MRGIR: Open Geographical Information Retrieval using MapReduce

Zhiang Wu[1]*, Bo Mao[2], Jie Cao[1]
[1] Jiangsu Provincial Key Laboratory of E-business, Nanjing University of Finance and Economics, Nanjing, China
[2] Geodesy and Geoinformatics, KTH, Drottning Kristinas vag 30, SE-100 44 Stockholm, Sweden
*Corresponding author, e-mail: zawuster@gmail.com

Abstract—**City objects recommendation based on characteristics of users, location, time and weather is a challenging issue in geographical information retrieval (GIR). In the meanwhile, city objects recommendation is a computation-intensive and data-intensive application. Cloud computing has gained significant attention in recent years to process the large volume of data. MapReduce framework is currently a most dominant technology in cloud computing. Augmented User-based Collaborative Filtering (AUCF) algorithm which can effective deal with hybrid variable types is proposed firstly. Then, MapReduce for GIR (MRGIR) is presented and AUCF is implemented within MRGIR as an example. The MRGIR is implemented in Hadoop which is an open source framework for MapReduce. Experimental results shows that with moderate number of map tasks, the execution time of GIR algorithms (i.e., AUCF) can be reduced remarkably.**

Key words-geographical information retrieval; cloud computing; MapReduce; Hadoop

I. Introduction

It has been more than 20 years since the term “Geographic information science” was proposed [1]. Eight topics were suggested by GoodChild [2] about the science, four of which the data collection, capture, modeling and structuring are connected with the geographic information retrieval methods. He also indicate that in the next decade, “knowing where everything is, at all time” as the first challenge lying ahead for GIScience [1]. To achieve that target, Geographic information retrieval methods suitable for the World Wild Web are extremely essential.

Along with the rapidly development of Internet, the amount of open accessible information is increasing dramatically. Especially nowadays, everyone can use their smart phone to update their status anytime, anywhere. Some of the users even are keeping online all the time. They post comments and share photos. The information updated by the Internet users is the base for us to achieve the goal of “knowing where everything is, at all time”. That information contains a lot geographical information which can be used to support the data mining, decision making, geo-visualization, navigation and other applications. The open geographic information on the Internet is playing an increasingly important role in these applications and services such as search engine (Google Map), Location based services, etc. It is necessary to retrieval the open geographical information from the Internet and to manage it for different purposes. As a specialized branch of traditional Information Retrieval (IR), the Geographical Information Retrieval (GIR) emphases on spatial and geographic indexing and retrieval. GIR Systems often contain following parts: geo-tagging, text and geographic indexing, data storage, geographic relevance ranking, clustering and visualization for results. These processing steps require powerful computation capacity especially for the large volume data set such as the data on the Internet. The existing data processing and management methods of GIR such as database cannot efficiently deal with such massive data. Therefore, the cloud computing techniques should be applied for the purpose.

Recommending surrounding city objects to users based on characteristics of users, user’s location, time and weather is an important application in GIR. For instance, potentially interesting city objects may be recommended to the tourist who is not familiar with the local environment. The contextual information, such as user’s gender, user’s age, user’s accompaniers, user’s location, time, weather and so on, will affect the recommendation. The geographic information of users’ visiting histories is a good resource for recommendation. In the meanwhile, since the data volume of the geographic information is quite huge especially when everyone can upload their real time location information with the smart phones. This case is prevalent in GIR applications.

On the basis of cluster computing, P2P, Grid computing and Web 2.0, Cloud computing rapidly emerges as a hot issue in both industrial and academic circles [3]. Cloud computing is considered to be an effective and efficient technique to solve data-intensive applications, and a multitude of applications have adopted cloud computing to data processing, such as machine learning, satellite data processing, PageRank, scientific data analysis, etc. The Google’s MapReduce programming model, which serves for processing large data sets in a massively parallel manner, is a representative technique in cloud computing [4].

In this paper, we formally describe the city objects recommendation problem, and define the data format of users’

Supported by National Natural Science Foundation of China (No.71072172) the program for New Century Excellent Talents in university (No.NCET-07-0411), the Natural Science Foundation for key basic research of the Jiangsu Higher Education Institutions of China (No.07KJA52004), Jiangsu Provincial Key Laboratory of Network and Information Security (No. BM2003201), Project for industrialized advance of scientific achievements of universities in Jiangsu (No.JH09-21) and the joint research for Forward Looking Production, Teaching & Research in Jiangsu (No.by20091005).

978-1-61284-848-8/11/$26.00 ©2011 IEEE

visiting histories exhibiting good scalability. Then, we present a recommendation algorithm called *Augmented User-based Collaborative Filtering (AUCF)*. AUCF can effectively deal with variable data types and generate recommendation based on various contextual information. MRGIR aims to implement various GIR algorithms within MapReduce framework. We elaborate the implementation details of AUCF within MapReduce. Since AUCF reads input dataset twice and does self-merge operation, the original MapReduce does not fit for AUCF. Therefore, *Map-Reduce-Merge model,* an effective framework for processing multiple heterogeneous datasets, is employed to implement AUCF.

The remainder of this paper is organized as follows. Section 2 reviews related work on GIR, cloud computing and recommender algorithms. Section 3 introduces problem formalization. Section 4 elaborates details of the AUCF. Section 5 presents implementation of AUCF within MapReduce framework. Experimental evaluation of our work is presented in Section 6 and a conclusion is drawn in Section 7.

II. Related Work

Geographic information retrieval has been acknowledged for a long time. Woodruff and Plaunt [5] first implemented a system that automatically indexing the documents based on places name. They identified all the geographic words which were analyzed and assigned with coordinate data. At last the polygonal overlay of the outputted data was preformed and the geo-referenced index for the document is generated for information retrieval. Their system can not only support the explicit spatial locations i.e. the place name but also the more descriptive geographic characteristics. ADL Geospatial Integration Project [6] and the Berkeley Going Places in the Catalog Project [7] are also text parsing based GIR system designed for document retrieval in the library. Bordogna et al. [8] proposed a GIR system—Geo-Finder which retrieval the information based on two criteria: content constrain and spatial constrain. His system allowed user to specify a trade off between the two constrains.

As the Internet becomes the main information source in our daily life, the focus of GIR is turned into the World Wild Web. McCurley [9] investigated some methods to discovering geographic information from the web pages, and described a navigational tool for browsing web resources by geographic proximity. The project SPIRIT: Spatially-Aware Information Retrieval on the Internet [10] made use of the geographical and conceptual ontologies for information retrieval. The geographical ontology is used to create geographical similarity measures and rank the document relevance. In 2005, the SPIRIT has annotate around 900,000 web pages taken from a 1TB web crawl, focused on regions in the UK, France, Germany and Switzerland [11]. Markowetz et al. [12] studied the problems of the GIR in web environment by first outlining the architecture of mapping Internet resources into geographic locations; then presenting geospatial search engines and proposing geospatial analysis of web crawls. Kotera et al. [13] indicated that searching the digital maps by directory services can yield more information about places and objects on the maps. They proposed a method to recommend the geographic information based on the search history of the user.

Along with the development of Internet, especially the dramatically increasing of the data volume generated everyday; the existing GIR systems require more powerful computation, processing and storage capacity to deal with the massive data from Internet. Cloud computing is firstly proposed in 2007 by IBM and Google. Currently, providers such as Amazon, Google, Salesforce, IBM, Microsoft and Sun Microsystems have begun to establish new data centers for hosting cloud computing applications in various locations around the world to provide redundancy and ensure reliability in case of site failures [14]. The Google's MapReduce programming model, a most dominant technology in cloud computing, can effectively process large datasets in a massively parallel manner. Hadoop developed by the Apache Software Foundation is an open source MapReduce framework [15]. The key components of Hadoop are HDFS (Hadoop Distributed File System) and MapReduce framework. HDFS is a distributed file system designed to run on hardware, and it also manages all files scattered in nodes and provides high throughput of data access. MapReduce provides a programming model for processing large scale datasets in distributed manner. Jobs submitted to Hadoop consist of a map function and a reduce function. Hadoop breaks each job into multiple tasks. Firstly, map tasks process each block of input (typically 64MB) and produce intermediate results, which are key-value pairs. These are saved to disk. Next, reduce tasks fetch the list of intermediate results associated with each key and run it through the reduce function, which produces output.

Recommender algorithm is the core of recommender system, and *Collaborative Filtering (CF)* is a widely used recommendation algorithm [16]. CF algorithm relies on historical information of users. *User-based Collaborative Filtering (UCF*) is probably the most successful and widely used technique which is adopted by a multitude of well-known recommender systems such as Tapestry, MovieLens, Amazon Book Store, YouTube, Facebook, and so forth [17]. However, UCF requires computation that grows linearly with the number of users and items. With millions of users and items, UCF suffers serious scalability problem. Unfortunately, the data volume of the geographic information is quite huge. Therefore, it is necessary to utilize the remarkable capability provided by MapReduce processing large datasets to enhance the efficiency of UCF. In the meanwhile, data types of geographic information are complicated. For instance, gender of user and city are binary variables, location is an interval-scale variable, and age bracket, accompanier, time, weather are categorical variables. Traditional UCF cannot deal with them, thus we propose a new UCF called AUCF to solve the problem.

III. Problem Statement

Recommending city objects to users based on his/her characteristics and locations is a challenging issue in geographical information retrieval. This section presents the data format of geographical information, and states the problem formally.

Let City denote the city object set with size |C| and User denote the user class set with size |U|. We utilize several characteristics to describe a user, i.e. gender, age bracket, accompanier. Then, a record of input data illuminates a user belong to a specific class has travelled what city objects on a certain position at a certain time under a weather condition. Fig. 1 shows the input data format.

User	Location	Time	Weather	City			
				C_1	***C_2***	****……****	***$C_{\|C\|}$***
U_1	***<x,y>***	***noon***	***Sunny***	***1***	***0***	***……***	***1***

Figure 1. Input data format

Now, the problem is that how to recommend city objects to a new user according to the existing records. This is a kind of active geographic information retrieval method which could be useful in many applications. For example, in the navigation related applications, when the user goes to some place, he/she may don't want miss the nearby hot spots. This information is especial valuable for the tourist who is not familiar with the local environment. Therefore, the users' visiting histories are a good resource for the information retrieval. In this paper, we will first collect the visiting data of different users as the user profiles, based on which the recommend algorithm is designed and implemented using MapReduce methods for geographic information retrieval.

The data format of the user profile shown in Fig. 1 contains many kinds of data types. For instance, gender of user and city are binary variables, location is an interval-scale variable, and age bracket, accompanier, time, weather are categorical variables.

IV. Augmented User-based Collaborative Filtering

Traditional UCF algorithms can only deal with interval-scale variables. However, geographical data information consists of hybrid variable types. Similarity computation, the foundation of UCF algorithms, based on hybrid variable types offers a challenge to us. This section presents AUCF algorithm to solve the problem. AUCF algorithm consists of two phases, where the first phase computes similarity and then select k nearest neighbors, and the second phase predicts potential useful city objects for users.

A. Phase 1: Computing similarity based on hybrid data types

Fields in input data format as shown in Fig. 1 are easy to extended. Assume that there are p variables determining the similarity of users. The *dissimilarity* between u_i and u_j can be defined as:

$$d(u_i,u_j)=\frac{\sum_{f=1}^{p}\delta_{ij}^{f}d_{ij}^{f}}{\sum_{f=1}^{p}\delta_{ij}^{f}} \tag{1}$$

In Eq. (1), if u_i or u_j do not assign a value to the f-th variable (x_{if} or x_{jf} are missing), $\delta^f_{ij}=0$; otherwise $\delta^f_{ij}=1$. The dissimilarity of the f-th variable between u_i and u_j is written d^f_{ij} which relies on the variable type. We consider binary variable, interval-scale variable and categorical variable respectively.

- the f-th variable is binary variable or categorical variable: if $x_{if}=x_{jf}$, $d^f_{ij}=0$; otherwise $d^f_{ij}=1$.
- the f-th variable is interval-scale variable: only location variable belongs to the type in our problem. *Manhattan* distance is employed to measure the distance between two positions in city. Let $<L_{xi}, L_{yi}>$ and $<L_{xj}, L_{yj}>$ denote coordinates of u_i and u_j respectively, and d^f_{ij} can be computed as Eq. (2) shown.

$$d_{ij}^{f}=|L_{xi}-L_{xj}|+|L_{yi}-L_{yj}| \tag{2}$$

Selecting k nearest neighbors (kNN) of users is an important step for making accurate recommendation. Based on the dissimilarity calculated by Eq. (1), we select k users whose dissimilarity with the new user are as small as possible.

B. Phase 2: Recommendation based on votes of top-k neighbors

This phase predicts the potential interesting city objects for a new user on the basis of his/her top-k neighbors. Since *City* is a binary vector where $C_i=1$ denotes the user has travelled and is interested in the city object, we predict the interesting degree according to the number of votes of a user's top-k neighbors. The prediction process can be formalized as Eq. (3).

$$P_N=\underset{C_i\in City}{\arg\max}\sum_{m=1}^{k}City_m \tag{3}$$

Top-N city objects having the most votes are recommended to the new user.

V. Implementation of AUCF with Map-Reduce-Merge

Since the data volume of the geographic information is quite huge especially when everyone can upload their real time location information with the smart phones. To deal with such data, MapReduce technology is required to implement the efficient geographic information retrieval. Therefore, we proposed the MRGIR framework, and its implementation is given as follows.

As the increasing in size of the input data, AUCF has become a data intensive operation. The MapReduce program framework proposed by Google simplifies the distributed computing programming, and can be used to process the massive data in parallel. This paper utilizes the MapReduce programming framework for implementing the AUCF algorithm in order to increase the speed of AUCF for large date set.

AUCF needs to read input data two times. The first time is used to compute dissimilarity, and the second time is used to obtain votes of top-k neighbors when computing P_N. The proposed MRGIR framework follows the Map-Reduce-Merge model [18] which enables processing multiple heterogeneous datasets. The signatures of the Map-Reduce-Merge primitives are listed below, where α, β, γ represent dataset lineages.

$$\begin{aligned}
&map:(key_1, value_1) \rightarrow [(key_2, value_2)]_\alpha \\
&reduce:(key_2,[value_2])_\alpha \rightarrow (key_2,[value_3])_\alpha \\
&merge:((key_2,[value_2])_\alpha,(key_3,[value_4])_\beta) \rightarrow [(key_4, value_5)]_\gamma
\end{aligned} \tag{4}$$

In AUCF with Map-Reduce-Merge model, α=β satisfies. Then, the merge function in Eq. (4) does a *self-merge*, similar to *self-join* in relational algebra. Notice that the map and reduce function in Eq. (4) are almost the same as those in the original MapReduce. The only differences are the lineages of the datasets and the production of a key/value list from reduce instead of just values. These changes are introduced because the merge function needs input datasets organized by keys and these keys have to be passed into the function to be merged.

TABLE I. PSEUDO-CODE DESCRIBING THE AUCF ALGORITHM

```
Procedure Map (<a,La>)
d(u,a) is computed by Eq. (1)
Emit(u,<a,sim'(u,a)>)

Procedure Reduce(u; [<a1,d(u,a1)>,<a2,d(u,a2)>.....])
Initialize.PriorityQueue(Q)
for all <a,d(u,a)> in [<a1,d(u,a1)>,<a2,d(u,a2)>.....] do
    if Q:Size() < k then
        Q:Insert(<a,d(u,a)>)
    else if d(u,a) < Q:Max() then
        Q:ExtractMax()
        Q:Insert(<a,d(u,a)>)
    end if
end for
Emit(u;Q:ExtractAll())

Procedure Merge (<u,N(u)>)
for all a in N(u) do
    read City field of a
end for
$P_N$ is computed by Eq. (3)
Emit(u; $P_N$)
```

The details of MRGIR is as follows: (1) Store the rows of input data shown in Fig. 1 as a line of the input files; all the files are managed by the Distributed File System (DFS) and transparent to the users. (2) The Map function reads the record in *<a, La>* from the DFS, computes their similarity with the current user, and returns the top-k similar objects as the middle results. (3) Reduce function gathers all the middle results from Map function and computes the k-nearest neighbor set of the current user. (4) The Merge function will read the DFS again for the input data to obtain *City* field. Then, P_N can be computed based on the votes of top-*k* neighbors on *City* field. The pseudocode of Map, Reduce and Merge functions are given in Table 1.

In table 1, *u* denotes the new user and *N(u)* denotes the top-*k* neighbors of u. <*a, La*> represents key-value pair in MapReduce.

VI. EXPERIMENTAL RESULTS

We have built MRGIR system based on Hadoop. Machines integrated by this data cloud system consist of two clusters each with 32 blade servers in SEUGrid (Southeast University Grid) and an army of PC servers in our laboratory. Two VMs are created in each PC servers with 2.03GH and 1GB of RAM. Two clusters in SEUGrid utilize Red Hat Enterprise 4.0 and all VMs in PC servers utilize Fedora Core 11, and on the top of Linux, Hadoop 0.20.1 is installed and configured as core middleware.

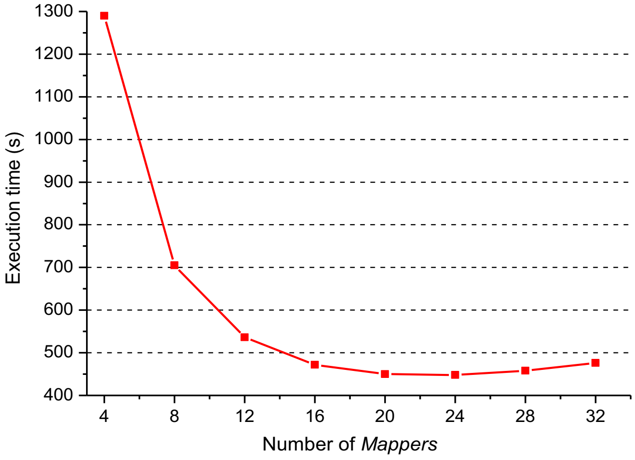

Figure 2. Effect of number of Mappers on execution time

We investigate the effect of the number of *mappers* on the execution time of AUCF. A transaction database is created with 100,000 records and 500 kinds of users. The transaction database is split on average according to the number of *mapper* nodes. In other words, equal number of records is stored in all mapper nodes. Fig. 2 shows the effect of number of *mappers* on execution time. It indicates that with the increase of the number of *mappers*, the execution time decreases. At the beginning, the execution time decreases dramatically with the increase of the number of *mappers.* When the number of *mappers* reaches a threshold (i.e. 16 in Fig. 2), the execution time varies moderately or even increases slightly (i.e. 24, 28 and 32 in Fig. 2). The reason for the above-mentioned phenomenon is that with the increase of *mappers*, the number of records stored in each *mapper* node decreases. When the number of records reduce to a threshold, the time of scanning the database is hard to further decline. Therefore, the time of *Reducer* and communications is rising, thus playing the major role in the whole execution time.

VII. Conclusion

In this paper, we investigate how to apply MapReduce to GIR. City objects recommendation problem, an important application in GIR, is chosen as an example. Firstly, the city objects recommendation problem is stated formally. Secondly, since variable types of geographical information are widely different, traditional CF algorithms cannot effectively deal with it, thus a recommendation algorithm called *AUCF* is proposed. AUCF can effectively deal with variable data types and generate recommendation based on various contextual information.

MRGIR aims to implement various GIR algorithms within MapReduce framework. We elaborate the implementation details of AUCF within MapReduce. Since AUCF reads input dataset twice and does self-merge operation, the original MapReduce does not fit for AUCF. Therefore, *Map-Reduce-Merge model,* an effective framework for processing multiple heterogeneous datasets, is employed to implement AUCF. Experimental results shows that with moderate number of map tasks, the execution time of GIR algorithms (i.e., AUCF) can be reduced remarkably.

Now, MRGIR system based on Hadoop is even at initial stage. In the future, we will continue to improve MRGIR system in order to incorporate more algorithms and applications of GIR.

Acknowledgment

The authors want to thank all members in the team of JSELAB for their hard-working on MRGIR system and other helps.

References

[1] M. F. Goodchild, “Twenty years of progress: GIScience in 2010,” Journal of Spatial Information Science, 2010(1), pp. 3-20.

[2] M. F. GoodChild, “Geographical information science,” International Journal of Geographical Information Systems, vol. 6(1), 1992, pp. 31-46.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, et al., “Above the Clouds: A Berkeley View of Cloud Computing,” Technical Report 2009-28, UC Berkeley, 2009.

[4] L. Ralf, “Google’s MapReduce programming model—Revisited,” The Journal of Science of Computer Programming, vol. 70(1), 2008, pp. 1-30.

[5] A. G. Woodruff and C. Plaunt, “GIPSY: Automated geographic indexing of text documents,” Journal of the American Society for Information Science, vol. 45(6), 1994, pp. 645–655.

[6] J. Frew, “ADL Textual-Geospatial Integration Project,” http://nkos.slis.kent.edu/2002workshop/frew.ppt, Accessed at April 5, 2011.

[7] M. Buckland, F. C. Gey, and R. Larson, “Going Places in the Catalog: Improved Geographical Access,” http://www.sims.berkeley.edu/~buckland/catplace.pdf, Accessed at April 5, 2011.

[8] B. Gloria, G. Ghisalberti, M. Pagani, and G. Psaila, “Geographic Information Retrieval based on Two Orthogonal Criteria,” Proceedings of IFSA/EUSFLAT Conf, pp.867-872, 2009.

[9] K. McCurley, “Geospatial Mapping and Navigation of the Web,” Proc. of the WWW10 Conf. Hong Kong, pp. 221–229, 2001.

[10] C. B. Jones and et al, ”Spatial Information Retrieval and Geographical Ontologies: An Overview of the SPIRIT project,” In: SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 387-388, 2002.

[11] P. Clough, “Extracting metadata for spatially-aware information retrieval on the internet,” In Proceedings of the 2005 workshop on Geographic information retrieval (GIR '05). ACM, New York, pp. 25-30, 2005.

[12] A. Markowetz, T. Brinkhoff, and B. Seeger, “Geographic Information Retrieval,” In 3rd International Workshop on Web Dynamics, pp. 115-125, 2004.

[13] R. Kotera, D. Kitayama, and K. Sumiya, “Geographical information retrieval based on user's operation on both digital maps and directory services,” Geoinformatics 17th International Conference, pp.1-6, 2009.

[14] R Buyya, C. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility,” Future Generation Computer Systems, vol. 25(6), pp. 599–616 , 2009.

[15] Hadoop, “The Apache Software Foundation,” http://hadoop.apache.org/core, Accesssed at April 5, 2011.

[16] R. M. Bell and Y. Koren, “Improved neighborhood-based collaborative filtering,” Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining California, pp. 7-14, 2007.

[17] G. Linden, B. Smith, and J. York, “Amazon.com Recommendations: Item-to-item Collaborative Filtering,” IEEE Internet Computing 7, pp. 76-80, 2003.

[18] H. Yang, A. Dasdan, R. Hsiao, and D. S. Parker, “Map-reduce-merge: simplified relational data processing on large clusters,” Proceedings of the 2007 ACM SIGMOD international conference on Management of data. New York, USA, pp. 1029-1040, 2007.