1-1-2013

A personalized hybrid recommendation system oriented to e-commerce mass data in the cloud

Fang Dong
Southeast University

Junzhou Luo
Southeast University

Xia Zhu
Southeast University

Yuxiang Wang
Southeast University

Jun Shen
University of Wollongong, jshen@uow.edu.au

A personalized hybrid recommendation system oriented to e-commerce mass data in the cloud

Abstract

Personalized recommendation technology in Ecommerce is widespread to solve the problem of product information overload. However, with the further growth of the number of E-commerce users and products, the original recommendation algorithms and systems will face several new challenges: (1) to model user's interests more accurately; (2) to provide more diverse recommendation modes; and (3) to support large-scale expansion. To address these challenges, from the actual demands of E-commerce applications (as Made-in-China website), a personalized hybrid recommendation system, which can support massive data set, is designed and implemented in this paper by using Cloud technology. Hereinto, the recommendation algorithms are designed based on a novel user interesting model for different scenarios; and the massive data parallel processing techniques in Cloud computing is utilized to realize the effective execution of recommendation algorithms. Finally, several experiments are presented to highlight the system performance.

Keywords

e, commerce, oriented, personalized, cloud, system, mass, recommendation, data, hybrid

Disciplines

Engineering | Science and Technology Studies

Publication Details

Dong, F., Luo, J., Zhu, X., Wang, Y. & Shen, J. (2013). A personalized hybrid recommendation system oriented to e-commerce mass data in the cloud. IEEE International Conference on Systems, Man and Cybernetics (pp. 1020-1025). IEEE Xplore: IEEE SMC.

A Personalized Hybrid Recommendation System Oriented to E-Commerce Mass Data in the Cloud

Fang Dong, Junzhou Luo, Xia Zhu, Yuxiang Wang
School of Computer Science and Engineering
Southeast University
Nanjing, P.R. China
{fdong, jluo, xzhu, lsswyx }@seu.edu.cn

Jun Shen
School of Information Systems and Technology
University of Wollongong
Wollongong NSW 2522, Australia
jshen@uow.edu.au

***Abstract*—Personalized recommendation technology, as an important method for information filtering, can effectively solve the problem of information overload of Internet. It has become the core technology of E-commerce applications. However, with the further growth of the number of E-commerce users and products, the original recommendation algorithms and systems will face many new challenges: (1) to model user's interests more accurately; (2) to provide more diverse recommendation modes; and (3) to support large-scale expansion. To address these challenges, from the actual demands of E-commerce applications (as Made-in-China website), a personalized hybrid recommendation system, which can support massive data set, is designed and implemented in this paper by using Cloud technology. There are three parts of this paper, the first part is to introduce the recommendation algorithms which are designed for different demands; In the second part, the massive data parallel processing techniques in Cloud computing is utilized to realize the effective execution of recommendation algorithms; At last, the real personalized hybrid recommendation system and relevant algorithms have been implemented and deployed upon SEUCloud Platform, then several experiments are presented to highlight the system performance.**

Keywords- E-Commerce; Personlized Recommendation; Cloud; Massive Data

I. INTRODUCTION

With the rapid spread of Internet, E-commerce is booming with an incredible rate. In the recent years, several famous E-commerce websites are sprung up: such as Amazon, eBay and TaoBao(in China) in B2C field, and Kompass, Thomasnet, and Made-in-Chin [1] (in China) in B2B field. However, the overload of product information becomes more and more severe in E-commerce. It is quite difficult to find the products what users really need from a large number of products.

Thus, how to learn as much as possible about the interests of the consumer, in order to facilitate consumer shopping, become key issue problems for the development of E-commerce. The personalized recommendation technology for E-commerce has come into existence for a few years already[1]. The main concept of personalized recommendation is to extract the characteristics, and potential preferences, of consumers according to the relevant online browsing behaviors and purchasing records, and then to recommend proper products to the consumers.

Meanwhile, in industry, personalized recommendation has become the core technology for E-commerce online video and other Internet applications. The typical systems include: the book recommendation system of Amazon [2]; the movie recommendation system of Netflix [3] and video recommendation system of YouTube [4] etc..

However, with the further growth of the number of E-commerce users and products, the interests and needs of users, as well as the amount of data of recommendation system have undergone great changes: users have more diverse interests and more personalized demands, meanwhile the amount of data from recommendation system grows rapidly. In such situation, the original recommendation algorithms and systems will face several new challenges:

(1) Need to describe consumer's interest more accurately

The consumer's interests will change with the lapse of time. However, as the existing recommendation strategies could not take into consideration the visit time of historical records, they cannot make certain responses in time when consumer's current interests are changing (such as visiting some products recently). Thus, it will lead to a great difference between recommended resources and consumer's actual current interests. Therefore, a new model of consumer's interest is needed to support dynamic update and furthermore to improve the recommendation accuracy rate.

(2) Need to provide more diverse recommendation modes

By analyzing the business logic of Made-in-China and TaoBao, we observe that the websites usually include several types of Webpages (take Made-in-China as example, it includes user login page, product page, product search page and inquiry page), each of them corresponds to different functions and logic. As for the sight of consumer, different page browse behaviors usually correspond to different demands. However, the existing recommendation systems usually apply single mode algorithm (such as collaborative filtering)[1], they

[1] Made-in-China has more than 8 million members, and ranks the first three in the field of B2B E-commerce in China. Its owner Focus Technology Co., Ltd (Stock code: 002315) has a long-term partnership with Southeast University. And a Joint R & D Center with E-commerce and Cloud computing has been established. The site address is : www.made-in-china.com.

can only meet consumer's demand partially, and cannot satisfy consumer's diversity demand. Therefore, the personalized recommendation system is required to provide more diverse services according to different needs from the consumers.

(3) Need to support large-scale expansion of recommendation

Nowadays, the data of consumers browsing, commodity trading, consumer rating, and system log have made explosive growth (take TaoBao as example, the data amount of daily increase is over 0.1PB, and the overall amount of data reaches 28PB). In such case, to analyze and process such massive data needs to consume a lot of computing power and storage space. If we still use the traditional centralized processing mode, it will result in very long response time, and will be unable to meet consumer's real-time requirements, which greatly affects the shopping experience. As a new distributed computing mode, cloud computing integrates massive distributed resources to construct a shared resource pool by using virtualization techniques for providing on-demand computational power. Cloud computing becomes a very popular technique to achieve massive data processing[5]. Therefore, utilizing Cloud computing to speed up the execution of recommendation becomes the effective solution to solve the problems mentioned above.

Based on these challenges, more and more companies plan to improve their traditional recommendation systems. Made-in-China is one of them, so there is a cooperation project between Focus Technology[1] Company and our lab to design a new recommendation system which can satisfy their demands.

From the actual demands of E-commerce website (as Made-in-China), a personalized hybrid recommendation system which can support massive data set is designed and implemented by using cloud technology. In order to provide a variety of recommendation services, the relevant recommendation algorithms are designed for different webpages respectively. Herein, with the consideration of product's multi-dimension attributes, consumer's rating information and product visiting timestamp of consumer, a tree structure for the modeling of consumer's interest is defined to reflect consumer's actual interests accurately and timely. Then, the massive data parallel processing techniques as used in cloud computing is utilized to realize the effective execution of relevant large-scale recommendation algorithms. At last, the real personalized hybrid recommendation system has been developed and deployed upon SEUCloud (Southeast University Cloud) Platform. Several experiments and system usage are presented to demonstrate the performance of relevant recommendation algorithms and the overall system, where all the massive data including consumers, products and visit behaviors were derived from the real data of Made-in-China.

II. RELATED WORKS

In recent years, the research on recommender technologies has attracted increasing attention due to the "information overload" problem caused by the rapid development of information technology. There are many companies having designed their own recommendation system to support their Web applications, such as the Google news recommendation [6], FOFs system of Facebook [7] and the music recommender of Yahoo! [8], etc.

Among these typically systems, we note that the collaborative filtering (CF) is the commonly used recommender technology. And there are many researches having been carried out to improve the different aspects of CF. For example, the papers [9-10] are focused on the sparsity issue of CF, in which Wang et al. [9] proposed a unifying user-based and item-based approach by similarity fusion, and Sarwar et al. [10] proposed a Latent Semantic Indexing (LSI) to reduce the dimension space and increase the data density, making the user similarity much more obviously. On the other hand, Mehta et al. [11] discussed the attack resistance and trust issue of CF algorithms. And many other new CF recommenders such as Bayesian network-based [12], Horting graph-based [13] and item-based [14] technologies and algorithms are proposed to improve the accuracy and performance. But they did not consider the dynamic changes in consumer's interests.

On the other hand, all the above-mentioned methods and algorithms are centralized so that they cannot satisfy the scalable requirement of massive data affiliated E-commerce. In order to overcome this problem, a few researchers proposed several approaches which concentrate on CF algorithms. For example, Liu et al. [15] proposes a P2P-based hybrid CF mechanism to combine user-based and item-based ratings. Clustering process of the mechanism is required to move all ratings to local client and its calculation complexity is $O(n^2)$. Zhen et al. [16] propose a novel model of distributed knowledge recommender system to maintain two sets for every peer: source peers set which records M peers whose similarity is the most similar with the active peer; destination peers set which records N peers where active peer is regarded as source peer. Since the model initialization is required to search the whole network, the model is inefficient while the network size is large.

To face today's new challenges as we identified earlier, the existing mechanisms also have some other limitations such as 1) the current model of consumer interest cannot effectively reflect the change of consumers' interest for products, 2) lack of a hybrid framework for different demands and logics, and 3) current distributed algorithms cannot support massive data processing.

III. THE ARCHITECTUE OF OUR SYSTEM

In this section, we describe the overall architecture of our recommendation system depicted in Figure 1. In order to respond to the consumers' recommendation requirements in a real-time mode, we divided the system into two parts: NEARLINE and ONLINE.

The function of NEARLINE component is to calculate recommendation information from the massive raw data periodically with the accumulation of historical data (which can be seen as a nearline background data processing). The procedure of NEARLINE is as follows: Firstly, NEARLINE needs a preprocessing to extract the useful data from raw data and store them in HDFS. Then, two intermediate calculations are needed to produce some summary information, which are the input data for background recommendation algorithm.

Finally, the background recommendation algorithm will be accelerated based on MapReduce to produce the similarity matrix, such as user and product similarity. Then it chooses some valuable information to store them in database. Such operation executes periodically (30mins) to guarantee the effectiveness and availability of our recommendation information.

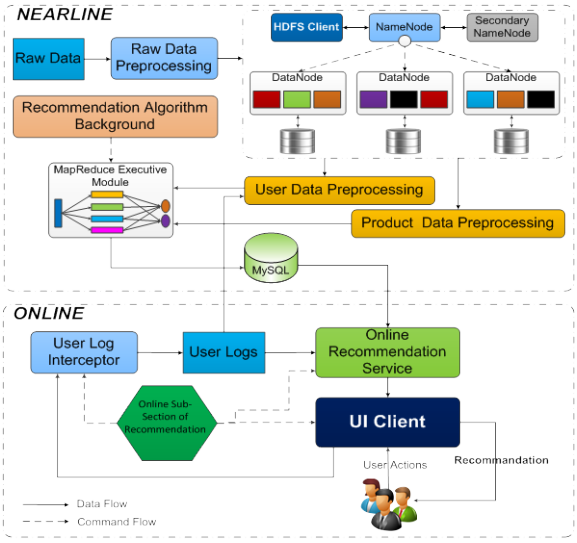


Figure 1. The architecture of Hybrid Recommendation System

The function of ONLINE component is to generate and present the final recommendation for the active consumers immediately (which can be seen as an online recommendation feedback). Given an active consumer, the ONLINE component processes as: (1) capture the current consumer's behavior and record them in logs, (2) calculate partial recommendation based on active consumer's real-time logs and the recommendation information from database server, and (3) return the final recommendation through online recommendation service.

In the next part, the algorithms and parallel processing mechanisms that were adopted in our recommendation system will be discussed in details.

IV. THE PERSONALIZED HYBRID RECOMMENDATION MECHANISM

By analyzing the business logic of the Made-in-China platform, it consists mainly of four aspects: user login, product browse, and product search and inquiry basket. And from the consumer perspective, different page views behaviors usually correspond to different demands. Thus, in order to satisfy the diverse demands of consumers, the more personalized hybrid recommendation mechanism should be designed and implemented with different business logics.

(1) For user login page: consumer has no specific behaviors yet. The demand of consumer is to get a comprehensive product recommendation result. Thus, the similar neighbors of consumer (with the similar interests) are utilized to generate abundant recommendation results.

(2) For product browse page: the demand of consumer is to obtain the recommendation contents which are similar with this product. Thus, the similarity between products is utilized to generate recommendation results.

(3) For product search page: the demand of consumer is to get the hot products which are related to query keywords. Thus, the keyword based hot product recommendation is considered.

(4) For inquiry basket page: the demand of consumer is to get the products which have inter-relationship with the chosen products and are likely to be purchased in the future. Thus, the sequence pattern mining concept is considered to extract the potential needs of consumers to generate recommendation.

Based on the analysis mentioned above, four separate recommendation algorithms are proposed respectively. Due to limited space of paper, the recommendation algorithms about user login and product browse will be highlighted.

A. UPT-based Similar User Collaborative Recommendation

The user login recommendation is based on similar neighbor concept which belongs to the collaborative recommendation mechanism, where 'user' and 'consumer' denote the same meaning in the whole paper.

The core process of collaborative product recommendation is to find k neighbors which have the similar interests as the target user and then to make a recommendation, where the user interest model plays the key role in this kind of recommendation algorithm. At present, most of the existing user interest description methods are based on vector-space model. However, with the gradual refinement of the classification of products in the e-commerce platform, only considering the rough classification of product only can not accurately reflect the multidimensional information of product. On the other hand, most of the existing user interest modeling methods ignores the characteristics of dynamic changes of user's interest over time. Based on these considerations, in order to correctly reflect the user's interests in products, the field classification vector CV and the Interest Energy(IE) are defined respectively. And then we will propose a tree structure for the modeling of user's interest, called User Preference Tree(UPT).

Definition 1. The field classification vector of product is defined as $CV=<(CVK_1,CW_1),(CVK_2, CW_2),\ldots,(CVK_m,CW_m)>$, where CVK_x denotes the x-th dimensional attribute's name of product, CW_x denotes the relevant weight value. The attributes of certain product P_j can be defined as $CA(P_j) = <cvk_1,cvk_2,\ldots,cvk_m>$, where cvk_x denotes the x-th dimension attribute's value of P_j. For example, CV = <(First Categories, 0.4), (Secondary Categories, 0.3), (Brand, 0.2), (Style, 0.1)>, $CA(P_j)$= <Clothes, Jacket(Man), Adidas, Black >.

Definition 2. Interest Energy (IE): for user U_i's **visited** (including purchasing/inquiry, browsing, etc.) product P_j, $IE_{i,j}$ denotes the interest degree of U_i to P_j at current time.

Rule 1. The design rule of interest energy attenuation function $IE_{attenuation}$ can be defined as follows:

(1) User's interested products in future will be similar to recently visited products, thus these products should be endowed with a larger IE value;

(2) In order to avoid the effect of occasional product visit to long-term visit rule, IE of recent visited resource should not be sustained in large value for a long time;

(3) The previous long-term preference of user (continuous product visit with same type) will still affect the product recommendation;

(4) In each product visit behavior, purchasing/ inquiry makes more important impacts onto user interest than browse behavior.

Then, the negative exponential function is used to denote the preference energy attenuation function as follows:

$$IE_{attenuation}(x) = ke^{-\lambda(x-1)} \qquad x \geq 1 \qquad (1)$$

Therein, x denotes the visit order of U_i, $\lambda \in (0,1)$ is the attenuation parameter, k is used to distinguish the visit type , the k value of purchase behavior is larger than browse behavior.

Definition 3. User Preference Tree (UPT): U_i's UPT is defined as a (|CV|+1)-depth tree. The leaf node which represents a visited product of U_i is defined as five-tuple UPT_{leaf}= {PID, IE, IW, CR, level}, where PID denotes product ID, IW denotes the interest weight of certain product, CR denotes the rating of U_i to certain product. The non-leaf node is defined as a four-tuple as $UPT_{non\text{-}leaf}$= {cvk, IW, CR, level}. And there is:

(1) The IW of non-leaf node t can be defined as follows, where s denotes each node, $Pred(t,s)$ denotes whether t is the predecessor of leaf node s.

$$IW_{it} = \frac{\sum_s (IE_{is} \bullet Pred(t,s))}{\sum_s IE_{is}} \qquad (2)$$

(2) The CR of non-leaf node t can be defined as follows, where CR_{is} denotes the rating of U_i to P_s.

$$CR_{it} = \frac{\sum_s (CR_{is} \bullet Pred(t,s))}{\sum_s Pred(t,s)} \qquad (3)$$

A four-level UPT is shown in Figure 2, where a visited product P_j uniquely corresponds to a path from root to corresponding leaf node, where each keyword corresponds to the relevant attribute of product P_j.

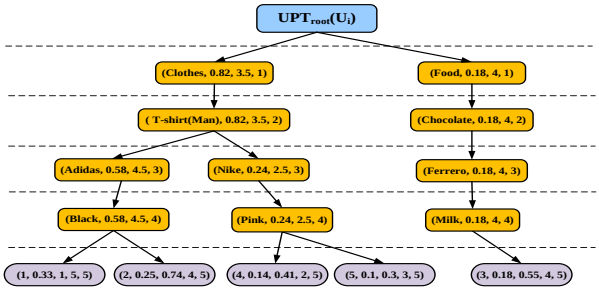


Figure 2. User preference tree based on a four-dimensional field classification

Our proposal is to integrate the user rating, relevant attributes of visited products and dynamic interest energy based on user's UPT to solve the sparsity rating problem and to increase the accuracy of the calculation of similarity.

Rule 2. In order to enhance the flexibility of similarity calculation, if there is some common attributes of two users' visited products, they can still be judged as similar neighbors even if there are no identical visited products of these two users. Therefore, the UPT based similarity calculation rule between two users can be described as follows:

(1) The more similar the attributes of U_i and U_q's visited products, the larger similarity between them will be;

(2) The more similar the dynamic preference weight with same domain of U_i and U_q, the larger similarity will be;

(3) The more similar the rating data with same domain of U_i and U_q, the larger similarity between them will be.

As described in rule 2, the similarity between two users can be calculated based on the Intersection Subtree of two UPT(called ISU), the details can be defined as follows:

Definition 4. Intersection Sub-tree of two UPTs (as ISU): for U_i and U_q, $ISU(U_i,U_q)$ denotes the maximum connected intersection between $UPD(U_i)$ and $UPD(U_q)$ with same node's keyword. The generating of ISU can be shown in Figure 3 (the serial numbers denote the relevant matching order):

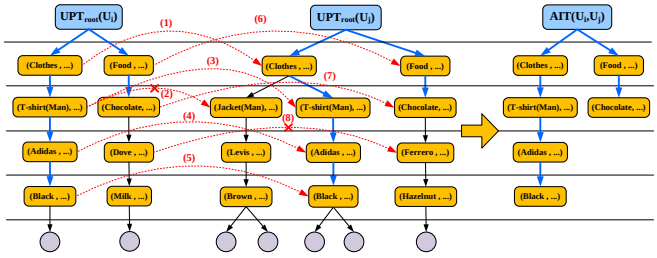


Figure 3. The generation process of ISU

Then, the calculation of similarity between two users can be divided into two aspects: interest weight based similarity and user rating based similarity.

Difinition 5. The interest weight based similarity Sim_{IW} reflects the similarity between U_i and U_q's dynamic interests and preferences. According to IW of each matching node on $UPT(U_i)$ and $UPT(U_q)$ which correspond to each node u on $ISU(U_i,U_q)$ (denoted as $MN_u(U_i)$ and $MN_u(U_q)$), the calculation of $Sim_{IW}(U_i,U_q)$ can be defined as formula 4. MW_k is the matching weight of the k-th layer of ISU.

$$Sim_{IW}(U_i,U_q) = \frac{\sum_{u \in ISU(U_i,U_q)} MW_{u.level} \bullet MN_u(U_i).IW \bullet MN_u(U_q).IW}{\sqrt{\sum_{s \in UPT(U_i)} MW_{s.level} \bullet s.IW^2} \bullet \sqrt{\sum_{t \in IPT(U_q)} MW_{t.level} \bullet t.IW^2}} \qquad (4)$$

Definition 6. The user rating based similarity Sim_{CR} reflects the similarity between rating vectors of U_i and U_q. In order to overcome sparsity rating problem, Sim_{CR} needs to compute the similarity between CR values of each matching node on $UPT(U_i)$ and $UPT(U_q)$ which correspond to each leaf node on $ISU(U_i,U_q)$. The calculation of $Sim_{CR}(U_i,U_q)$ is defined as formula 5, where L denotes the leaf nodes set of ISU, $\overline{CR_i}$ and $\overline{CR_q}$ denote the mean value of U_i and U_q's rating data.

$$Sim_{CR}(U_i,U_q) = \frac{\sum_{l \in L} |(MN_l(U_i).CR - \overline{CR_i}) \bullet (MN_l(U_q).CR - \overline{CR_q})|}{\sqrt{\sum_{l \in L} (MN_l(U_i).CR - \overline{CR_i})^2} \bullet \sqrt{\sum_{l \in L} (MN_l(U_q).CR - \overline{CR_q})^2}} \qquad (5)$$

The similarity between U_i and U_q is calculated as follows, where Nor is the normalized function.

$$Sim(U_i, U_q) = \alpha \bullet Nor(Sim_{IW}(U_i, U_q)) + (1-\alpha) \bullet Nor(Sim_{CR}(U_i, U_q)) \quad (6)$$

Based on formula 6, the similarity between U_i and other users can be obtained. Then the top-k users with the highest similarity are chosen to construct the similarity user set of U_i (called $SU_k(U_i)$). Assumed that the visited product set of each user in $SU_k(U_i)$ are PS_1, PS_2, …,PS_k respectively, the candidate products set for recommendation can be denoted as $CP(U_i) = PS_1 \cup PS_2 \cup \cdots \cup PS_k$. For each product P_j in CP, the relevant degree between U_i and P_j can be denoted in formula 7, and then we chose top-k products with highest RD value for recommendation in this application scenario.

$$RD(U_i, P_j) = \frac{\sum_{U_s \in SU_k(U_i)} Sim(U_i, U_s) \bullet IW(U_s, P_j)}{\sum_{U_s \in SU_k(U_i)} Sim(U_i, U_s)} \quad (7)$$

B. Product Similarity based Recommendation

As consumers in the stage of product page have not yet purchased, product page recommendation algorithm should calculate the similarity between different products and recommend the products which show the highest similarity to the current product. There are 7 factors influencing recommendation results: category, consumer, keyword, purchase flow, seller, district and popularity, where category, consumer and keyword are the three most important factors.

Category: Assume that the height of the product category tree is *n*. Every leaf node represents a product in the tree. Similarity between product *i* and product *j* is represented by:

$$CS(i, j) = e^{-\frac{(n-1-k)^2}{n}} \quad (8)$$

where *k* denotes the depth of common parent node of product *i* and product *j*. And there are: when $k=n-1$, which means two products are identical, $CS(i, j)= 1$; when $k=0$, which means two products are completely different, $CS(i, j)=e^{-n}\approx 0$.

Consumer: We assume that if two products have more common consumers, the similarity of these two products is higher. An alternative way of computing the similarity between each pair of products *i* and *j* accordingly is to use a measure value that is based on the conditional probability-based similarity. Formula 9 records common times of purchasing/ inquiring one of the products given that the other has already been purchased. Hereinto, Freq(*x*) and Freq(x,y) denote the number of consumers who have purchased the products *x* or both of *x* and *y*. α is a parameter that takes a value between 0 and 1 to avoid the situation that the hot products might get inflated similarity.

$$BS(i, j) = \frac{Freq(i, j)}{Freq(i) \times (Freq(j))^{\alpha}} \quad (9)$$

Keyword: If two products have more than one identical keyword (all keywords are input by sellers in Made-in-China), the similarity of two products will be higher. Similarity between i and j according to keyword is defined as follows.

$$KS(i, j) = \frac{2 \bullet |key(i) \cup key(j)|}{|key(i)| + |key(j)|} \quad (10)$$

The above three factors are more important for recommendation results and the following four are secondary factors, and for the reference of recommendation only.

Purchase flow: Purchase flow means that if the consumer purchased the products *a*, then he will also purchase products *b* associated with *a*. And there may be several sequential patterns between these behaviors, which will be discussed in section D. The similarity between *i* and *j* with purchase flow is:

$$FS(i, j) = \begin{cases} 0 & i \text{ and } j \notin \text{any rules sequential pattern set} \\ 1 & \text{otherwise} \end{cases} \quad (11)$$

Seller and District: Similarity between *i* and *j* according to seller and district can both be defined as boolean function:

$$SS(i, j) = \begin{cases} 1 & seller(i) = seller(j) \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

$$DS(i, j) = \begin{cases} 1 & district(i) = district(j) \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

Popularity: Similarity between *i* and *j* with popularity is:

$$PS(j) = \frac{Freq(j)}{max(Freq(t))} \quad (14)$$

where max(Freq(*t*)) denotes the max value of consumer numbers that have purchased.

In conclusion, the similarity between product *i* and *j* is:

$$Sim(i, j) = \partial_1 CA(i, j) + \partial_2 BU(i, j) + \partial_3 KW(i, j) + \partial_4 PF(i, j) + \partial_5 SE(i, j) + \partial_6 DI(i, j) + \partial_7 PO(j) \quad (15)$$

where ∂_t is from 0 and 1 and $\sum_{t=1}^{7} \partial_t = 1$.

C. Product Popularity based Recommendation

The main concept of recommendation in searching page is to focus on the products which are related to the consumer's query keywords. In this case, consumer usually wants to know which products are the favorite products with the same category as query keywords. The main phases of this recommendation are: (1) the hot products (with more inquiry time) of each category are counted in a nearline way; (2) find the maximum matching catalog according to the query keywords; (3) find the products which belong to this catalog and other brother-catalogs, then the top-n products with larger popularity and similarity will be recommended to consumer.

D. Sequential Pattern Mining based Recommendation

In E-commerce environments, the purchase or inquiry processes of consumers (displayed at inquiry basket page) usually have some temporal-dependency relationship, which may reflect consumer's potential behavior pattern and preference. Thus, in order to further improve the performance of recommendation, the relevant product purchase/inquiry sequential patterns need to be mined according to consumer's historical records. Then, the most probable products which will be purchased or inquired in near future can be predicted according to consumer's recent purchase/inquiry products.

The main recommendation phase can be described as follows: (1) The purchase/inquiry product records and the corresponding product categories of all consumer's UPT are preprocessed to form a transaction set; (2) the sequential patterns are mined based on transaction set to discover the potential relationship among individual categories and products, then generation the sequential pattern set (such as $X \Rightarrow Y$); (3) search the set with products in inquiry basket as prefix, and then generate the relevant recommendation results with additionally considering the popularity of products.

V. The Speedup of Recommendation Algorithms with Map-Reduce Framwork

With the great explosion of the number of products and visit logs of consumers, the efficiency and scalability of recommendations are facing a great challenge. In this case, if we still use the traditional centralized processing mode, the consumer's requirement could not be satisfied (for TB-level dataset, the response time of recommendation may be up to several hours). Therefore, in order to greatly reduce the recommendation response time, we adopt MapReduce paradigm in Cloud computing[17] to realize the parallel processing of time-consuming part of the hybrid recommendation mechanism mentioned in section IV. As the time complexity of the recommendation algorithm in product search page is relatively low, this algorithm as well as the sequential pattern mining procedure in inquiry basket page can be executed in an offline way, herein only the parallel processing methods of user similarity and products similarity calculation are proposed.

A. The Parallel Processing of user similarity calculation

MapReduce technique can be utilized to parallelize the calculation of user similarity. The details are described as follows, which contains four processes (as shown in Figure 4 and Table 1).

(1) To scan all the nodes in UPT as the source file, where each row represents the information of a certain node;

(2) During the **first** Map-Reduce phase, take UserID as the key and node information as value for mapping, and calculate the mean value of CR ($\overline{CR}$) of certain nodes which are located at same level, then take the results as the input of phase 2;

(3) During the **second** phase, take cvk as the key and left part of information as value for mapping and calculation. Then take ID of two users as key, and take PW, CR, $\overline{CR}$, level as the input value of next phase;

(4) During the **third** phase, find the nodes of two different users with same cvk values, and then calculate the part of similarity of each two users with all level of UPT respectively according to relevant formula. And take two UserID as the key and the partial similarity and level as the input value of next phase;

(5) During the **fourth** phase, calculate the finally similarity of each two users according to the formula 4-6, then take two UserID as the key and the final similarity of each two users as result to output.

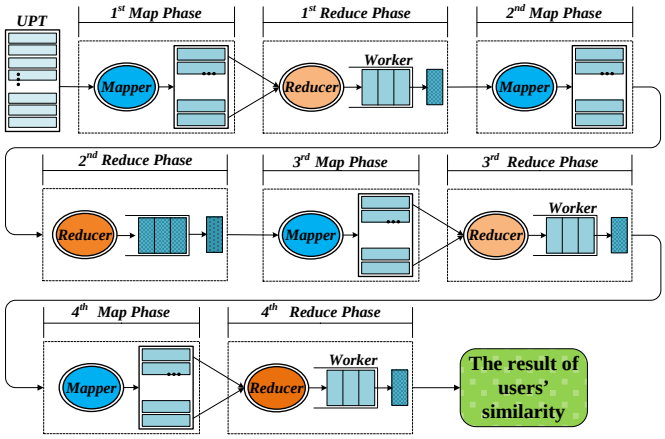


Figure 4. The MapReduce Procedure of User Similarity Calculation

TABLE 1. THE MEANING OF KEY-VALUE IN ALL MAPREDUCE PHASE

	Map		Reduce	
	key	value	key	value
1st	UserID	Node Msg	UserID	Update Node Msg
2nd	cvk(PID)	Other Node Msg	UserID1+ UserID2	cvk+PW+ CR +level
3rd	2nd Reduce key	2nd Reduce value	UserID1+ UserID2	Part of Sim with level
4th	3rd Reduce key	3rd Reduce value	UserID1+ UserID2	Sim(U_i,U_j)

B. The Parallel Processing of product similarity calculation

MapReduce programming model can also be utilized to parallelize the calculation of product similarity. The details are described as follows, which contains three Map-Reduce processes (as shown in Figure 5 and Table 2).

(1) To scan all the products' information as the source file, where each row represents the information of certain product. Then 7 separated Map-Reduce streams (the number of attributes) are executed for different attributes of products to calculate the similarity of each dimension;

(2) During the first Map-Reduce phase of certain stream, take the relevant attribute as the key and product ID as value for mapping, then the products with same attribute value will be put together. Then generate the records about each two products with the same attribute value, where take these two product ID as key and 1as value for output;

(3) During the second phase, take the outputs from each attribute bounded Map-Reduce as the input, then all the key-value pairs with same key (two product ID pair) will be merged and the relevant values will be added together;

(4) During the last phase, take certain two product ID pair as key and the number of common attributes as the value for map. Then, calculate all the similarity for each common attribute according to formula 8-15. At last, the finally similarity for each two products can be obtained by summing each common attribute's similarity.

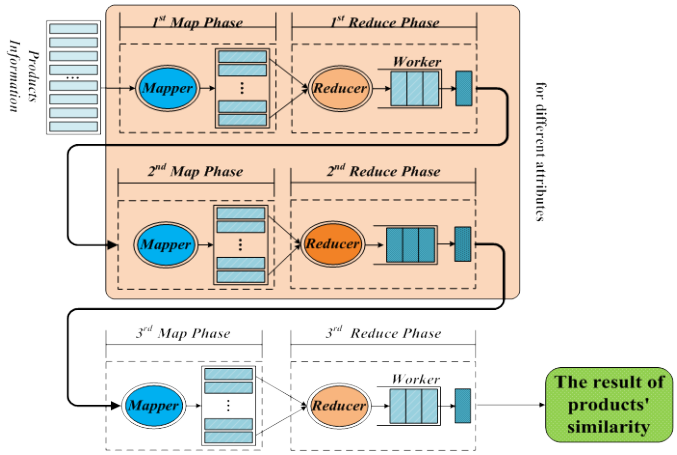


Figure 5. The MapReduce Procedure of Product Similarity Calculation

TABLE 2. THE MEANING OF KEY-VALUE IN ALL MAPREDUCE PHASE

	Map		Reduce	
	key	value	key	value
1st	cvk	PID	PID1+ PID2	1+flag
2nd	1ST Reduce key	1ST Reduce value	PID1+ PID2	Sum of value+flag
3rd	1ST&2nd Reduce key	1ST&2nd Reduce value	PID1+ PID2	Sim (i,j)

VI. SYSTEM IMPLEMENTATION AND PERFORMANCE EVALUATION

A. *The Implementation of Recommendation System*

Based on the system architecture, the different recommendation algorithms and relevant parallel processing mechanisms as mentioned above, the personalized hybrid recommendation system which can support massive data has been developed and deployed upon SEUCloud Platform with real demands of Made-in-China.

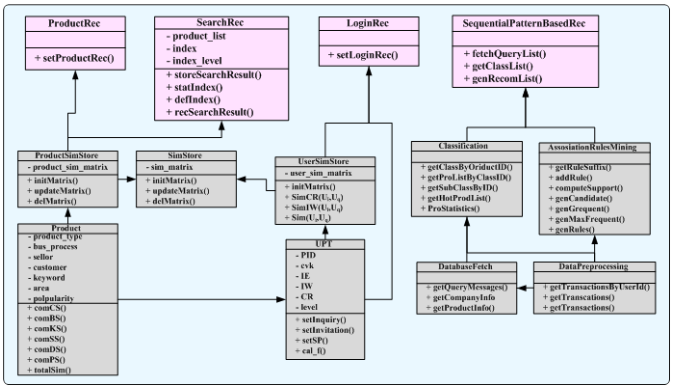


Figure 6. The Class Diagram of Recommendation Algorithms

The relevant UML diagram of recommendation algorithms is shown in Figure 6. And the NEARLINE and ONLINE parts are developed based on hadoop 0.20.0, jdk 1.7.0 (for NEARLINE) and MySQL 5.5.29, jdk 1.7.0 with Spring, Struts, MyBatis and Quartz framework (for ONLINE) respectively. Herein, the NEARLINE part mainly includes three modules: data storage and access, main part of recommendation algorithms and parallel processing. And the ONLINE part mainly includes: web front-end user interface, MySQL database operation, data acquisition and processing

The real deployment environment is based on SEUCloud. SEUCloud mainly consists of compute module and storage module. The compute module contains 252 Blade Servers, eight 4-Way Rack Servers and two 8-Way Rack Servers, offering 3.7 TFlops of computing power. The storage module is set up by IBM DS5300 storage array with 500TB HD via 8Gpbs Fiber Channels. Each of these compute and storage nodes has a separate 1Gb/s Ethernet and 40Gb/s Infiniband link respectively to different switches. Through the campus network, researchers from Southeast University can enter the login node to use computer resources. The architecture is depicted by Figure 7.

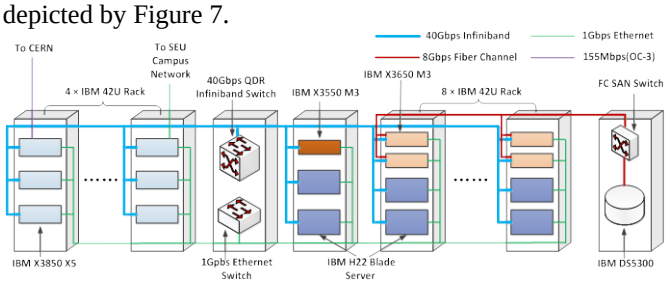


Figure 7. The Hardware Architecture of SEUCloud

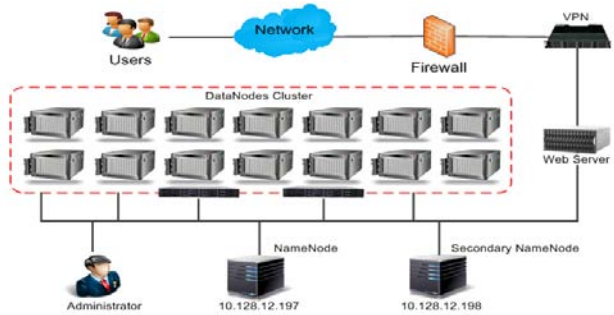


Figure 8. The System Deployment Architecture

We have applied 14 blade servers (168 cores) and 3 rack servers from SEUCloud to deploy the recommendation system, where the Data-Node modules of hadoop are deployed on each blade servers for parallel processing whose IP addresses are from 10.128.12.199 to 10.128.12.212; the Name-Node and the secondary Name-Node are deployed on two rack servers (with IP addresses 10.128.12.197 and 10.128.12.198); the web server for ONLINE part is deployed onto another rack server (10.128.12.213). The detailed real deployment is shown in Figure 8.

Furthermore, we have stored about 0.8 TB E-commerce data onto SEUCloud (in HDFS), which includes about 506490 consumers, 964290 products, 1095050 browse records and 492740 inquiry records.

B. *The Performance Evaluation*

In this section, the performance of proposed recommendation algorithms and relevant mechansims will be verified and evaluated based on the real recommendation system with real data from Made-in-China mentioned above. Due to limited space of this paper. We mainly focus on UPT-based Similar User Collaborative Recommendation (in user login page) and Product Similarity based Recommendation (in product browse page). The experiments can be divided into two parts: the **performance evaluation of recommendation**

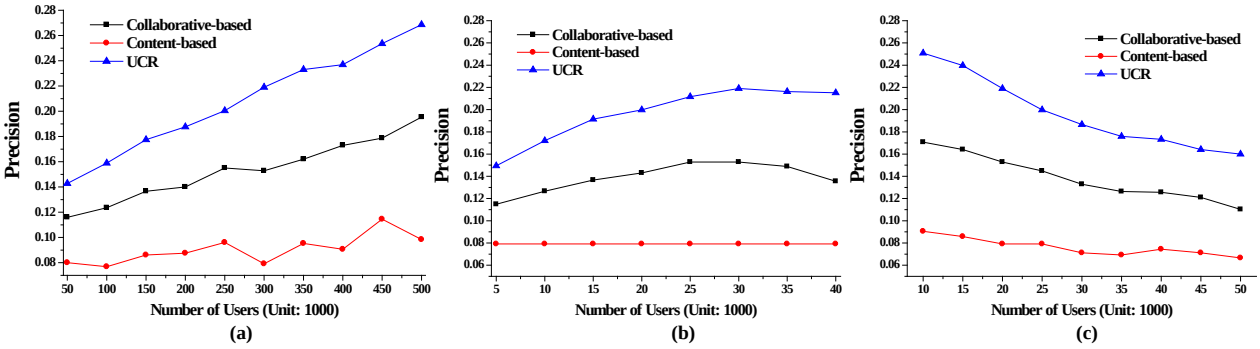


Figure 9. The Comparison of Three Recommendation Algorithms' Performance

algorithms themselves and **the evaluation of the execution performance of recommendation algorithms**.

The relevant evaluation metrics include two aspects: Precision for recommendation performance and Speedup for recommendation execution performance, which can be defined as formula 16 and 17, where P denotes the number of users, $Top\text{-}N(U_i)$ denotes the recommendation products set for U_i, $|Testing \cap Top\text{-}N(U_i)|$ denotes the number of correct recommendation. $ET_{parallel}$ denotes the execution time of parallel processing and ET_{single} denotes the execution time of centralized processing.

$$Precision = \frac{\sum_{i=1}^{P} |Testing \cap Top\text{-}N(U_i)|}{P \bullet |Top\text{-}N(U_i)|} \quad (16)$$

$$Speedup = \frac{ET_{parallel}}{ET_{single}} \quad (17)$$

1) The evaluation of recommendation performance

The proposed recommendation algorithms are compared with content-based recommendation algorithm [18], traditional collaborative-based recommendation algorithm [1] with user number P, the similarity user number K and recommendation product number N changing. As Product Similarity based Recommendation algorithm(PCR) is customized according to the acutal request of Made-in-China for a certain page, there may be no related and sophisticated recommendation algorithms to be compared. Therefore, only UPT-based Similar User Collaborative Recommendation algorithm(UCR) is considered for global recommendation in this part.

The first comparison is to compare the precision of three recommendation algorithms with respect to P while N=20 and K=30. As shown in Figure 9(a), with the increase of P, the precision of each algorithm is increasing except content-based algorithm, where UCR always obtain better performance. This is because that when P is small, as users' information can not be utilized efficiently, the collaborative based recommendation cannot find effective similar users. With P increasing, as more users' information can be used, the performance of collaborative based mechanism will be enhanced gradually. As UCR can model the consumer interest more accurately (by considering dynamic interest changing), it will always lead to a better result.

The second comparison is to compare the precision of three recommendation algorithms with respect to K while N=20 and P = 300000. As shown in Figure 9(b), when K is limited in a certain value range, with the increasing of K, the precision of each algorithm is increasing except content-based algorithm. When K reaches to a certain value, with increasing of K, the precision of each algorithm is decreasing, especially for collaborative-based algorithm. Moreover during the change of K, UCR always produces better performance. The reason is that when K increases to a certain value, since several dissimilar users may be denoted as similar users by collaborative-based algorithm, the recommendation accuracy will decrease. But in UCR, as UPT can reflect user's interest much more accurately, as it can find more real similar users meanwhile it will set a threshold in the similar users calculation process to guarantee the quality of them.

The third comparison simulation is to compare the precision of three recommendation algorithms with respect to N while P = 300000 and K = 30. As shown in Figure 9(c), with the increasing of N, the precision of each algorithm is decreasing. And moreover, UCR always produces better performance, especially when N is small. It is because that during the changing process, according to formula 17, the numerator and denominator will increase synchronously, but denominator gets the higher increasing rate. And for the same reason, UCR can get better results.

2) The evaluation of execution performance

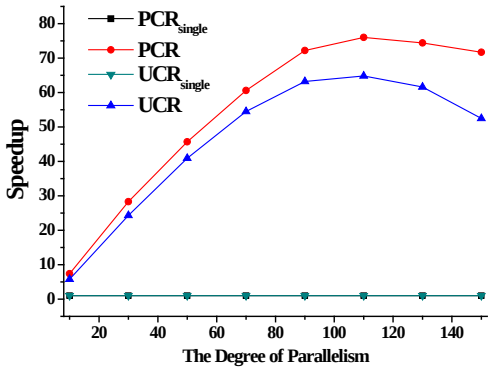
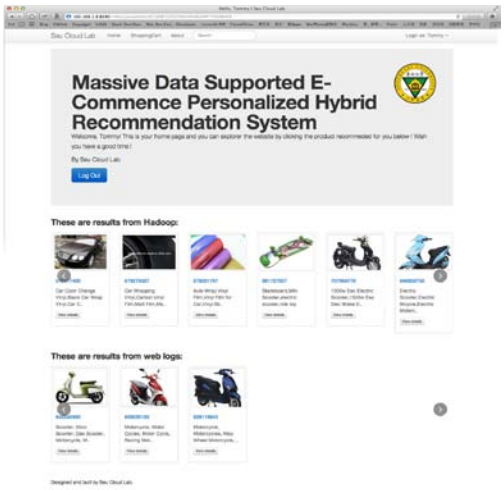
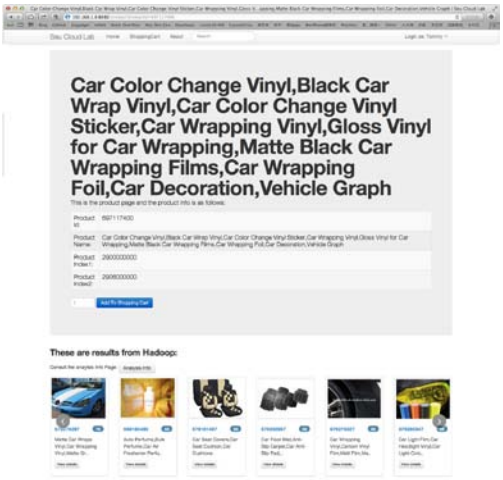


Figure 10. The Execution Performance of UCR and PCR

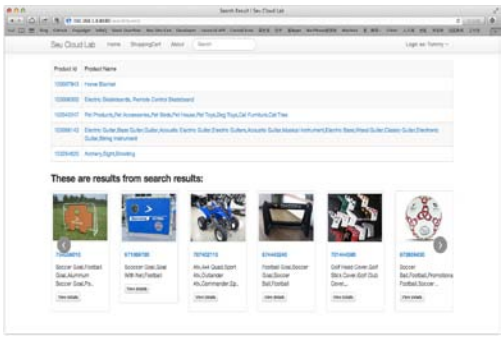
The execution performance with parallel and centralized processing about UCR and PCR is compared in this part with various degree of parallelism (as the number of Maps) while P



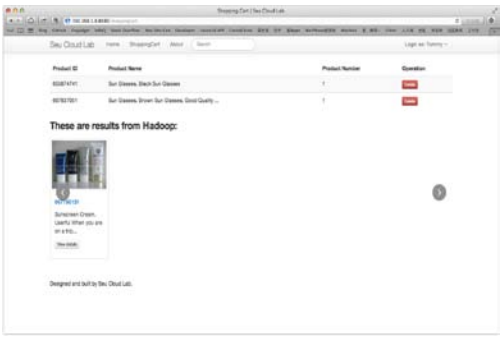
(a) User Login Page



(b) Product Page



(c) Product Search Page



(d) Inquiry Basket Page

Figure 11. The Usage Demonstration of All Pages in Recommendation System

= 300000, N=20 and K=30. As shown in Figure 10, with the increase of PD, the Speedup of UCR and PCR are also increasing. But when PD exceeds a certain value, the Speedup begins to decline instead of increasing. The reason is that, when PD is small, as the algorithms are mainly executed in a relevant centralized way, the execution time will be much higher and will lead to lower Speedup. With PD increasing, the algorithms will be divided into more sub-tasks for parallel processing. When PD is too much large (even if less than the total number of cores), the too much Map-Reduce processes will bring up the overhead of systems (such as shared memory and hard disk I/O channel).

C. Usage Demonstration

In this part, we will show how the actual system works. We have not yet integrated system with Made-in-China, so the system with relevant simple page (but complete function) is given. The overall operations can be illuminated as follows:

(1) Users enter the URL (192.168.1.8:8080), and after enter the correct user name and password, the user will be redirected to Home page.

(2) At Homepage, in order to show the nearline and online part of system clearly, the recommendation results (from UCR) are divided into two parts: The first part is from Hadoop recommended results (nearline part): according to the user ID, query the recommendation information from MySQL table, and then generate the recommendation. The second part is to get similar user based on user ID and log information table through the Web module, to generate the online recommendation (online part).

(3) The user can click on the product link to the product page. Then at product page, the part information of product is shown on the top. And the below part is the recommendation results, where the product similarity result generated from Hadoop (nearline part) has stored into MySQL, then the recommendation result can be generated by searching product ID in MySQL(online part). At the same time, after user enters the product page, the user ID and product ID will be record.

(4) At search page, users can search by keywords. At the bottom of page, the recommendation result with product popularity is shown.

(5) Users can add products into inquiry basket at product page. And at inquiry basket page, the sequential patterns have been stored into MySql, and the recommendation result can be generate by searching the Inquiry rules table

from MySQL according to the sequence of products in inquiry basket.

VII. Conslusion and Future Works

In this paper, a personalized hybrid recommendation system which can support massive data set is designed and implemented. Herein, several recommendation algorithms are designed for different webpages of actual E-commerce application to satisfy user's diverse demands. The execution process of recommendation algorithms can be speedup by using MapReduce. The real system is developed and deployed onto SEUCloud Platform, where the experiments and system displays demonstrate the system performance.

From this paper, we find that: during the parallel processing phase, the execution of MapReduce can be further optimized. Therefore, in the future, we will mainly focus on the performance optimization of MapReduce procedure. On the other hand, the vitalization techniques should be considered to support elastic resource provision especially when the user access varies dynamically.

Acknowledgment

This work is supported by National Key Basic Research Program of China under Grants No. 2010CB328104, National Natural Science Foundation of China under Grants No.61070161, No.61003257, No.61202449, No.61272054, China National High Technology Research and Development Program No.2013AA013503, China National Key Technology R&D Program under Grants No.2010BAI88B03 and No.2011BAK21B02, China Specialized Research Fund for the Doctoral Program of Higher Education under Grants No.20110092130002, China National Science and Technology Major Project under Grants No. 2010ZX01044-001-001, Jiangsu Provincial Natural Science Foundation of China under Grants No. BK2008030, Jiangsu Provincial Perspective Production and Research Alliance under Grants No. BY2012202, Jiangsu Provincial Key Laboratory of Network and Information Security under Grants No. BM2003201, Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grants No. 93K-9, and Shanghai Key Laboratory of Scalable Computing and Systems under Grants No.2010DS680095.

References

[1] Z. Huang, D. Zeng and H. Chen. A Comparison of Collaborative-Filtering Recommendation Algorithms for E-commerce. IEEE Intelligent Systems 22 (2007), 68–78.

[2] URL:www.amazon.com.

[3] URL:www.netflix.com.

[4] URL:www.youtube.com.

[5] M. Armbrust, A. Fox, R. Griffith, et al. Above the Clouds: A Berkeley View of Cloud Computing[J]. Commun. ACM. 2010, 53: 50-58.

[6] J. Liu, P. Dolan, E. Pedersen. Personalized news recommendation based on click behavior. Proceedings of the 15th international conference on intelligent user interfaces. ACM, 2010: 31-40.

[7] L. Backstrom. Dealing with structured and unstructured Data at Facebook. The 8th Extended Semantic Web Conference ESCW 2011.

[8] Aizenberg N, Koren Y, Somekh O. Build your own music recommender by modeling internet radio streams. Proceedings of the 21st international conference on World Wide Web. ACM, 2012: 1-10.

[9] J. Wang, AP. Vries, MJT. Reinders. Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion. In: Proc. of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '06).

[10] Sarwar, B.M., Karypis, G., Konstan, J.A., et al. Application of dimensionality reduction in recommender system——a case study. In: Jhingran, A., Mason, J.M., Tygar, D., eds. Proceedings of the ACM WebKDD Workshop on Web Mining for E-Commerce. New York: ACM Press, 2000.

[11] B. Mehta, W. Nejdl. Attack resistant collaborative filtering. In: Proc. of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '08).

[12] Breese, J.S., Heckerman, D., Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. In: Cooper, G.F., Moral, S., eds. Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence. San Francisco: Morgan Kaufmann Publishers, 1998. 43~52.

[13] Aggarwal, C.C., Wolf, J.L., Wu, K., et al. Horting hatches an egg: a new raph-theoretic approach to collaborative filtering. In: Chaudhuri, S., Madigan, D., Fayyad, U., eds. Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 1999. 201~212.

[14] Sarwar, B., Karypis, G., Konstan, J., et al. Item-Based collaborative filtering recommendation algorithms. In: Shen, V.Y., Saito, N., eds. Proceedings of the 10th International World Wide Web Conference (WWW10). 2001. 285~295.

[15] ZB. Liu, WY. Qu, HT. Li, CS. Xie. A hybrid collaborative filtering recommendation mechanism for P2P networks. Future Generation Computer Systems, Volume 26, Issue 8, 2010:1409-1417.

[16] L. Zhen, ZH. Jiang, HT. Song. Distributed recommender for peer-to-peer knowledge sharing. Information Sciences, Volume 180, Issue 18, 2010: 3546-3561.

[17] Jeff Dean and Sanjay Ghemawat. MapReduce: Simplified data processing on large clusters. In Proceedings of the 6th Symposium on Operating Systems Design and Implementation (OSDI): 137–150, December 2004.

[18] Liren Chen and Katia Sycara. WebMate: A Personal Agent for Browsing and Searching [C]. Proceedings of the Second International Conference on Autonomous Agents, 1998.