# UBPS :User Based Personalized Search With Big Data

**S.Ilakkiya, R.Shenbagavalli.**

*Abstract*— **The main aim of this project is to develop a User Based Personalized Search With Big Data, named UBPS, to address scalability and inefficiency problem in Big Data with traditional service recommender systems, which fails to meet users' personalized requirements and diverse Preferences. In the existing system, service recommender systems, such as hotel reservation systems and restaurant guides, They have not considered user's different preferences. and also it solve the scalability problem by dividing dataset. But their method doesn't have favorable scalability and efficiency if the amount of data grows. In the proposed system, which is based on a user-based Collaborative Filtering algorithm .It is a robust method to achieve a personalization. In UBPS ,keywords are used to indicate both of user's preferences and the quality of candidate service. It improve the scalability and efficiency of recommendation method in "Big Data" environment.**

*Index Terms*—**Big data, Distributed file system, Hadoop, Recommendation.**

## I.INTRODUCTION

**B**ig Data[1] "Big data" refers to datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyze. This definition is intentionally subjective and incorporates a moving definition of how big a dataset needs to be in order to be considered big data—i.e., we don't define big data in terms of being larger than a certain number of terabytes (thousands of gigabytes). We assume that, as technology advances over time, the size of datasets that qualify as big data will also increase. Also note that the definition can vary by sector, depending on what kinds of software tools are commonly available and what sizes of datasets are common in a particular industry. With those caveats, big data in many sectors today will range from a few dozen terabytes to multiple petabytes (thousands of terabytes).

***S.Ilakkiya*** *is with the Computer Science Department, Krishnasamy College of Engineering and Technology, Cuddalore, India.Phone:8675861006.*
***R.Shenbagavalli*** *is with the Computer Science Department, Krishnasamy College of Engineering and Technology, Cuddalore, India.Phone:8903240078;*

The Hadoop framework provides the necessary tools to implement big data. The default file system used in it is the Hadoop Distributed File System (HDFS). Hadoop framework provides support for other file systems to be used and other distributed file systems are easily pluggable to the Hadoop framework. This provides a way to use different file system according to the usage. So we have particularly focused on using different file systems so as to get the maximum benefits and thereby overcome the problems with using and relying on a single file system for handling large volume of big data.

Apache Hadoop[3] is an open-source software framework for distributed storage and distributed processing of Big Data on clusters of commodity hardware. Its Hadoop Distributed File System (HDFS) splits files into large blocks (default 64MB or 128MB) and distributes the blocks amongst the nodes in the cluster. For processing the data, the Hadoop Map/Reduce ships code (specifically Jar files) to the nodes that have the required data, and the nodes then process the data in parallel. This approach takes advantage of data locality, in contrast to conventional HPC architecture which usually relies on a parallel file system (compute and data separated, but connected with high-speed networking).

## II. SYSTEM ARCHITECTURE

A User Based personalized Search with big data in this paper, which is based on a user-based Collaborative Filtering algorithm[6].

In UBPS, 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 Map Reduce as its computing framework.

 

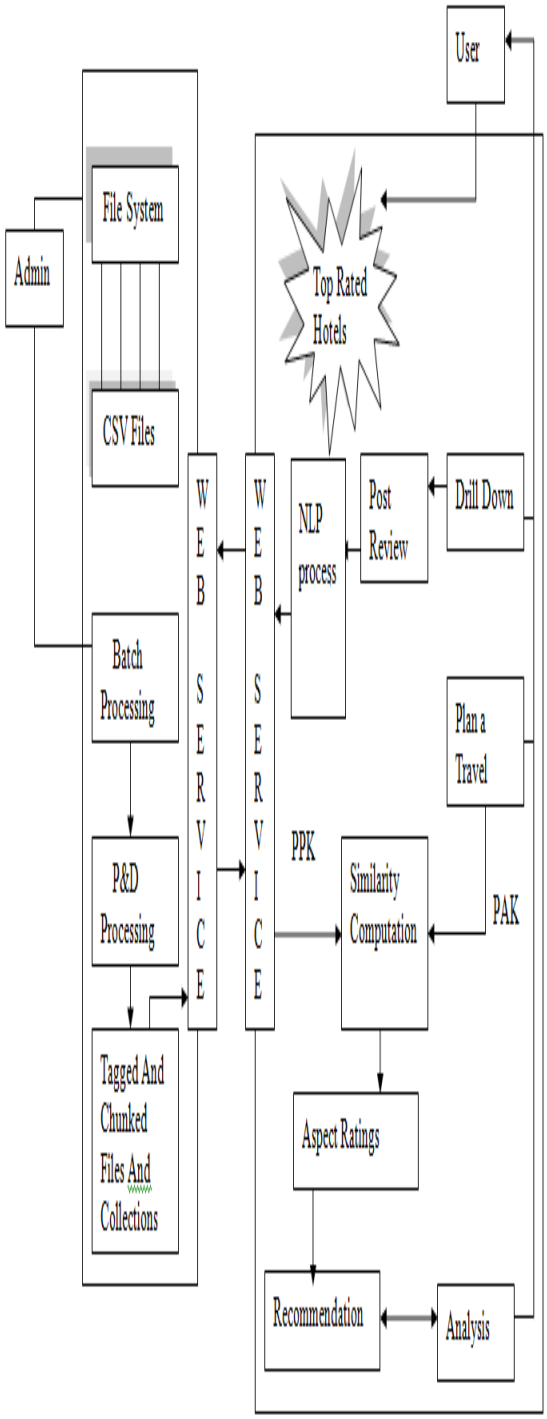


*Fig 1.System Architecture*

The architecture of the system includes ,the file system stored to the csv(comma separated value)files i.e. data to be stored in the table structured format. And the batch processing which uses the parallel and distributed processing, it is used to create tagged and chunked files collection. it is used to split up the process in the basis of NLP(Natural Language Process)and also consider the previous user keyword and active user keyword both having similarity computation provide aspect rating and recommend users the top rated service.

UPBS 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.

Service recommender systems have been shown as valuable tools for providing appropriate recommendations to users. In the last decade, the amount of customers, services and online information has grown rapidly, yielding the big data analysis problem for service recommender systems. Consequently, traditional service recommender systems often suffer from scalability and inefficiency problems when processing or analyzing 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.

Current recommendation methods usually can be classified into three main categories: content-based, collaborative, and hybrid recommendation approaches[14]. Content-based approaches recommend services similar to those the user preferred in the past. Collaborative filtering (CF) approaches recommend services to the user that users with similar tastes preferred in the past. Hybrid approaches combine content-based and CF methods in several different ways. In CF based systems, users receive recommendations based on people who have similar tastes and preferences, which can be further classified into item-based CF and user-based CF. In item-based systems; the predicted rating depends on the ratings of other similar items by the same user. While in user-based systems, the prediction of the rating of an item for a user depends upon the ratings of the same item rated by similar users. And in this work, we will take advantage of a user-based CF algorithm to deal with our problem.

## III RELATED WORKS

### *A. Big Data and Environment Module*

Huge Collection of data is retrieved from open source datasets that are publicly available from major Travel Recommendation Applications. Big Data Schemas were analyzed and a Working Rule of the Schema is determined. The CSV(Comma separated values) files were read and manipulated using Java API that itself developed by us which is

 

developer friendly ,light weighted and easily modifiable.

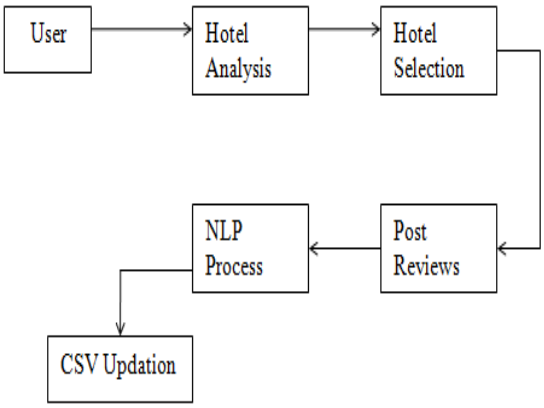


*Fig 2. Big Data Environment module*

*B. Digging in Big Data Module*

The CSV Files in distributed Systems are invoked through Web Service Running in the Server Machine of the Host Process through a Web Service Client Process in the Recommendation System. The data that Retrieved to the Recommendation Systems are provided with a clean GUI and can be queried on Demand. Each and Every process on the Recommendation Application invokes Web Service which uses light weighted traversal of data using XML. The Users can Review each hotel and can post comments also. The Reviews gets updated to the CSV Files as it get retrieved.

C.Implementation Of New Service Recommendation System Module

The Traditional View of Service Recommender Systems that shows Top-K Results are displayed with Paginations with which a user can navigate Back and Forth of the Result sets. All Services Ratings and Reviews of Each Hotels are listed. A User can Plan or Schedule a Travel highlighting his requirements in a detailed way that shows the Preference Keywords Set of the Active User. A Domain Thesaurus is built depending on the Keyword Candidate List and Candidate Services List. The Domain Thesaurus can be Updated Regularly to get accurate Results of the Recommendation System.

*D. Capture user preferences by a keyword aware approach Module*

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

*1. 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. Besides, the active user should also select the importance degree of the keywords.

*2. 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. And a review of the previous user will be formalized into the preference key-word set of User.

*E. The keyword extraction process Module*

*1.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. And the Porter Stemmer algorithm is used to remove the commoner morphological and in flexional endings from words in English.

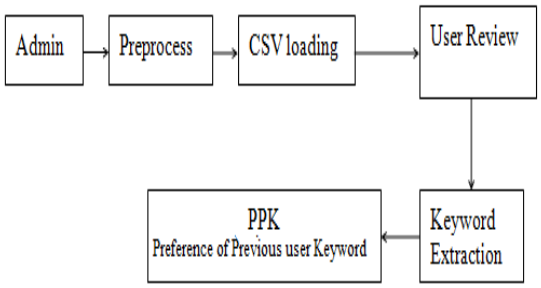


*Fig3.preprocess*

*2.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

*3.Similarity Computation*

The Third 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.

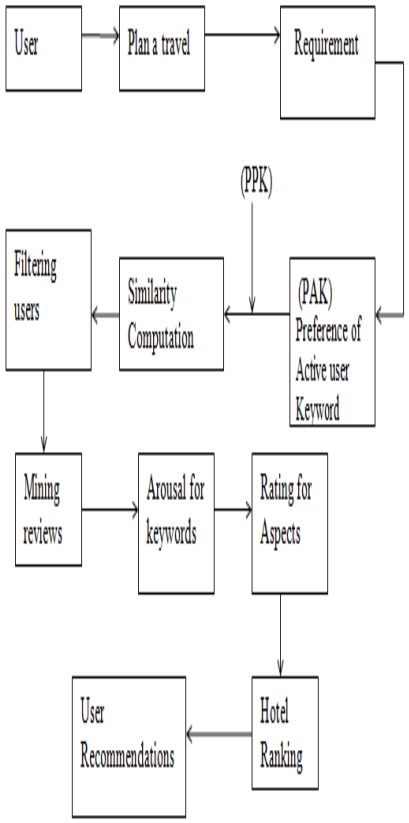


*Fig4. Similarity Computation*

*VI. Conclusion*

In this paper, we have proposed a User Based Personalized Search With Big Data, named UBPS. In UBPS, key-words are used to indicate users' preferences, and a user-based Collaborative Filtering algorithm is adopted to generate appropriate recommendations.. 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 UBPS in "Big Data" environment, we have implemented it on a MapReduce framework in Hadoop platform. Finally, the experimental results demonstrate that UBPS 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].www.cs.virginia.edu/~zaher/classes/CS656/levy.

[3]. en.wikipedia.org/wiki/Apache_Hadoop

[4] C. Lynch, "Big Data: How do your data grow?" Nature, Vol. 455, No. 7209, pp. 28-29, 2008.

[5] 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

[6] 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.

[7] 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.

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

 

[9] 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.

[10] 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.

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

[12] 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 Computing System, pp. 194-201, 1995.

[13] P. Resnick, N. Iakovou, M. Sushak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," In CSCW '94 Proceedings of the 1994 ACM conference on Computer supported cooperative work, pp. 175-186, 1994.

[14] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," User Modeling and User-Adapted Interaction, Vol. 12, No.4, pp. 331-370, 2002.

[15] G. Adomavicius, and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State of- the-Art and Possible Extensions," IEEE Transactions on Knowledge and Data Engineering, Vol.17, No.6 pp. 734-749, 2005.

[16] D. Agrawal, S. Das, A. El Abbadi, "Big Data and cloud computing: new wine or just new bottles?" Proceedings of the VLDB Endow-ment, Vol. 3, No.1, pp. 1647-1648, 2010.

**S.Ilakkiya,** received B.Tech degree in Information Technology from V.R.S College of Engineering and Technology, Arasur,Villupuram Tamilnadu, India, 2012.

She is pursuing her M.E, Computer Science in Krishnasamy college of Engineering and Technology, Cuddalore, Tamilnadu, India in the Year 2013-2015.

**R.Shenbagavalli**, working as Asst.Professor in Department of Computer Science and Engineering,cuddalore.She has 7 years experience in teaching.Her area of interest includes Analysis of algorithm,Theory of computation and Compiler Design.

She received B.E degree in Computer Science and Engineering from V.R.S College Of Engineering and Technology, Arasur, Villupuram,India, 2004 and received M.Tech degree in Computer Science and Engineering in SRM University ,Kottankulathur,Chennai, India,2011.

 