

# Tourism Information Pushing System based on Intelligent Recommendation

Bai Xiaojun

School of Computer Science and Engineering  
Xi'an Technological University  
Xi'an, China  
bxjem@163.com

Zhang Xiaoxu

School of Computer Science and Engineering  
Xi'an Technological University  
Xi'an, China  
757688085@qq.com

Zhang Shuyi

School of Computer Science and Engineering  
Xi'an Technological University  
Xi'an, China  
light\_blue\_green@qq.com@qq.com

He Lei

School of Computer Science and Engineering  
Xi'an Technological University  
Xi'an, China  
htime@qq.com

**Abstract**—User usually retrieve information of interest via search engines, but for a mobile client, the user experience of manually search will be poor, for it is inefficient, and may bring huge data traffic for mobile user. To provide a convenient method for mobile users to obtain tourism information, putting forward a tourism information pushing system based on intelligent recommendation. The item based collaborative filtering rules and the location based recommendation algorithm were introduced to generate recommendation items, and the XMPP based pushing technology was used to push these items to mobile client. By these means, the tourism information will be searched and filtered intelligently according to user's profile, and pushed to end user automatically, thus provide customized recommendation service for each user, and improve user experience of tourism service platform. To solve the problem of huge data and huge calculating, Map-Reduce optimized algorithm were discussed to improve the efficiency of recommendation creation. Introduced the total structure of this system, the design of each component, as well as the detailed implementation of recommendation algorithm. experiments proved the efficiency of this system.

**Keywords**—Tourism service; Intelligent recommendation; Collaborative filtering; Location based recommendation; Map-Reduce

## I. INTRODUCTION

With the development of mobile network, Internet users are gradually shifting from PC client to mobile client, the number of mobile Internet users raised sharply in recent years, while, more and more client applications were designed to make convenience for mobile users. User usually retrieve information of interest through search engine on traditional platform, thus the user experience was poor while search manually on mobile platform, firstly it is inefficient, secondly, the mobile client is sensitive to network flow and energy consumption. So for mobile users, it will be better to generate recommendation items on server side, and push them to the mobile clients.

In this paper, the author will discuss of building an intelligent tourism information pushing system, introduce intelligent recommendation technology into tourism service platform, make it an automatic search engine, which will generate tourism recommendations for users in accordance with their preference and there location, and push these recommendations to users automatically, it will eliminate the trouble of searching and filtering from large amount of information manually.

## II. TOTAL STRUCTURE OF THIS SYSTEM

The whole system is made up of three components: App Server, Push Server, and mobile Client(App Client), as shown in Fig .1.

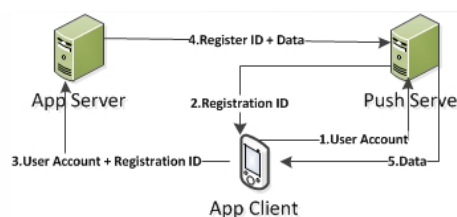


Figure 1 Total structure of this system

App Server is the core component of this system, It will maintain basic information of tourism items, users and their evaluations, meanwhile, it will process requests from App clients, get location of users, generate recommendation items for user, and then push these items to user via push server.

When the application on a mobile client startup, it will firstly register itself to the push server, meanwhile, the push server will generate a Registration\_ID for the mobile client; Then, App client will send user information as well as Registration\_ID to App server; based on the received profile, the application server automatically query tourism information database to generates recommendation, then send the recommendation and the Registration\_ID to the push server; Finally, the push server will find the mobile client by its Registration\_ID, and send the recommendation items to it.

### III. DESIGN OF EACH COMPONENT

#### A. The mobile client

The client will regularly report its location to the application server, meanwhile, receive recommend items from the push server and display them, the structure of mobile client is shown in Fig. 2.

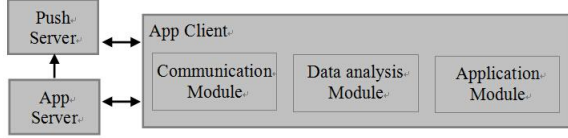


Figure 2 Structure of mobile client

The communication module is responsible for message communication with push server and app server, more over, it will maintain the connection with push server.

For app server, the client need to establish an active connection periodically, report its location and Registration\_ID on this connection, these data will be send via an HTTP request.

For push server, the client need to establish a connection and listen on it, when receive a notification from the push server, execute some preprocessing on it, and then pass it to the next module. For there is a restriction on TCP connection of GPRS network, that the priority of a connection will be decreased till disconnected when there is no data transfer on it for a long time, so as to process push messages in time, the client need to keep the connection via sending heartbeat messages at regular interval.

The data that push to the client is organized in JSON format, which is simple and easy for process, for example:

[{"title": "hello", "message": "welcome to the ancient city of xi'an"}]

This JSON string means that: the title of this message is "hello", and the content is "welcome to the ancient city of xi'an".

When received such a string, the data analysis module will break it into value objects, and passed them to the application module to display.

#### B. The push server

The structure of push server is shown in Fig. 3.

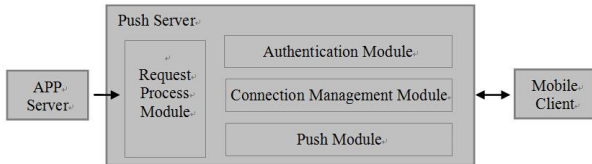


Figure 3: structure of push server

Push server authenticate connection for each client, Create session for each authenticated client, thus recommendation items are pushed on this session. The connection management module is designed to maintain persistent connections between push server and mobile clients, to make sure that each connection would keep active.

When received a HTTP request from the App server, the request process module will handle it, obtain

parameters from it, include recommendation content and Registration\_ID of mobile client, thus call the push module to send this message to the corresponding client.

#### C. The application server

The structure of application server is shown in Fig. 4.

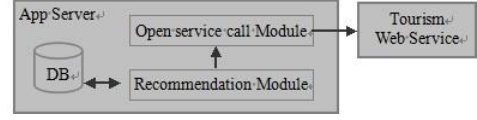


Figure 4: structure of application server

The application server will generate recommendations based on the user's preference as well as their location. Whatever to say, the local resource on application server is limited, in order to provide more comprehensive information, other tourism web service should be integrated into this system. For example, to push the weather condition of a tourism destination, it is not needed to develop a weather forecast system, we just need to select a suitable weather service website, and call its open service to obtain the weather information.

### IV. THE INTELLIGENT RECOMMENDATION ALGORITHM

The recommendation algorithm is the key element to design in application server, the item based collaborative filtering algorithm and location based recommendation algorithm is adopted here to generate recommendations.

#### A. Item based collaborative filtering algorithm

##### 1) Group the users data and tourism items data.

Set the grouped users as  $U = \{u_1, u_2, u_3, \dots, u_M\}$ , in which M stands for the number of users in this group; set the grouped tourism items as  $S = \{s_1, s_2, s_3, \dots, s_N\}$ , in which N stands for the number of items in this group. After visite a tourism item, the user will make a score and remark on this item, thus an evaluation matrix can be established based on these scores, as shown in formula 1.

$$RS_{M \times N} = \begin{bmatrix} r_{s_1 u_1} & r_{s_2 u_1} & \dots & r_{s_N u_1} \\ r_{s_1 u_2} & r_{s_2 u_2} & \dots & r_{s_N u_2} \\ \vdots & \vdots & \ddots & \vdots \\ r_{s_1 u_M} & r_{s_2 u_M} & \dots & r_{s_N u_M} \end{bmatrix} \quad (1)$$

##### 2) Calculate the similarity between user $u_i$ and $u_j$ .

The Person similarity algorithm will be used here, as shown in formula 2, where  $r_{s u_i}$  means score of user i to item k, and  $\bar{r}_{s u_i}$  means average score of user i to all items.

$$\text{sim}(u_i, u_j) = \frac{\sum_{s_k \in US_{u_i, u_j}} (r_{s u_i} - \bar{r}_{s u_i})(r_{s u_j} - \bar{r}_{s u_j})}{\sqrt{\sum_{s_k \in US_{u_i, u_j}} (r_{s u_i} - \bar{r}_{s u_i})^2} \sqrt{\sum_{s_k \in US_{u_i, u_j}} (r_{s u_j} - \bar{r}_{s u_j})^2}} \quad (2)$$

In this formula,  $US_{u_i, u_j}$ , which stands for the joint items that marked by both  $u_i$  and  $u_j$ , is very important, the more greater of this value, the more accurate the result will be.

That means, if  $\text{sim}(u_i, u_j) = \text{sim}(u_i, u_k)$ , but  $|US_{u_i, u_j}| > |US_{u_i, u_k}|$ , then the similarity between  $u_i$  and  $u_j$  will be more accurate

than that between  $u_i$  and  $u_k$ . Formula 2 did not consider this, So revise it to formula 3.

$$sim'(u_i, u_j) = \frac{1}{1 + e^{\frac{|US_{ui}, u_j|}{2}}} \times sim(u_i, u_j) \quad (3)$$

3) Calculate the prediction mark for items that did not marked by a user, as formula 4.

$$P_{u_i, s_j} = \bar{r}_{S_{u_i}} + \frac{\sum_{u_p \in S_{u_i}} (r_{S_{u_p}, s_j} - \bar{r}_{S_{u_p}}) \times sim'(u_i, u_p)}{\sum_{u_p \in S_{u_i}} sim'(u_i, u_p)} \quad (4)$$

In this formula,  $\bar{r}_{S_{u_i}}$  stands for the average mark of user  $i$  to all items,  $su$  stands for the set that include all similar users of the target user  $u_i$ ,  $u_p$  means any user in the set  $su$ .

After calculated, sort all items that has a predict value on descending order, and pick from the header of the list as recommendations, then push them to the mobile client.

### B. Location based recommendation algorithm

After the processing of item based collaborative filtering, the generated recommendation list maybe still big, thus we will execute a more accurate filter on the list based on location of the user.

1) Represent the historical mark of a user into 5-tuple set, as {user, item, user\_location, item\_location, mark}.

2) Decide a threshold value, then pick up all items that have a mark greater than the value.

3) According to the picked 5-tuple sets, calculate the distance between user location and item location by the Euclidean distance formula, and construct a distance array, as {hd1, hd2, ..., hdm}.

4) Pick the recommend items come from item based collaborative filtering, calculate the distance between current location of the user and item location of each recommendation, and construct the current distance array, as {d1, d2, ..., dn}.

5) Construct a 2-dimension array as following:

$$P_{oo_{hd_m, d_n}} = \begin{bmatrix} hd_1 - d_1 & hd_1 - d_2 & \dots & hd_1 - d_n \\ hd_2 - d_1 & hd_2 - d_2 & \dots & hd_2 - d_n \\ hd_3 - d_1 & hd_3 - d_2 & \dots & hd_3 - d_n \\ \dots & \dots & \dots & \dots \\ hd_m - d_1 & hd_m - d_2 & \dots & hd_m - d_n \end{bmatrix} \quad (5)$$

6) Calculate the cost of each recommend item to this user from the above array, remark each result as TravelPenalty( $u, i$ ). It is easy to calculate, just average each column.

7) Re-calculate the recommendation degree with formula 6.

$$RecScore(u, i) = P(u, i) - TravelPenalty(u, i) \quad (6)$$

In this formula,  $P(u, i)$  comes from formula 4, means the prediction mark between this user and an recommended item.

Again, filter the final results by the threshold value, and Obtain the final recommended list.

Implementation of algorithm on mapreduce

The recommendation algorithm of Collaborative Filtering also named memory based algorithm, because it needs huge amount of memory for calculating. In this system, when calculate for the recommendation item list,

data of all users and all items should be load into memory in advance to improve the efficiency of algorithm. For there usually be a large number of data of users and items, so it is hard to process in a single server, especially when the server is busy responding for user's requests. Yet we need a way to calculate in parallel to improve the efficiency of recommendation, cloud computing and map-reduce framework provide an appropriate choice.

In item based collaborative filtering algorithm, the step 2 and 3 need huge amount of memory and calculating, so optimize it to run on a cloud computing environment, and design the parallel algorithm on map-reduce framework, as shown in Fig .5.

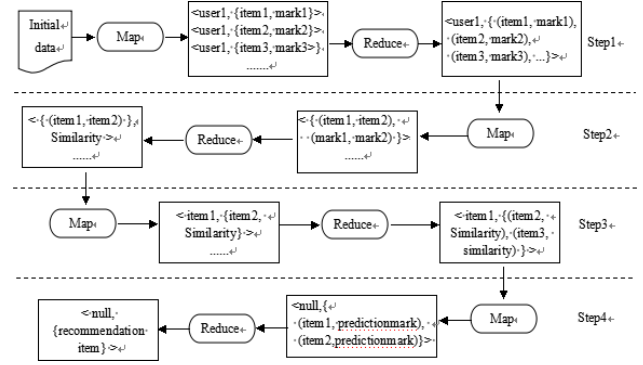


Figure 5 Flow of Map-Reduce algorithm

In Map-Reduce program, all data were represented as key-value pairs, in form of: <key, value>. The key of an initial data will be null, and the value will contain historic evaluation value of a user to an tourism item, for example, {user1, item1, mark1} means that user1 has visited item1, and set his interest point as mark1.

Step1: Data preprocessing. Map function designed to Collect historic data, and Reduce function will merge all evaluation data of each user into one key-value pair.

Step2: Calculate the similarity between items for each user. In Map function the input data will be transformed into columns of evaluation matrix (as Matrix1), and in Reduce function will calculate the similarity by formula 2.

Step3: Clean up the similarity data and save them into persistent storage.

Step4: Generate recommendation items for users. In Map function, combine the user evaluation matrix and the item similarity matrix to work out the prediction mark for each items that not been evaluated by a user, then, in Reduce function, pick up some items that have high prediction mark for recommendation items.

Fig .6 shows the test result of calculate for 10,000-50,000 tourist items, the upper line indicate to calculate with a single server, and the below one indicate to calculate with a cloud environment of 3 virtual machines. We can find that the Map-Reduce optimized algorithm runs fastly than a single server.

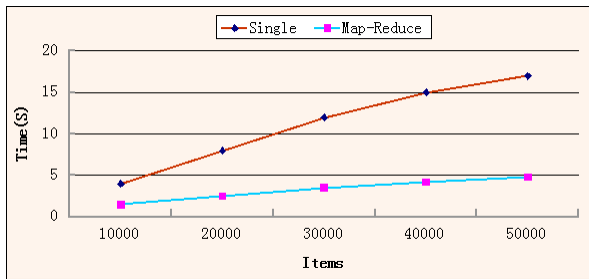


Figure 6 Compare between single server and multiple VM

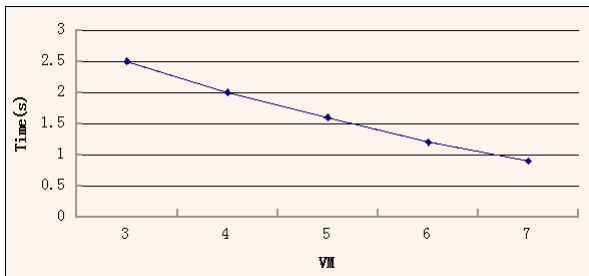


Figure 7 Curve on VM number and calculate time

Fig. 7 shows a curve between virtual machines number and calculate time on a fixed item number of 20,000, it is clearly that the calculate time was shortened when adding more virtual machines.

## V. CONCLUSION

Information pushing technology has changed the method for getting information, combined with the intelligent recommendation technology, customized information service can be provided to users conveniently. In this paper, the author applied technologies of intelligent recommendation as well as information pushing into tourism service platform, and developed a tourism information pushing system based on intelligent recommendation. The architecture of this system is designed to be open and extensible, thus easy to integrate with other open services, and provide user more comprehensive service. The introduction of cloud computing and Map\_Reduce programming make it applies to big data and huge calculation.

## REFERENCES

- [1] ZHOU Er-hu, ZHANG Shui-ping, XIE Bi-chang. The Application of Information Filtering Technology in The Intelligent Information Push System[J], Air Force Engineering University Institute of Telecommunication Engineering, 2010.
- [2] ZHANG Yue-tian, XU Zhong-shan. Review of Current Internet Domain Information Push Technology[J]. Shanxi Library Journal, 2009(6):18-22
- [3] The Future of The Mobile Internet: Push based on context /Interests[OL]. <http://www.web20share.com/2011/03/mobile-internet-trend-interest-based-push.html>.
- [4] Android Implementation Push Solution[OL]. <http://www.cnblogs.com/hanyonglu/archive/2012/03/04/2378971.html>
- [5] TIAN Lei, Guo Tong-qiang, Ji Shou-cun, Wu Bin-wei. The Tourism Information Active Push Service System Based on The Position Related[J]. Zhejiang University College of Computer Science and Technology, 2009.
- [6] WU Jian-ming. Research on Intelligent Recommendation System Based on Web log Mining[D]. Zhejiang University College of Computer Science and Technology, 2006
- [7] LV Xue-ji. Research on Intelligent Recommendation System Based on Cloud Computing Platform[D]. Anhui University, 2012.
- [8] XIANG Liang. Recommendation System Implementation, People's Posts and Telecommunications Press, 2010.
- [9] Bissell, D. On Watain. Time Soc. 18(2009):410-426
- [10] Buyukozkan, G, Ergun, B. Intelligent system application in electronic tourism. Expert Syst. Appl. 38(2010)368-455
- [11] Fu, J., Chakpitak, N., Goldsmith, P., Sureephong, P., Kunarucks, T., Knowledge Management in e-Tourism Curriculum Design: A Knowledge Supply Chain. Proceedings of the 8th International Conference on Intellectual Capital, Knowledge Management and Organisational Learning, Vols 1 and 2. Acad Conferences Ltd, Nr Reading, 2011, pp.660-683.
- [12] Icaza, S.U.: Driving for Santiago., Exploring the use of vehicles by low income residents from the perspective of sustainable mobility. Eure 35, 2009, pp.78-95
- [13] Hsu, M.K., Huang, Y.H., Huang, F.C., Exploring the Determinations of Virtual Tourist Community Participation. Proceedings of International Symposium on Green Hospital and Tourism Management. American Scholars Press, Marietta, 2010, pp.339-390
- [14] Jensen, O.B., Flows of Meaning, Cultures of Movement – Urban Mobility as Meaningful Everyday Life Practice. Mobilities 4, 2009, pp.1005-1132
- [15] Hui, A.S, Things in motion, things in practices: How mobile practice networks facilitate the travel and use of leisure objects. J. Consum. Cult. 12, 2012, pp.190-206