overview of recommender systems and algorithms used in this system [8]. Saikat Bagchi has studied the performance and quality of similarity measures for CF using Mahout and observed that Euclidean Distance Measure perform better than other measures provided in Mahout [9].

Xiwei Wang et.al. has studied different algorithm and models for recommender systems [10]. Juha Leino and Kari-Jouko Räihä tried to understand the user strategies used in

complex e-commerc system by taking the case study of Amazon [11].

Carlos E. Seminario and David C. Wilson has given a case study on Mahout and evaluated it as a recommender system platform[12]. Xing Xie has designed a friend recommendation framework on the basis of user interest characterization in online social network [13].

G. Adomavicius and A. Tuzhilin reviewed different recommendation methods and studied their limitations and discussed extensions to methods to provide better results in recommendations.

3. HADOOP

Apache Hadoop is an open-source software framework written in Java for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware [4].

Hadoop has two sub-divisions namely HDFS (Hadoop Distributed File Syetem) and MapReduce programming model. Hadoop ideally breaks the data into large chunks and distributes it to its commodity hardware cluster nodes for further processing using MapReduce programming model for distributed computing as shown in fig. – 2 thus able to handle large datasets. MapReduce was initially developed by Google for counting the no. of times a word occurs in particular document. It works well for applications where data is stored at distributed file system which allows local computing on each data node.

The base framework for Apache Hadoop includes HDFS, MapReduce, Yarn, Hadoop common utilities as shown in fig – 1. This depicts basic hadoop framework though after 2012, additional softwares also included into Hadoop package that can be installed on the top of Hadoop such as Apache Pig, Apache Spark, Apache Hive, Apache HBase etc.
HDFS is distributed file system for Hadoop to store and manage large datasets on commodity hardware. MapReduce is programming model to write applications to process large datasets on clusters with reliability and fault tolerance.
Apache Mahout is Java written library for machine learning algorithms that are scalable and can be implemented on the top of Hadoop using MapReduce framework for analyzing Big Data.

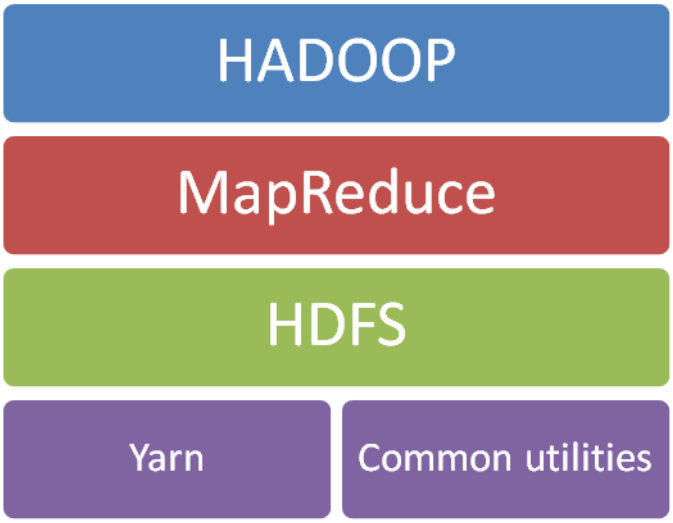

Fig-1: The base Hadoop framework.

The algorithms provided in Mahout include Collaborative filtering, clustering, and classification algorithm.

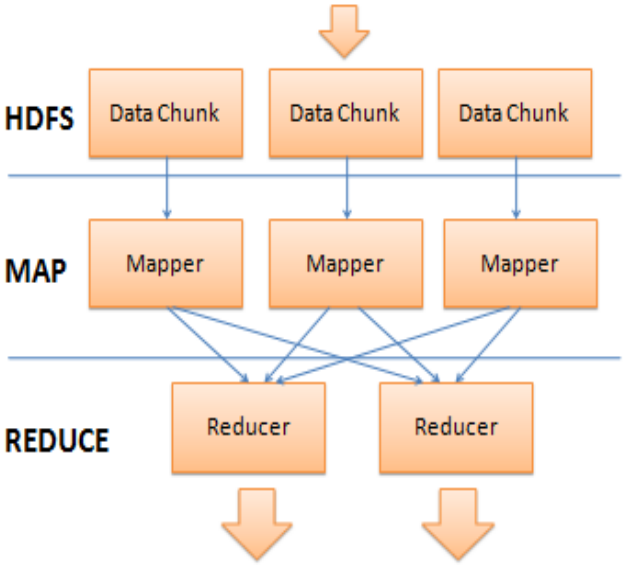

Fig-2 : MapReduce Framework

4. COLLABORATIVE FILTERING

Collaborative filtering is very popular recommendation algorithm. The basic idea behind this algorithm works on past behaviour of user/users. It says that if some users like an item then they may share their interest and they may have same preferences for others items as well. Let's say user X and Y likes item A and user X likes item B also than probably user Y will also like item B. It means that if some

users have same preference and choices for an item, they will probably agree on choice for some other item also.

4.1 User-user Collaborative Filtering

User-user CF is very straight forward algorithm. It implies that, search for those users whose rating for an item is similar to active user and use their preferences on other items to recommend item to active user. To make a suggestion for user Y, user-user CF will look for users who have similar rating for other items as user Y and predict items to user Y that has being rated by those user but not user Y.

4.2 Item-item Collaborative Filtering

Item-item CF has proved as more scalable than user-user CF and is able to handle large user bases. It has deployed successfully on e-commerce sites like Amazon [6]. It uses the similarities between items for making recommendations. It is based on past behaviour of user and recommend items that are similar to that were liked by user in past. The basic idea behind Item-item CF is that if two items have same rating from some users, or have same features(e.g. romantic, thriller, comedy in case of movies) it means they are similar items and next time when a user like one item of those two then he may like the other item as well. Let's say item A,B and D are rated similarly by user X so now they are similar item, when user Y liked item B in past then he will get suggestions for item A and D.

5. PROPOSED SYSTEM

In this paper we are designing a system that recommend movies using combined approach of collaborative filtering including both user-user and item-item CF techniques. We are going to implement CF using Apache Mahout on the top of Hadoop to achieve scalability. We are using Movielense datasets provided for research on recommendation systems on its official website [3].

Proposed system works in two phases: In first phase, recommendations are obtained from user-user CF and item-item CF one by one and in second phase, obtained output from both CF techniques is combined as shown in fig-3.

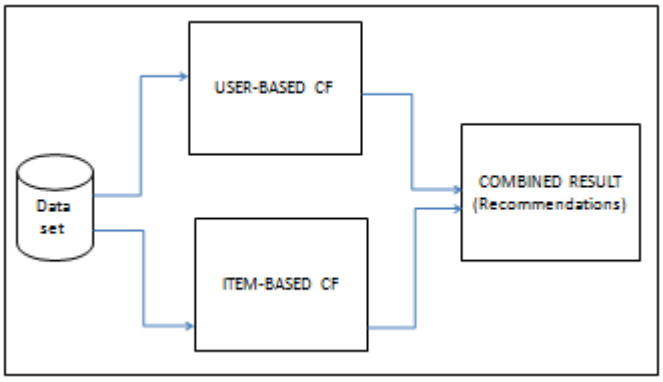

Fig -3: Combined approach for CF

In first phase, user-user CF is implemented on dataset and then item based CF is performed using Mahout. For user-user CF, user-item rating matrix is used and to find similarity between two users Pearson correlation coefficient is used. In item-item CF, past behavior of user (e.g. rating for past items) is used to find similar items and corresponding result has built. User-user CF, sometimes suffers from the problem of less nearest neighbor problem when preferences of current user for whom recommendations are building doesn't match any user then result of item-item CF can be helpful.

6. EXPERIMENTAL SETUP AND RESULT

In this section we elaborate our result analysis.

6.1 System Configuration

We implemented user-user and item based collaborative algorithm to get the recommendations on Haoop platform using Apache Mahout. Our Hadoop cluster is made up of four nodes out of which one is master and other are slave nodes.

Table – 1: System Configuration

Processor	2.00 GHz Intel Dual Core
RAM	3 GB
Operating System	Linux Ubuntu 14.04 LTS
Java	JRE 1.7
Hadoop	Apache Hadoop 2.6.0
Mahout	Apache Mahout 0.9
Dataset	Movie lens dataset

6.2 Result analysis

For experiment we have used stable benchmark dataset with 1,000,209 ratings of approximately 3,900 movies made by 6,040 MovieLens users. In this experiment we are comparing speedup of system with increasing numbers of nodes and increase of recommendation accuracy with increase in number of users. Speedup is given by the ratio of execution time of one processor and execution time with increasing number of nodes. Given by

$$\text{Speedup} = T(1)/T(n)$$

Where n = no. of nodes and T is execution time. T(1) represents times taken by single node and T(n) represents time taken by n numbers of nodes.

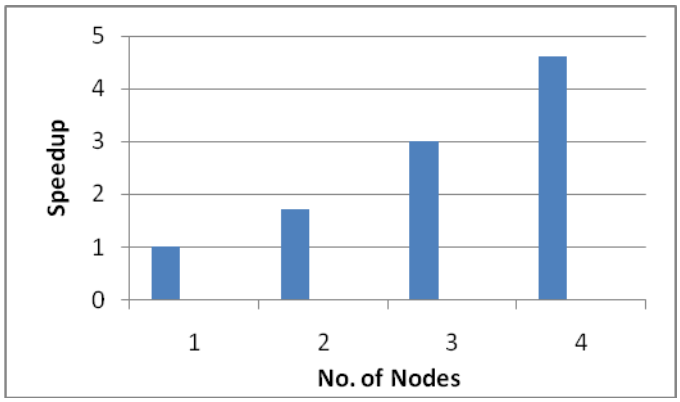

Chart – 1: Speedup with respect to no. of nodes

As we can see in chart – 1, we have taken four node cluster. When we increased the number of nodes one to four and with increasing number of nodes in Hadoop cluster, speedup also increases. So this algorithm is scaling well with Hadoop platform.

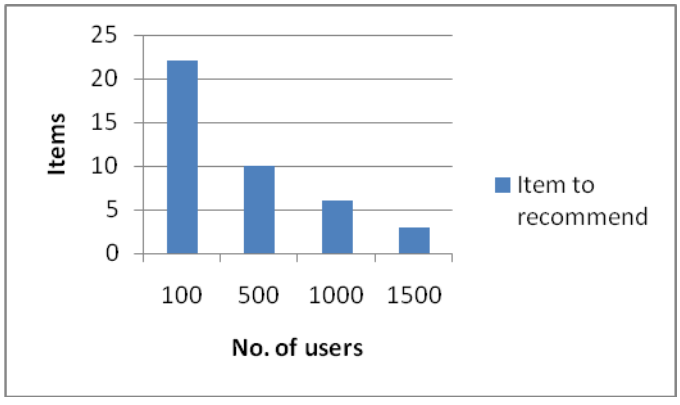

Chart – 2: Accuracy of recommendation

Next we have made four sets of data by taking different numbers of users as 100 users, 500 users, 100 users and 1500 users. As Chart – 2, shows the accuracy of recommendation has increased with respect to the increase in number of users who like an item because with increase in number of users the number of resulted recommendations are decreasing which shows less recommendations but with relevant result.

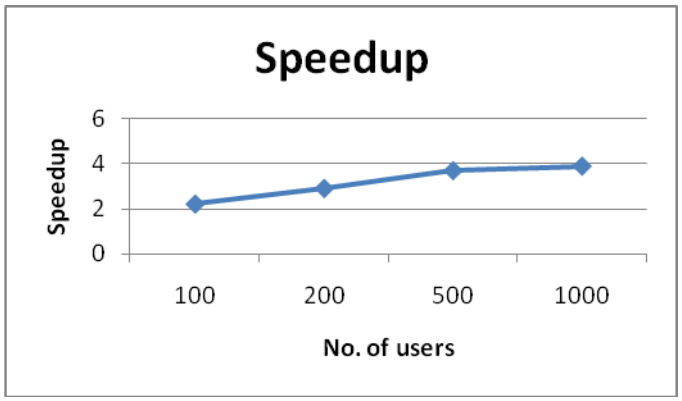

Chart – 3: Speedup with respect to increase in dataset.

Next we fixed the number of node to four and made 4 sets of users data. As shown in chart – 3, we have 100 users, 200 users, 500 users and 1000 users. We can see that as number of user increased in dataset the speed has also increased which shows that more the dataset more the speedup. We can see that to achieve the best performance by Hadoop we need to provide more data.

7. CONCLUSION

In this paper a combined approach of user-user CF and item-item CF has been presented to generate recommendations on Hadoop cluster using Apache Mahout, a library for machine learning algorithms. By using combined approach, accuracy of recommendation has improved. Using combined approach, accuracy of recommendation has improved. This approach has scaled well with the hadoop platform. Time needed to solve the problem has reduced. Mahout is able to handle big data but it still lack some algorithms. The recommendation for single user need to be improved for better results. . New computing platforms like Apache Spark are getting prominent in the field of Big Data analysis. Recommendation algorithms can be performed on such platforms for faster performance.

REFERENCES

[1] Apache Hadoop, https://hadoop.apache.org/
[2] Apache Mahout, http://mahout.apache.org/ .
[3] Movielens Dataset, http://grouplens.org/datasets/movielens/.
[4] D. Hee Park et al., "A literature review and classification of recommender systems research," Expert Systems with Applications 39 (2012) 10059–10072, Elsevier.
[5] X. Yang et al., "A survey of collaborative filtering based social recommender systems", Elsevier, Computer Communications 41 (2014) 1–10.
[6] G. Linden et al., "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1,pp. 76–80, 2003.
[7] Ruzhi Xu et al., "Distributed collaborative filtering with singular ratings for large scale recommendation", Elsevier, The Journal of Systems and Software 95 (2014) 231–241.
[8] J. Bobadilla et al. "Recommender systems survey", Elsevier, Knowledge-Based Systems 46 (2013) 109–132.
[9] Saikat Bagchi, "Performance and Quality Assessment of Similarity Measures in Collaborative Filtering Using Mahout", 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15), Procedia Computer Science 50 (2015) 229 – 234.
[10] Xiwei Wang et al., "A Case Study of Recommendation Algorithms", 2011 International Conference on Computational and Information Sciences, IEEE.
[11] Juha Leino and Kari-Jouko Räihä, "Case Amazon: Ratings and Review as Part of Recommendations", *RecSys'07*, October 19–20, 2007, ACM Press.
[12] Carlos E. Seminario and David C. Wilson, "Case Study of Mahout as a Recommender Platform" ACM *RecSys 2012.*
[13] Xing Xie, "Potential Friend Recommendations in Online Social Network" 2010 IEEE/ACM International Conference on Green Computing and Communications.
[14] G. Adomavicius and A. Tuzhilin. "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions", IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, June 2005.
[15] Ricci F. Et al., "Introduction to Recommender Systems Handbook" (http://www.inf.unibz.it /~ricci/papers/intro-rec-syshandbook.pdf), Recommender Systems Handbook, Springer, 2011.
[16] Owen S. Et al., "Mahout In Action", 2012. Manning Publications Co. ISBN 978-1-9351-8268-9.
[17] Wu Yueping and Zheng Jianguo, "A research of recommendation algorithm based on cloud model", IEEE 2010.
[18] Yanhong Guo et al., "An improved collaborative filtering algorithm based on trust in e-commerce recommendation system", IEEE 2010.
[19] B.M. Sarwar et al., "Item-item Collaborative Filtering Recommendation Algorithms," *10th Int'l World Wide Web Conference*, ACM Press, 2001, pp. 285-295.
[20] B.M. Sarwarm et al., "Analysis of Recommendation Algorithms for E-Commerce," *ACM Conf. Electronic Commerce*, ACM Press, 2000, pp.158-167.
[21] Jeffrey Dean and Sanjay Ghemawat, "Map-Reduce: Simplified data processing on large clusters", to appear in OSDI 2004.

