Big Data Application Using Map Reduce

Mr. M.P.Gaikwad[1] Mr. C.S.Arage[2]
Mr. Ashu Dhotre[3], Mr. Akshay Patil[4]
Department of Computer Science and Engineering
Sou. Sushila Danchand Ghodawat Charitable Trust's Sanjay Ghodawat Group of Institutions Maharastra
[1] gaikwad.mp@sginstitute.in [2] Arage.cs@sginstitute.in [3] ashudhotre43@gmail.com
[4]banshelkikarakshay@gmail.com

ABSTRACT

Service recommender systems have been shown as beneficial tools for providing proper recommendations to users. In the last decade, the amount of customers, services and online information has grown rapidly.Therefore, traditional service recommender systems often endure from scalability and inefficiency problems when processing or analysing such large-scale data. Moreover, most of existing service recommender systems present the same ratings and rankings of services to different users without considering diverse users' preferences, and therefore fails to meet users' personalized requirements. In this paper, we propose a Keyword-Aware Service Recommendation method, named KASR, to address the above challenges. It aims at presenting a personalized service recommendation list and recommending the most appropriate services to the users effectively. Specifically, keywords are used to indicate users' preferences, and a user-based Collaborative Filtering algorithm is adopted to generate appropriate recommendations. To improve its scalability and efficiency in big data environment, KASR is implemented on Hadoop, a widely-adopted distributed computing platform using the MapReduce parallel processing paradigm. Finally, extensive experiments are conducted on real-world data sets, and results demonstrate that KASR significantly improves the accuracy and scalability of service recommender systems over existing approaches.

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 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 confront is turning more and more from providing hardware to provisioning more manageable package Solutions. Big Data as well brings new opportunities and significant challenges to trade and department.

Similar to most big data applications, the large data tendency again posture 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 apt 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 analytical information; or past shopping for behaviour of prime customers.

I.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 break the proposed algorithm into multiple MapReduce phases. The proposed system has following advantages. The proposed method presenting a personalized service recommendation list and recommending the most appropriate service(s) to the users. By implementing the KASR in Big Data environment we have improved the scalability and efficiency. The accuracy of the service recommender systems over exiting approaches will be improved. Data analysis will be faster with the growth of data requirements. Integration and gain of multiple data sources can be

managed effectively. Different cloud environments data sharing can be analyzed.

II. 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.

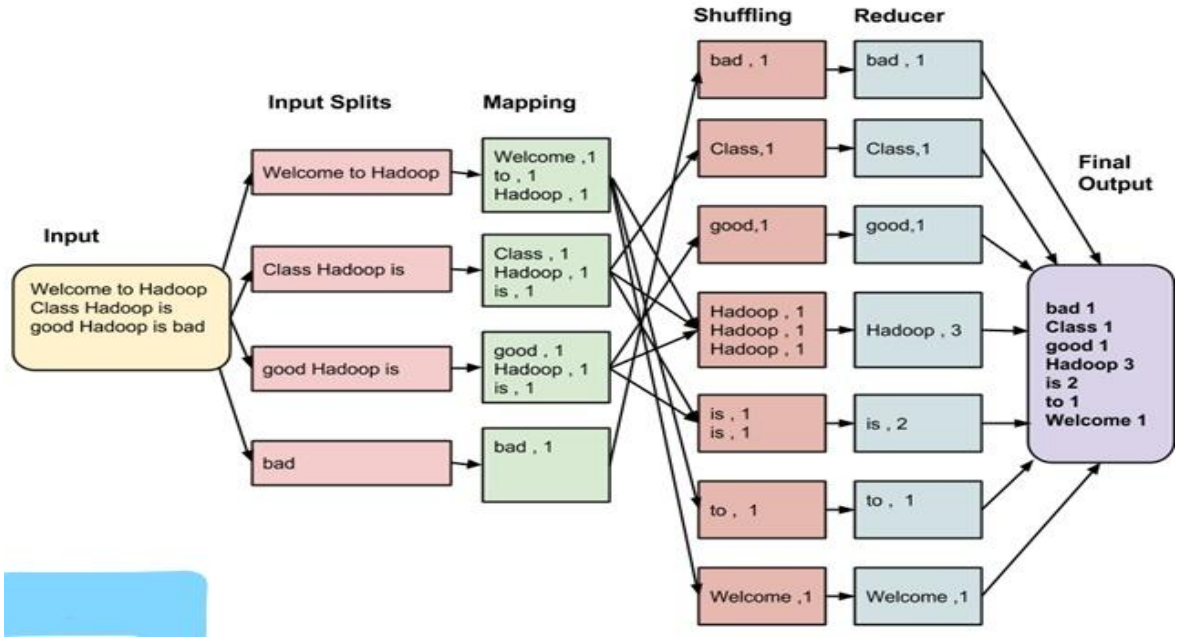
b. Preferences of previous users: The preferences of a previous user for a candidate service are extracted from his/her reviews for the service according to the keyword candidate list and domain thesaurus. Preprocess: Firstly, HTML tags and stop words in the reviews snippet collection should be removed 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.

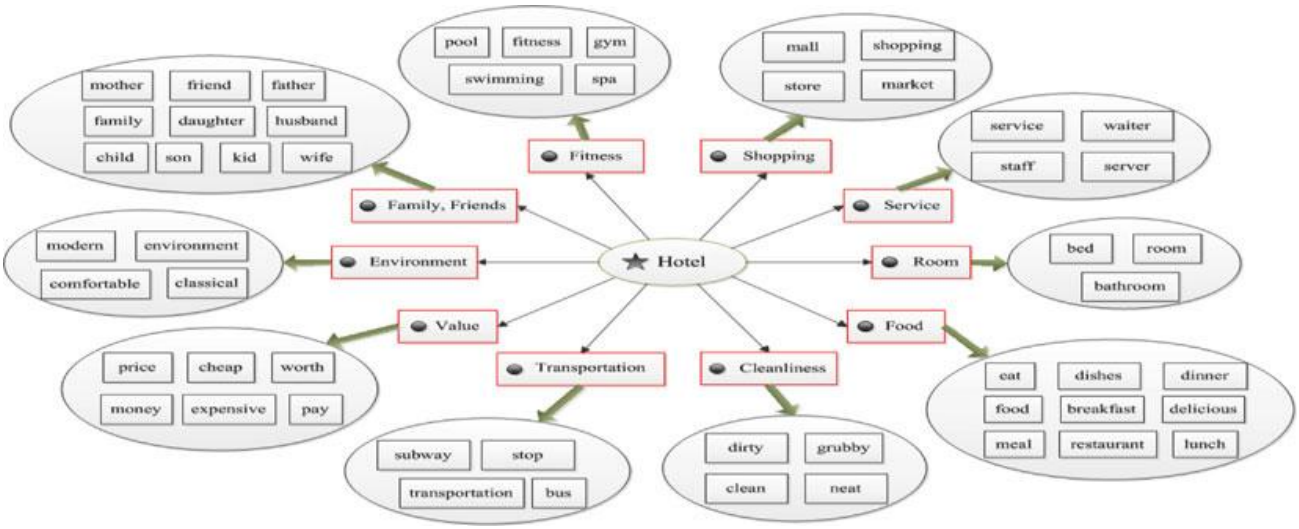
3) EXECUTION OF MAPREDUCE MODEL:-

A. The Map invocations are distributed across multiple machines by automatically partitioning the input data into a set of M 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.One of the copies of the program is special-the master. The rest are workers that are 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.assigned work by the master. There are M map tasks and Reduce tasks to assign. The master picks idle workers and assigns each one a map task or a reduce task. 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.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. 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. 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. When all map tasks and reduce tasks have been completed, the master wakes up the user program. 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



III.CONCLUSIONS AND FUTURE WORK:-

In this paper, 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. In our future work, we will do further research in how to deal with the case where term appears in different categories of a domain thesaurus from context and how to distinguish the positive and negative preferences of the users from their reviews to make the predictions more accurate.

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] W. Dou, X. Zhang, J. Liu, J. Chen, "HireSome-II: Towards Privacy-Aware Cross-Cloud Service Composition for Big Data Applications," IEEE Transactions on Parallel and Distributed Systems, 2013.

[4] 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

[5] Z. Zheng, X Wu, Y Zhang, M Lyu, and J Wang, "QoS Ranking Prediction for Cloud Services," IEEE Transactions on the Parallel and the Distributed Systems, Vol. 24, No. 6, pp. 1213-1222, 2013

[6] MR.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.

[7] M. Bjelica, "Towards TV Recommender System Experiments with the User Modeling,"

[8] IEEE Transactions on Consumer Electronics, Vol. 56, No.3, pp. 1763-1769, 2010