Overview Of Recommender System & A Speedy Approach In Collaborative Filtering Recommendation Algorithms With Mapreduce

[1,]Prof. Dr. R. Shankar , [2,] Prof. Ateshkumar Singh , [3,]Ms. Ila Naresh Patil

*[1,]IES College of Technology, [2,]M Tech. CSE * M Tech. CSE**

[1,]Bhopal, MP, India IES College of Technology, [2,] IES College of Technology,

[1,]Bhopal, MP, India [2],Bhopal, MP, India

ABSTRACT

Recommender systems suggest people items or services of their interest and proved to be an important solution to information overload problem. The big problem of collaborative filtering is its. In order to solve scalability problem, we can implement the Collaborative Filtering algorithm on the cloud computing platform using Hadoop's MapReduce. The work given here is focusing on the algorithm of recommendation mechanism for mobile commerce using combination of MapReduce and user based CF algorithm to overcome scalability. MapReduce is a programming model for expressing distributed computations on massive amounts of data and an execution framework for large-scale data processing on clusters of commodity servers. It was built on well-known principles in parallel and distributed processing.

KEYWORDS*: recommender system, collaborative filtering, speedup, partitioning, cloud-computing, hadoop, Map-Reduce.*

I. INTRODUCTION

Recommender systems provide an important response to the information overload problem as it presents users more practical and personalized information services. Three types of recommender systems: content-based recommender systems, collaborative recommender systems and trust based recommender system. Recommendation systems generate a ranked list of items on which a user might be interested. It is useful to approximate the degree to which specific user will like a specific product. The Recommender systems are useful in predicting the helpfulness of controversial reviews [1]. Recommender systems are a powerful new technology & help users to find items they want to buy from a business. Recommender systems are rapidly becoming a crucial tool in E-commerce on the Web.

In this paper we propose a new method that is implementing the user-based Collaborative Filtering algorithm on distributed implementation model, MapReduce model, on Hadoop platform to solve sparse data problem. The MapReduce model is inspired by the Lisp programming language map and reduces operations. Typically, the Map/Reduce framework and the Hadoop Distributed File System (HDFS Architecture) are running on the same set of nodes. This configuration allows the framework to effectively schedule tasks on the nodes where data is already present, resulting in very high aggregate bandwidth across the cluster.
The Map/Reduce framework consists of a single master JobTracker and one slave TaskTracker per cluster-node. The master is responsible for scheduling the jobs' component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves execute the tasks as directed by the master.This paper is organized as follows. Section 2 explains related work in commonly used recommendation strategies. Section 3 explains MapReduce model on Hadoop platform. Section 4 includes the experimental analysis & Section 5 concludes the paper.

II. RELATED WORK

2.1.Classification of Recommender Systems

2.2.Ontology-based Recommender System

In [2], the peer-to-peer network (P2P network) is based on decentralized architecture has the progress of ontology-based recommender system. This is basically works with dynamically changing large scale environment. In [3] a ontology-based multilayered semantic social network, is introduced. This model works on a set of users having similar interest and the correlation at different semantic levels.

2.3.Collaborative Tagging-based Recommender System

In [4], the collaborative tagging-based recommender allows users particularly consumers to freely connect tags or keywords to data contents. In [5] & [6] a generic model of collaborative tagging to recognize the dynamics behind it. The tag-based system suggests the use of high quality tags, by which spam and noise can be avoided.

2.4.Recommendation Methodologies

Basically there are three methods as content based, collaborative and trust based [7].

2.5.Content based Strategy

In [8], the Content Based (CB) method provides suggestions based on items similar to those that user has previously purchased or reviewed. It provides the recommendations based on the contents of documents & each user's preferences.

2.6.Collaborative Filtering Strategy

Collaborative filtering (CF) provides personalized recommendations based on the knowledge of similar users in the system. CF is focused on the principle that the finest recommendations for an individual are given by people who have similar interest. Collaborative filtering identifies users with choice similar to the target user and then built predictions based on the score of the neighbors. The job in collaborative filtering is to guess the usefulness of product to a particular user which is based on a database of user votes. The CF algorithms predicts ranking of a target item for target user with the help of ranking of the similar users that are known to item under consideration[9]. There are six collaborative filtering algorithms are evaluated. These algorithms accepts values for a interaction matrix A of order M x N = a_{ij} where M represents number of consumers ($C_1,C_2,C_3,\ldots\ldots C_M$) and N represents number of products or services ($P_1,P_2,P_3,\ldots\ldots P_N$). The value of a_{ij} varies based on transaction. The value of a_{ij} can be either 0 or 1. When $a_{ij} = 1$, means transaction between C_i & P_i (C_i has brought P_i). when $a_{ij} = 0$, means absence of transaction between C_i & Pi. The outcome of the algorithm is a list of probable ranked product for each consumer.

The user–based CF algorithm

This algorithm generates a list of recommendation of user interest in three steps. In first step, it searches N users in database which are similar to active user by creating customer similarity matrix $W_C = (Wc_{st})$. The high value of Wc_{st} indicates the consumers X & Y have similar liking as they have already brought many similar products .In second step, it calculates union of the items purchased by these users & link a respective weight with every item based on its significance in the set. Finally, in the third step, generates the list of recommended items & which have not already been brought by the active user. The resulting matrix will have element at C^{th} row & P^{th} column combine S the scores of the similarities between consumer C and other consumers who have purchased the product.

2.7.The item-based CF algorithm

This algorithm is based on the similar principal of user-based. The item-based algorithm determines product similarities instead of consumer similarity. It generates a product similarity matrix, $Wp = (Wp_{st})$ which is based on the column vectors of A. A high Wp_{st} shows that products X and Y are similar as many consumers have brought both of them. A WP will give the products' probable scores for each consumer. Resulting matrix will be containing the element at the C^{th} row and P^{th} column combines the scores of the similarities between product *P* and other products that consumer *C* has purchased. This algorithm provides higher efficiency and comparable or better recommendation quality than the user-based algorithm for many data sets [10].

2.8.The dimensionality-reduction algorithm

This algorithm compresses original interaction matrix & produce recommendations which are based on compressed, less-sparse matrix to simplify the scarcity problem. It applies standard singular-vector decomposition (SVD is a technique of matrix factorization) to decompose the interaction matrix A into U· Z · V' where U and V are two orthogonal matrices of size M x R and N x R respectively, and R is the rank of matrix A. Z is diagonal matrix of size R x R which has all singular values at its diagonal values. SVD can be used in recommender systems & has two features. SVD can be used to construct a low-dimensional image of customer-product space & calculates region in reduced space[11].

2.9.The generative-model algorithm

This algorithm approximates appropriate possibility & conditional probability. Based on estimated probability it creates score of product p for consumer c[12]. The approach is a statistical technique called as probabilistic Latent Semantic Analysis which was initially extended in the context of information retrieval. The method accomplishes competitive recommendation and calculation accuracies, is highly scalable, and extremely flexible.

2.10.The spreading-activation algorithm

This algorithm focuses on scarcity problem by discovering transitive associations between consumers & products by using bipartite consumer-product transitive-graph. The algorithm uses the graph to find out transitive connections [13]. It focuses on high quality recommendations when sufficient data is not available. The spreading activation algorithm consists of a series of nodes both user and item nodes. These nodes are then connected by edges where each weighted edge represents the ratings the item has received from the users. The higher the weight of the edge the higher the rating that item has received. The item nodes then send back "pulses" to the active user and their neighbors thus spreading the activation to the other nodes in the neighborhood of the active user.

2.11.The link-analysis algorithm

In this consumer-product graph, the global graph structure is used to help collaborative filtering under sparse data [14]. In this graph first set of nodes consists of products, services & information items for probable utilization. The second set consists of consumers or users. The feedback and transaction are represented as links connecting nodes between these two sets. This graph is referred as consumer-product graph. The link analysis algorithms such as HITS (Hypertext Induced Topic Selection) & PageRank are used for identifying essential web pages in a Web graph. The algorithm focuses on the extract helpful link structure details from the consumer-product graph & make more effective recommendation with sparse data.

2.12.Trust-Based Strategy

In Trust Based Recommendation systems, trust network is used in which users are joined by trust scores which indicate how much faith they have in each other. The user's trust network is constructed for generating predictions [15] & [16]. It has three steps. In first step, considers direct trust. The direct has two methods: explicitly or implicitly. The second step is propagation of trust. It is possible to propagate the trust i.e. create new relations among users. The third step is predicting ratings. From the trust network, we can predict what ratings the particular user would give for items.

2.13.MapReduce Collaborative Filtering Model

The conventional Collaborative Filtering consumes intensive computing time & computer resources especially when the dataset is very high. Here the MapReduce model is a distributed implementation model which is proposed by Google com. We introduce the MapReduce model and describe its working mechanism on Hadoop platform. The MapReduce model abstracts the calculation process into two core phases:

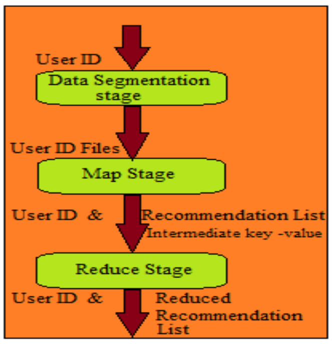

Figure1. Collaborative Filtering Using MapReduce

A. Data segmentation stage

In this phase[17], we separate the user ID into different files, in these files, each row store a user ID. These files as the input files of map phase, the data partitioning should satisfy 2 principles as follows:In the total running time, the proportion of computing time, the bigger the better. That means the most proportion of run time should be spent in the computation process, rather than frequently initialize the mapper. The same end of the tasks running time. That is the end of each mapper task time should be at the same time.

B. Map stage : Map stage : It is the function written by user which takes a set of input key/value pairs, and produces a set of output key/value pairs. In the Map, written by the user, takes a set of input key/value pairs, and produces a set of output key/value pairs. At this stage, the Hadoop platform estimate the algorithm's memory and others resources consumption, specified each DataNode the number of mapper it can be initialized. The Hadoop platform determines whether initialize a mapper to deal with the user ID files. If there have enough resources to initialize a mapper, the Hadoop platform initializes a new mapper. The mapper's setup function build the ratings-matrix between the users and the items at first, the mapper read the user ID file by line number, take the line number as the input key and this line corresponding user ID as the value. The next step is to calculate the similarity between this user and other users. The final step is to identify the user's nearest neighbors (by similarity values), and in accordance with the predict rating on items. We sort the predict ratings and store them in recommendation list. The user ID and its corresponding recommend-list as the intermediate key/value, output them to the reduce phase.Mapping creates a new output list by applying a function to individual elements of an input list[18].

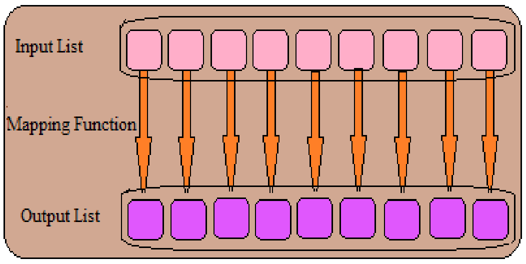

Figure2. Mapping

C. Reduce stage

It is also user defined function, accepts an intermediate key I and a set of values for that key. It merges together these values to form a possibly smaller set of values. Typically, none output value or only one is produced per Reduce invocation.In the reduce phase, the reducer collects the users ID and its corresponding recommend list, sort them according to user ID, and then output them to the HDFS in which the reducers are generated by hadoop platform.The computation takes a set of input key/value pairs, and produces a set of output key/value pairs. The user of the MapReduce library expresses the computation as two functions: map and reduce. Map, written by the user, takes an input pair and produces a set of intermediate key/value pairs[19]. The MapReduce library groups together all intermediate values associated with the same intermediate key I and passes them to the reduce function. The reduce function, also written by the user, accepts an intermediate key I and a set of values for that key. It merges these values together to form a possibly smaller set of values. Typically just zero or one output value is produced per reduce invocation. The intermediate values are supplied to the user's reduce function via an inter-mediator. This allows us to handle lists of values that are too large to fit in memory.

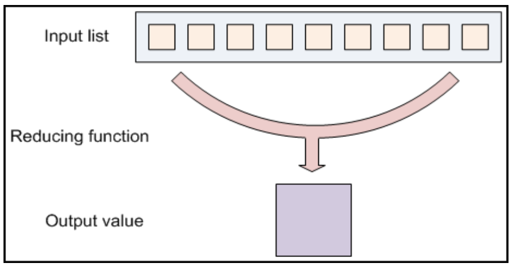

Figure3. Reducing

3.1 MapReduce framework for CF

Here, we present the implementation of the Collaborative-Filtering algorithm within the MapReduce framework. In [20], when we make recommendation, we would store the user ID which need to calculate in some txt files, then these files as the input of the Map function. The MapReduce framework initializes some mapper to deal with these user ID files. Our algorithm could be divided into the following stages, as shown in Figure 4:

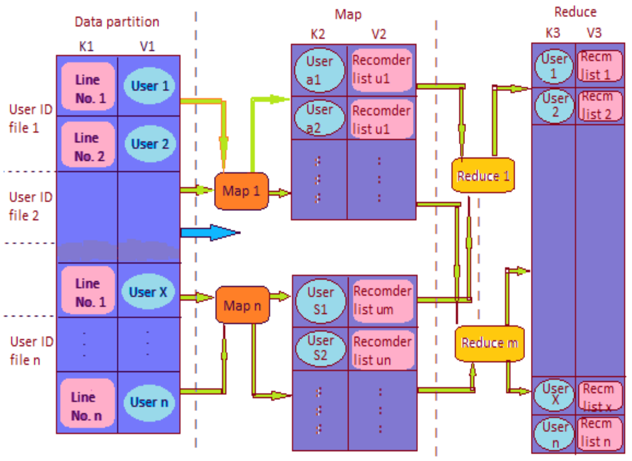

Figure 4. Collaborative Filtering with MapReduce

3.2.Experimental Analysis

We have implemented our experiments for CF algorithm on Java platform. As explained earlier in section 3, the Hadoop computer-cluster created on five computers. Here, we refer one of the computers as MainNode & remaining four as DataSetNodes. Each computer is having 4 GB RAM & Intel(R)core(TM) i5 CPU with 2.5GHz speed & Operating System Ubuntu 10.10. also the software used for the experiments are Hadoop MapReduce framework, Java JDK 1.6, the Mobile device (Android 3.0 & above), wireless Router are additional hardware we have used. The dataset is created by Netflix data set. The list of different movies is maintained in the dataset and more than 10,000 users. The users will define different ratings for each movie, not necessary the same rating. The role of our CF algorithm is to compare the runtime between standalone & Hadoop platform, so that we don't focus on accuracy. We take 3 copies of sub-datasets with 100 users, 200 users, 500 users & 1000 users. The DataSetNode is also divided into 2 nodes, 3 nodes, 5 nodes.

For the comparative analysis of standalone & Hadoop platform, we have considered average time $\mathbf{t_{avg}}$ as the Hadoop platform at current DataSetNode and the data set running time. Here the speedup is an important criteria to measure the efficiency of our algorithm. The speedup is given by,

Speedup = $\mathbf{t_{avg} / t_{sd1}}$

In our CF algorithm the recommendation is based on the division of each user theoretically, if we consider N nodes the speedup should be N. in other words, ideally the speedup should be linearly related to the number of DataSetNode. In the figure 5 we have shown the analytical result in graph which implies, increase in number of DataSetNodes, the speedup increases linearly. Also from the graph we can observe for 100 users, 200 users, the speedup is not linearly increase, this is because the data set is too small, thus the Hadoop platform is unable to demonstrate its efficiency.

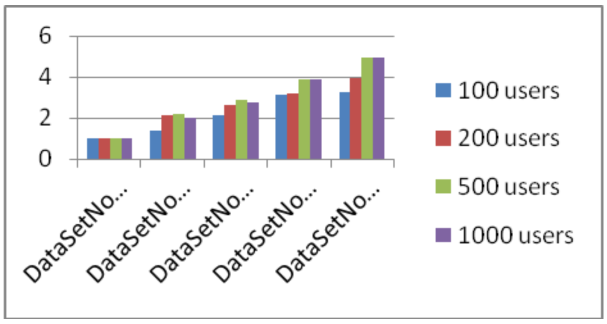

Figure 5. Speedup of CF of MapReduce

CONCLUSION

MapReduce can be used to parallelize Collaborative Filtering. We propose to apply this concept to Recommender System for the Web & Mobile Commerce as well.

REFERENCES :

[1] Patricia Victor and Chris Cornelis , Martine De Cock and Ankur M. Teredesai Trust- and Distrust-Based Recommendations for Controversial Reviews 15411672/11/$26.00 © 2011 IEEE INTELLIGENT SYSTEMS

[2] V. Diaz-Aviles, “Semantic peer-to-peer recommender systems,” M.S.thesis, Comput.-Based New Media Group, Inst. Comput. Sci., Albert Ludwigs Univ. Freiburg, Freiburg, Germany, 2005.

[3] I. Cantador and P. Castells, “Multilayered semantic social network modeling by ontology-based user profiles clustering: Application to collaborative filtering,” in Proc. Manag. Knowl. World Netw., 2006, pp. 334–349.

[4] S. Golder and B. Huberman,“The structure of collaborative tagging systems” in Proc. CoRR, 2005, pp.1–8.

[5] Z. Xu, Y. Fu, J. Mao, and D. Su, “Towards the semantic Web: Collaborative tag suggestions,” in Proc. CollaborativeWeb Tagging Workshop WWW, Edinburgh, U.K., 2006

[6] Shang Ming-Sheng, Zhang Zi-ke. Diffusion-Based Recommendation in Collaborative Tagging Systems. Chin. Phys. Lett.,2009,26(11): 118903

[7] Shang Ming-Sheng, Jin Ci-Hang, Zhou Tao, Zhang Yi-Cheng. Collaborative filtering based on multi-channel diffusion.Physica A: Statistical Mechanics and its Applications. 2009,388(23):4867-4871. M. D. Dikaiakos, D. Katsaros, G. Pallis, A.

[8] J.S. Breese, D. Heckerman, and C. Kadie, “Empirical Analysis of Predictive Algorithms for Collaborative Filtering,” Proc.14th Conf. Uncertainty in Artificial Intelligence (UAI 98), Morgan Kaufmann, 1998, pp. 43– 52.

[9] M. Deshpande and G. Karypis, “Item-Based Top-N

[10] Recommendation Algorithms,” ACM Trans. Information Systems, vol.22, no.1, 2004, pp. 143–177.

[11] B. Sarwar et al., “Application of Dimensionality Reduction in Recommender Systems: A Case Study,” Proc. WebKDD Workshop at the ACM SIGKDD, 2000; http://glaros.dtc.umn.edu/ gkhome/node/122

[12] T. Hofmann, “Latent Semantic Models for Collaborative Filtering,” ACM Trans. Information Systems, vol. 22, no. 1,2004,pp.89–115.

[13] Z. Huang, H. Chen, and D. Zeng, “Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering,” ACM Trans. Information Systems, vol. 22, no. 1, 2004, pp. 116–142

[14] Z. Huang, D. Zeng, and H. Chen, “A Link Analysis Approach to Recommendation under Sparse Data,” Proc. 2004 Americas Conf. Information Systems, 2004

[15] Kleinberg, J. Authoritative sources in a hyperlinked environment. in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (1998).

[16] Leonardo Zanette, Claudia L.R. Motta, Flávia Maria Santoro, Marcos Elia “A Trust-based Recommender System for Collaborative Networks” 2009 IEEE.

[17] [Zhili Wu, Xueli Yu and Jingyu “An Improved Trust Metric for Trust-aware Recommender Systems” 2009 IEEE.

[18] Vrba Z., Halvorsen P., Griwodz C., Beskow P. Kahn Process Networks are a Flexible Alternative to MapReduce”, High Performance Computing and Communications, 2009. HPCC ’09. 11th IEEE International Conference on ,25-27 June 2009.

[19] Dean J, Ghemawat S. Distributed programming with Mapreduce. In: Oram A, Wilson G, eds. Beautiful Code. Sebastopol: OReilly Media, Inc., 2007. 371 384.

[20] Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters. Communications of the ACM, 2005,51(1):107 113.

[21] Ranger C, Raghuraman R, Penmetsa A, Bradski G,Kozyrakis C. Evaluating MapReduce for Multi-core and Multiprocessor Systems. High Performance Computer Architecture, 2007. HPCA 2007. IEEE 13th International Symposium on,10-14 Feb. 2007.