International Journal On Engineering Technology and Sciences – IJETS™
ISSN(P): 2349-3968, ISSN (O): 2349-3976
Volume III, Issue V, May- 2016

KASR: A Keyword-Aware Service Recommendation Method on Map Reduce for Big Data Applications

Ms.S.Priya
Department of Computer Science and Engineering
SSM College of Engineering, Komarapalayam,
Tamil Nadu, India
priyashreenivasan@gmail.com

Mr.A.T.Ravi
Department of Computer Science and Engineering
SSM College of Engineering, Komarapalayam,
Tamil Nadu, India
csehod@ssmce.ac.in

Abstract— *Big data refers to datasets that aren't solely massive, however additionally high in variety and velocity that makes them troublesome to handle using tradition tools and techniques. Due to the rise of such Data, solutions got to be studied and provided so as to handle and extract worth and information from these datasets. Nowadays Web services are very widespread. Recommender systems represent user preferences for the aim of suggesting things to get or examine. They are many basic applications in electronic commerce and data access, providing suggestions that effectively prune massive data areas so users are directed toward those things that best meet their wants and preferences. A variety of techniques are projected for activity recommendation, including content-based, collaborative, knowledge-based and different techniques. In this project, we are presenting "Keyword-Aware Service Recommendation Method (KASR)", to deal with the above challenges. It aims at presenting a customized service recommendation list and recommending the foremost applicable services to the users effectively. Specifically, keywords area unit wont to indicate users' preferences, and a user-based cooperative Filtering algorithm is adopted to get applicable recommendations. To improve the scalability and efficiency of KASR in "Big Data" environment, the proposed system proposes techniques that have been implemented it on a Map Reduce framework in Hadoop platform.*

Keywords— *Big data, Keyword, Service, Recommendation, Hadoop*

I. INTRODUCTION

Data is growing at an enormous speed creating it tough to handle such large amount of data. The main problem in handling such large amount of data is as a result of that the result of that the amount is increasing quickly as compared to the computing resources. The Big data term that is getting used currently a days is quite name because it points out solely the dimensions of the data not putting an excessive amount of attention to its different existing properties.

Today, Big Data management stands out as a challenge for IT corporations. The answer to such a challenge is shifting more and more from providing hardware to provisioning more manageable package solutions. Big Data additionally brings new opportunities and significant challenges to trade and domain. Similar to most big data applications, the large data tendency conjointly poses significant impacts on service recommender systems. With the growing range of different services, effectively recommending services that users most well-liked has become a vital analysis issue. Service recommender systems are shown as valuable tools to assist users modify services over load and supply acceptable recommendations to them. Examples of such sensible applications include CDs, book, web content and numerous alternative product currently use recommender systems. Over the last decade, there has been a lot of analysis done each in business and world on developing new approaches for service recommender systems. Many major e-commerce Websites are used recommendation systems to produce relevant suggestions to their customers. The recommendations may be supported numerous parameters, like item popular on the company's Website; user characteristics like geographical location or different demographic information; or past shopping for behavior of prime customers.

II. RELATED WORK

There have been many recommender systems developed in both academia and industry. In [33], the authors propose a Bayesian-inference-based recommendation system for on-line social networks. They show that the proposed Baye-sian-inference-based recommendation is better than the existing trust-based recommendations and is comparable to Collaborative Filtering recommendation. In [13], Adomavi-cius and Tuzhilin give an overview of the field of recom-mender systems and describe the current generation of rec-ommendation methods. They also describe various limita-tions of current service recommendation methods, and dis-cuss possible extensions that can improve recommendation capabilities and make recommender systems applicable to an even broader range of applications. Most existing service

recommender systems are only based on a single numerical rating to represent a service's utility as a whole [34]. In fact, evaluating a service through multiple criteria and taking into account of user feedback can help to make more effec-tive recommendations for the users. With the development of cloud computing software tools such as Apache Hadoop, Map-Reduce, and Mahout, it be-comes possible to design and implement scalable recom-mender systems in "Big Data" environment. The authors of [35] implement a CF algorithm on Hadoop. They solve the scalability problem by dividing dataset. But their method doesn't have favorable scalability and efficiency if the amount of data grows. [36] presents a parallel user profiling approach based on folksonomy information and imple-ments a scalable recommender system by using Map-Reduce and Cascading techniques. Jin et al. [37] propose a large-scale video recommendation system based on an item-based CF algorithm. They implement their proposed approach in Qizmt, which is a .Net Map-Reduce framework, thus their system can work for large-scale video sites. Generally speaking, comparing with existing methods, KASR utilizes reviews of previous users to get both of user preferences and the quality of multiple criteria of candidate services, which makes recommendations more accurate. Moreover, KASR on MapReduce has favorable scalability and efficiency.

III. MAIN CONTRIBUTIONS

1) A keyword-aware service recommendation method, named KASR, is proposed in this paper, which is based on a user-based Collaborative Filtering algorithm. (2) In KASR, keywords extracted from reviews of previous users are used to indicate their preferences. Moreover, we implement it on a distributed computing platform, Hadoop, which uses MapReduce as its computing framework. The remainder of the paper is organized as follows: Section 2 introduces the preliminary knowledge of our method. Then a keyword-aware service recommendation method, named KASR, is described in Section 3. Section 4 presents the implementation of KASR on MapReduce. In Section 5, experiments are designed and analyzed to evaluate the ac-curacy and scalability of KASR. Related works are pre-sented in Section 6. Section 7 concludes the paper and gives an outlook on possible continuations of our work.

IV. PROPOSED SYSTEM

In this project, we propose a keyword aware service recommendation method, named KASR. In this method, keywords are used to indicate both of users' preferences and the quality of candidate services. A user based CF algorithm is adopted to generate appropriate recommendations. KASR aims at calculating a personalized rating of each candidate service for a user, and then presenting a personalized service recommendation list and recommending the most appropriate services to him/her.

Moreover, to improve the scalability and efficiency of our recommendation method in "Big Data" environment, we implement it in a MapReduce framework on Hadoop by splitting the proposed algorithm into multiple MapReduce phases. The proposed system has following advantages.

i) The proposed method presenting a personalized service recommendation list and recommending the most appropriate service(s) to the users.

ii) By implementing the KASR in Big Data environment we have improved the scalability and efficiency.

iii) The accuracy of the service recommender systems over exiting approaches will be improved.

iv) Data analysis will be faster with the growth of data requirements.

v) Integration and acquisition of multiple data sources can be managed effectively

vi) Different cloud environments data sharing can be analyzed

V. IMPLEMENTATION

1) CAPTURE USER PREFERENCES BY A KEYWORD-AWARE APPROACH

In this step, the preferences of active users and previous users are formalized into their corresponding preference keyword sets respectively.an active user refers to a current user needs recommendation.

a. Preferences of an active user:

An active user can give his/her preferences about candidate services by selecting keywords from a keyword candidate list, which reflect the quality criteria of the services he/she is concerned about.

b. Preferences of previous users:

The preferences of a previous user for a candidate service are extracted from his/her reviews for the service according to the keyword candidate list and domain thesaurus.

Preprocess: Firstly, HTML tags and stop words in the reviews snippet collection should be removed to avoid affecting the quality of the keyword extraction in

the next stage. The keyword stripping algorithm is used to remove the commoner morphological and in flexional endings from words in English.

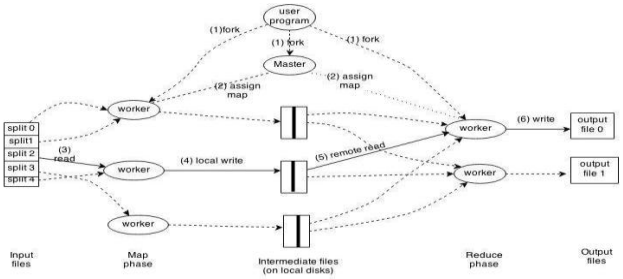
Keyword extraction: In this phase, each review will be transformed into a corresponding keyword set according to the keyword-candidate list and domain thesaurus. If the review contains a word in the domain thesaurus, then the corresponding keyword should be extracted into the preference keyword set of the user. If a keyword appears more than once in a review, the times of repetitions will be recorded. In this project, it is regarded that keywords appearing multiple times are more important. The times of repetitions will be used to calculate the weight of the keyword in preference keyword set in the next step.

2) SIMILARITY COMPUTATION

The process of this step is to identify the reviews of previous users who have similar tastes to an active user by finding neighborhoods of the active user based on the similarity of their preferences. Before similarity computation, the reviews unrelated to the active user's preferences will be filtered out by the intersection concept in set theory. If the intersection of the preference keyword sets of the active user and a previous user is an empty set, then the preference keyword set of the previous user will be filtered out. Two similarity computation methods are introduced in our recommendation method: an approximate similarity computation method and an exact similarity computation method. The approximate similarity computation method is for the case that the weights of the keywords in the preference keyword set are unavailable, while the exact similarity computation method is for the case that the weight of the keywords are available.

3) EXECUTION OF MAPREDUCE MODEL

The Map invocations are distributed across multiple machines by automatically partitioning the input data into a set of M splits. The input splits can be processed in parallel by different machines. Reduce invocations are distributed by partitioning the intermediate key space into R pieces using a partitioning function (e.g., hash (key) mod R). The number of partitions (R) and the partitioning function are specified by the user. Figure 6.1 shows the overall flow of a Map Reduce operation in our implementation, when the user program calls the MapReduce function. The following sequence of actions occurs.



i) The MapReduce library in the user program first splits the input files into M pieces of typically 16 megabytes to 64 megabytes (MB) per piece (controllable by the user via an optional parameter). It then starts up many copies of the program on a cluster of machines.

ii) One of the copies of the program is special-the master. The rest are workers that are assigned work by the master. There are M map tasks and R reduce tasks to assign. The master picks idle workers and assigns each one a map task or a reduce task.

iii) A worker who is assigned a map task reads the contents of the corresponding input split. It parses key/value pairs out of the input data and passes each pair to the user-defined Map function. The intermediate key/value pairs produced by the Map function are buffered in memory.

iv) Periodically, the buffered pairs are written to local disk, partitioned into R regions by the partitioning function. The locations of these buffered pairs on the local disk are passed back to the master, who is responsible for forwarding these locations to the reduce workers.

v) When a reduce worker is notified by the master about these locations, it uses remote procedure calls to read the buffered data from the local disks of the map workers. When a reduce worker has read all intermediate data, it sorts it by the intermediate keys so that all occurrences of the same key are grouped together. The sorting is needed because typically many different keys map to the same reduce task. If the amount of intermediate data is too large to fit in memory, an external sort is used.

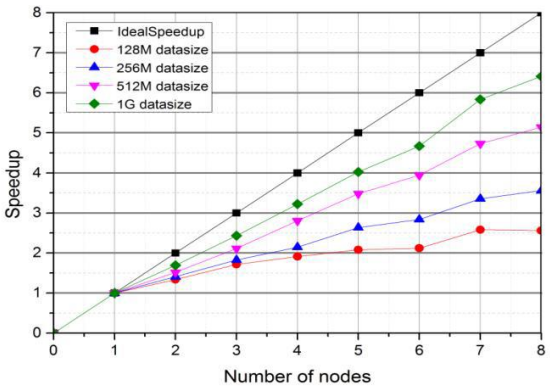
vi)The reduce worker iterates over the sorted intermediate data and for each unique intermediate key encountered, it passes the key and the corresponding set of intermediate values to the user's Reduce function. The output of the Reduce function

is appended to a final output file for this reduces partition.

vii)When all map tasks and reduce tasks have been completed, the master wakes up the user program. At this point, the MapReduce call in the user program returns back to the user code. After successful completion, the output of the MapReduce execution is available in the R output files (one per reduce task, with file names as specified by the user). Typically, users do not need to combine these R output files into one file. They often pass these files as input to another MapReduce call, or use them from another distributed application that is able to deal with input that is partitioned into multiple files

. VI. EXPERIMENTAL RESULTS

A well-accepted scalability metric, Speedup [33], is adopted to measure the performance in the scalability of KASR. Speedup refers to how much a parallel algorithm is faster than a corresponding sequential algorithm, which can be defined as follows:
where *p* is the number of processors, *T1* is the sequential execution time, *Tp* is the parallel execution time with *p* processors. If the speedup has a linear relation with the numbers of nodes with the data size fixed, the algorithm will have good scalability. To verify the scalability of KASR, experiment is conducted respectively in a cluster of nodes ranging from 1 to 8. There are 4 synthetic datasets used in the experiments (128M, 256M, 512M and 1G data size). Fig. 8 shows the speedup of KASR (Here, KASR-ESC method is adopted in the scalability experiment). From Fig. 8, we can see that the speedup of KASR increases relative linearly with the growth of the number of nodes. Meanwhile, larger dataset obtained a better speedup. When the data size is 1G and the number of nodes is 8, the speedup value reaches 6.412, which is 80.15% (6.412/8=80.15%) of the ideal speedup. The experimental result shows that KASR on Map-Reduce in Hadoop platform has good scalability over "Big Data" and performs better with larger dataset.



Overall, these experimental results show that KASR performs well in accuracy, and KASR on Mapreduce frame-work has good scalability in "Big Data" environment

VII.CONCLUSION AND FUTURE WORK

We have proposed a keyword-aware service recommendation method, named KASR. In KASR, keywords are used to indicate users' preferences, and a user based Collaborative Filtering algorithm is adopted to generate appropriate recommendations. More specifically, a keyword-candidate list and domain thesaurus are provided to help obtain users' preferences. The active user gives his/her preferences by selecting the keywords from the keyword-candidate list, and the preferences of the previous users can be extracted from their reviews for services according to the keyword-candidate list and domain thesaurus. Our method aims at presenting a personalized service recommendation list and recommending the most appropriate service(s) to the users. Moreover, to improve the scalability and efficiency of KASR in "Big Data" environment, we have implemented it on a MapReduce framework in Hadoop platform. Finally, the experimental results demonstrate that KASR significantly improves the accuracy and scalability of service recommender systems over existing approaches.

REFERENCES

[1] J. Manyika, M. Chui, B. Brown, et al, "Big Data: The next frontier for innovation, competition, and productivity," 2011.

[2] C. Lynch, "Big Data: How do your data grow?" Nature, Vol. 455, No. 7209, pp. 28-29, 2008. [3] F. Chang, J. Dean, S. Ghemawat, and W. C. Hsieh, "Bigtable: A distributed storage system for structured data," ACM Transactions on Computer Systems,Vol. 26, No. 2 (4) , 2008. [4] W. Dou, X. Zhang, J. Liu, J. Chen, "HireSome-II: Towards Privacy-Aware Cross-Cloud Service Composition for Big Data Applica-tions," IEEE Transactions on Parallel and Distributed Systems, 2013. [5] G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," IEEE Internet Computing, Vol. 7, No.1, pp. 76-80, 2003

[3] F. Chang, J. Dean, S. Ghemawat, and W. C. Hsieh, "Bigtable: A distributed storage system for structured data," ACM Transactions on Computer Systems,Vol. 26, No. 2 (4) , 2008. [4] W. Dou, X. Zhang, J. Liu, J. Chen, "HireSome-II: Towards Privacy-Aware Cross-Cloud Service Composition for Big Data Applica-tions," IEEE Transactions on Parallel and Distributed Systems, 2013

[4] W. Dou, X. Zhang, J. Liu, J. Chen, "HireSome-II: Towards Privacy-Aware Cross-Cloud Service Composition for Big Data Applica-tions," IEEE Transactions on Parallel and Distributed Systems, 2013. [5] G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," IEEE Internet Computing, Vol. 7, No.1, pp. 76-80, 2003. [6] M. Bjelica, "Towards TV Recommender System Experiments with User Modeling," IEEE Transactions on Consumer Electronics, Vol. 56, No.3, pp. 1763-1769, 2010.

[5] G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," IEEE Internet Computing, Vol. 7, No.1, pp. 76-80, 2003. [6] M. Bjelica, "Towards TV Recommender System Experiments with User Modeling," IEEE Transactions on Consumer Electronics, Vol. 56, No.3, pp. 1763-1769, 2010.

[6] M. Bjelica, "Towards TV Recommender System Experiments with User Modeling," IEEE Transactions on Consumer Electronics, Vol. 56, No.3, pp. 1763-1769, 2010.

[7] M. Alduan, F. Alvarez, J. Menendez, and O. Baez, "Recommender System for Sport Videos Based on User Audiovisual Consumption," IEEE Transactions on Multimedia, Vol. 14, No.6, pp. 1546-1557, 2013. [8] Y. Chen, A. Cheng and W. Hsu, "Travel Recommendation by Min-ing People Attributes and Travel Group Types From Community-Contributed Photos". IEEE Transactions on Multimedia, Vol. 25, No.6, pp. 1283-1295, 2012. [9] Z. Zheng, X Wu, Y Zhang, M Lyu, and J Wang, "QoS Ranking Pre-diction for Cloud Services," IEEE Transactions on Parallel and Dis-tributed Systems, Vol. 24, No. 6, pp. 1213-1222, 2013.

[8] Y. Chen, A. Cheng and W. Hsu, "Travel Recommendation by Min-ing People Attributes and Travel Group Types From Community-Contributed Photos". IEEE Transactions on Multimedia, Vol. 25, No.6, pp. 1283-1295, 2012. [9] Z. Zheng, X Wu, Y Zhang, M Lyu, and J Wang, "QoS Ranking Pre-diction for Cloud Services," IEEE Transactions on Parallel and Dis-tributed Systems, Vol. 24, No. 6, pp. 1213-1222, 2013

[9] Z. Zheng, X Wu, Y Zhang, M Lyu, and J Wang, "QoS Ranking Pre-diction for Cloud Services," IEEE Transactions on Parallel and Dis-tributed Systems, Vol. 24, No. 6, pp. 1213-1222, 2013

[10] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and Evaluating Choices in a Virtual Community of Use," In CHI '95 Proceedings of the SIGCHI Conference on Human Factors in Com-puting System, pp. 194-201, 1995.