ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 1, Issue 2, pp: (79-88), Month: October 2014 – March 2015, Available at: www.paperpublications.org

Improving Service Recommendation Method on Map reduce by User Preferences and Reviews

[1]Dayanand Bhovi, [2] Mr.Ashwin Kumar

[2]Sr. Assistant professor, [1,2]Dpt. Of Computer Science & Engg. Mangalore Institute of Technology and Engineering Mangalore, Karnataka, India

***Abstract*: 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 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, to address the above challenges and 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.**

***Keywords:* recommender system, user preference, keyword, Big Data, mapreduce, Hadoop.**

I. INTRODUCTION

In recent years, the amount of data in our world has been increasing explosively, and analyzing large data sets—so-called "Big Data"— becomes a key basis of competition underpinning new waves of productivity growth, innovation, and consumer surplus [1]. Then, what is "Big Data"?, Big Data refers to datasets whose size is beyond the ability of current technology, method and theory to capture, man-age, and process the data within a tolerable elapsed time. Today, Big Data management stands out as a challenge for IT companies.

The solution to such a challenge is shifting increasingly from providing hardware to provisioning more manageable software solutions [2]. Big Data also brings new opportunities and critical challenges to industry and academia [3] [4].Similar to most big data applications, the big data tendency also poses heavy impacts on service recommender systems. With the growing number of alternative services, effectively recommending services that users preferred has become an important research issue. Service recommender systems have been shown as valuable tools to help users deal with services overload and provide appropriate recommendations to them. Examples of such practical applications include CDs, books, web pages and various other products now use recommender systems [5], [6], [7].

Over the last decade, there has been much research done both in industry and academia on developing new approaches for service recommender systems [8], [9].

Cloud Computing and Map Reduce:

Cloud computing is a successful paradigm of service oriented computing and has revolutionized the way computing infrastructure is abstracted and used. The major goal of cloud computing is to share resources, such as infrastructure, platform, software, and business process [14].

Cloud computing can provide effective platforms to facilitate parallel computing, which has gained significant attention in recent years to process large volume of data. There are several cloud computing tools available, such as Hadoop (http://hadoop.apache.org/), Mahout (http://mahout.apache.org/), MapReduce of Google [15], the Dynamo of Amazon.com [16], the Dryad of Microsoft and Neptune of Ask.com [17], etc. Among these tools, Hadoop is the most popular open source cloud computing platform inspired by Map Reduce and Google File System papers [18], which supports Map Reduce programming framework and mass data storage with good fault tolerance.

Map Reduce is a popular distributed implementation model proposed by Google, which is inspired by map and reduce operations in the Lisp programming language. More details about Map Reduce can be found in Appendix A.2. Nowadays, the trend everything as a service has been creating a Big Services era due to the foundational architecture of services computing. And servicelization is the way of offering social networking services, big data analytics, and Internet services [19] [20]. Thus the cloud computing tools aforementioned can be used to improve the scalability and efficiency of service recommendation methods in the Big Data environment.

II. LITERATURE SURVEY

Cloud computing provides services to internet by utilizing resources of the computing infrastructure to provide different services of the internet. It allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. A distributed storage system for managing structured data at Google called Bigtable. Bigtable is designed to reliably scale to petabytes of data and thousands of machines. Bigtable has achieved several goals: wide applicability, scalability, high performance, and high availability. Bigtable is used by more than sixty Google products and projects, including Google Analytics, Google Finance, Personalized Search, Google Earth and many more. In this paper a review is done to analyze the cloud performance on data stored at data centers.[1]

In this paper, propose a contextual MAB(Multi-Armed Bandit algorithms) based clustering algorithm to design and deploy recommender systems, in which both the contexts and the large item space are considered. To improve the learning speed, we consider partitioning the item cluster tree into a set of clusters. The algorithm alternates between the exploration and exploitation phases and aggregates the contextual information from a sublinear number of past arrived contexts.[2]

Leveraging community imparted data for personalized recommendation is one of the active research problems since there are rich contexts and human activities in such explosively growing data. We focus on personalized travel recommendation and show promising applications. We conduct personalized travel recommendation by considering specific user profiles or attributes. We propose a personalized travel recommendation model considering users' attributes as well as their group types and the knowledge mined from travel logs .We investigate the association of people attributes such as time, popular landmarks, etc., We also recommend the nearby location suggestions in mobile using android.[3]

In this paper we present one such class of item-based recommendation algorithms that first determine the similarities between the various items and then used them to identify the set of items to be recommended. The key steps in this class of algorithms are (i) the method used to compute the similarity between the items, and (ii) the method used to combine these similarities in order to compute the similarity between a basket of items and a candidate recommender item.[4]

In this paper, the recommendation techniques applied in Personal Program Guide (PPG). This is a system generating personalized Electronic Program Guides for DigitalTV. The PPG manages a user model that stores the estimates of the individual user'spreferences for TV program categories. This model results from the integration of different preference acquisition modules that handle explicit user preferences, stereotypical information about TV viewers, and information about the user's viewing behavior. The observation of the individual viewing behavior is particularly easy because the PPG runs on the set-top box and isdeeply integrated with the TV playing and the video recording services offered by that type of device.[5]

III. PROBLEM DEFENITION

Implement Collaborative Filtering algorithm on mapreduce is adopted to generate appropriate recommendations. To improve the scalability and efficiency of Keyword Service Recommendation Method in Bigdata Environment. Presenting a personalized service recommendation list and recommending the most appropriate services to the users effectively.

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 1, Issue 2, pp: (79-88), Month: October 2014 – March 2015, Available at: www.paperpublications.org

IV. METHODOLOGY

In this paper, 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.

Table 1 summarizes the basic symbols and notations used in this paper.

Table 1 Basic symbols and notations

Symbols	Definition
K	Keyword-candidate list, $K=\{k_1,k_2,\ldots..k_n\}$
APK	The preference keyword set of the active user
PPK	The preference keyword set of the previous user
Sim(APK,PPK)	The similarity between APK and PPK
$\rightarrow$ W_P	A preference weight vector
$\rightarrow$ W_{AP}	A preference weight vector of active user
$\rightarrow$ W_{PP}	A preference weight vector of previous user

In our method, two data structures keyword-candidate list and specialized domain thesaurus are introduced to help obtain users' preferences.

The keyword-candidate list is a set of keywords about users' preferences and multi criteria of the candidate services, which can be denoted as $K=\{k_1,k_2,..k_n\}$, *n* is the number of the keywords in the keyword-candidate list. An example of a simple keyword-candidate list of the hotel reservation system is described in Table 2. Keywords in the keyword-candidate list can be a word or multiple words related with the quality criteria of candidate services.

In this paper, the preferences of previous users will be extracted from their reviews for candidate services and formalized into a keyword set. Usually, since some of words in reviews can not exactly match the corresponding keywords in the keyword-candidate list which characterize the same aspects as the words. The corresponding key-words should be extracted as well. In this paper, we assume that specialized domain thesauruses are built to sup-port the keyword extraction, and different domain thesauruses are built for different service domains.

A domain thesaurus is a reference work of the keyword-candidate list that lists words grouped together according to the similarity of key-word meaning, including related and contrasting words and antonyms [21] [22].

TABLE 2 Keyword-candidate list of hotel reservation system

No	Keyword	No	Keyword	No	keyword
1	Service	7	Transportation	13	Airport
2	Rooom	8	Family,friends	14	Wi-fi
3	Shopping	9	Location	15	Environment
4	Cleanliness	10	Quite	16	Bar
5	Food	11	View	17	Beach
6	value	12	Fittness		

A Keyword-aware Service Recommendation Method:

The main steps of KASR are depicted in Fig.1, which are described in detail as follows.

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 1, Issue 2, pp: (79-88), Month: October 2014 – March 2015, Available at: www.paperpublications.org

(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. In this paper, an active user refers to a current user needs recommendation.

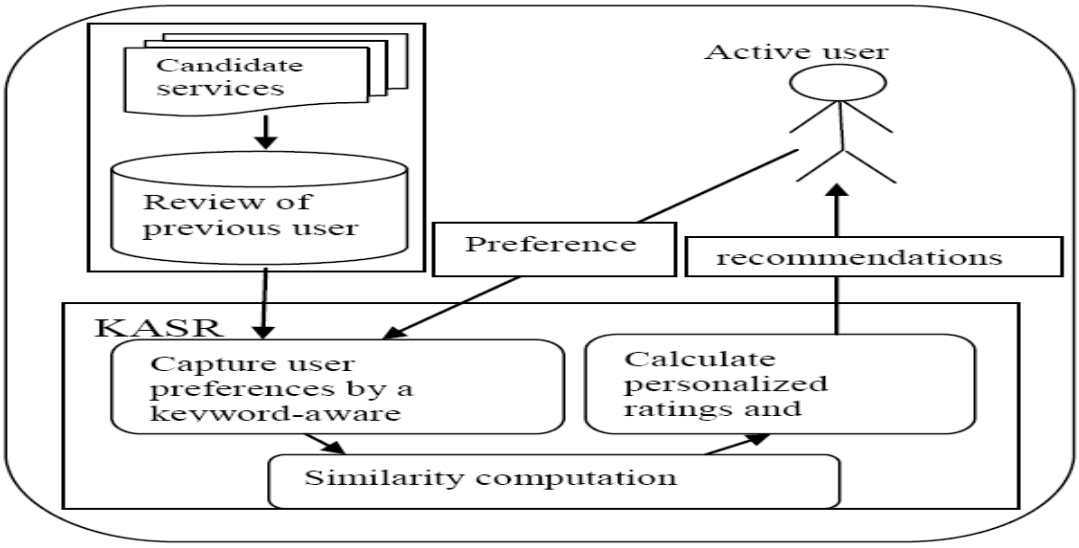

Fig:1 KASR's main steps

***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. The preference keyword set of the active user can be denoted as APK={ak_1,ak2,..ak_l} where ak_i ($1 \le i \le l$) is i^{th} keyword selected from the keyword-candidate list by the active user, *l* is the number of selected keywords.

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 him/her, which can be denoted as ,PPK={pk_1,pk_2,…pk_h} where pk_i ($1 \le i \le h$) is the *i*th keyword extracted from the review, *h* is the number of extracted keywords.

The keyword extraction process is described as follows:

a) 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 (keyword strip-ping) [23] is used to remove the commoner morphological and in flexional endings from words in English. Its main use is as part of a term normalization process that is usually done when setting up Information Retrieval systems [23].

b) 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. For example, if a review of a previous user for a hotel has the word "spa", which is corresponding to the keyword ``Fitness" in the domain thesaurus, then the keyword ``Fitness" should be contained in the preference keyword set of the previous user. If a keyword appears more than once in a review, the times of repetitions will be recorded. In this paper, 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 second 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:

a) Approximate similarity computation:

The approximate similarity computation method is for the case that the weights of the keywords in the preference keyword set are unavailable.

A frequently used method for comparing the similarity and diversity of sample sets, Jaccard coefficient, is applied in the approximate similarity computation. Jaccard coefficient is measurement of asymmetric information on binary (and non-binary) variables, and it is useful when negative values give no information. The similarity between the preferences of the active user and a previous user based on Jaccard coefficient is described as follows:

$$Sim(APK, PPK)=Jaccard(APK,PPK)=\frac{|APK\frown PPK|}{|APK\smile PPK|} \quad (1)$$

Where *APK* is the preference keyword set of the active user, *PPK* is the preference keyword set of a previous user. And the weight of the keywords is not considered in this approach.

Algorithm 1, SIM-ASC, illustrates the functionality of the approximate similarity computation method.

Algorithm1: SIM-ASC

Input: The preference keyword set of the active user *APK* The preference keyword set of a previous user *PPKj*

Output: The similarity of APK and PPKj,sim_{ASC}(APK PPK_j)

1. $sim_{ASC}(APK\ PPK_j)=\frac{|APK\frown PPK_j|}{|APK\smile PPK_j|}$

2. **Return** the similarity of *APK* and PPK_j, $sim_{ASC}(APK\ PPK_j)$

b) Exact similarity computation:

The exact similarity computation method is for the case that the weight of the keywords are available.

A cosine-based approach is applied in the exact similarity computation, which is similar to the Vector Space Model (VSM) in information retrieval [24] [25]. In this cosine-based approach, The preference keyword sets of the active user and previous users will be transformed into *n*-dimensional weight vectors respectively, namely preference weight vector, which can be denoted as W_P , *n* is the number of keywords in the keyword-candidate list, is the weight of the keyword *ki* in the keyword-candidate list. If the keyword *ki* is not contained in the preference keyword set, then the weight of *ki* in the preference weight vector is 0, i.e., *wi*=0. The preference weight vectors of the active user and a previous user are noted as

$\rightarrow_{W_{AP}}$ and $\rightarrow_{W_{PP}}$, respectively.

In this paper, we use the Analytic Hierarchy Process (AHP) model to decide the weight of the keywords in the preference keyword set of the active user. AHP method is provided by Saaty in 1970s to choose the best satisfied business role for its hierarchy nature [26]. The weight computing based on the AHP model is decied as follows:

Firstly, we construct the pair-wise comparison matrix in terms of the relative importance between each two key-words. The pair-wise comparison matrix *A*m = (*aij*) *m* must satisfy the following properties, *aij* represents the relative importance of two keywords:

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 1, Issue 2, pp: (79-88), Month: October 2014 – March 2015, Available at: www.paperpublications.org

1) $a_{ij} = 1, \quad i=j=1,2,3\ldots\ldots\ldots m$

2) $a_{ij} = 1/a_{ij}, \quad i=j=1,2,3\ldots\ldots\ldots m$ and $i \neq j$

3) $a_{ij} = a_{ik}/a_{jk}, \quad i=j=k=1,2,3\ldots\ldots m$ and $i \neq j$

After checking the consistence of the matrix, then we calculate the weight by the following function:

$$W_i = 1/m \sum_{j=1}^{m} (aij / \sum_{k=1}^{m} akj) \qquad (2)$$

Where *aij* is the relative importance between two keywords, *m* is the number of the keywords in the preference keyword set of the active user

The weight vector of the preference keyword set of a previous user can be decided by *the term frequency/inverse document frequency* (*TF-IDF*) measure [27], which is one of the best-known measures for specifying the weight of key-words in Information Retrieval.

In the *TF-IDF* approach, to calculate the preference weight vector of a previous user *u'*, all reviews by user *u'* should be collected. Here, all reviews contain the reviews by user *u'* for the candidate services and similar ser-vices not in the candidate services. The reviews should also be transformed into keyword sets respectively according to the keyword-candidate list and the domain thesaurus.

TF, the term frequency of the keyword *pki* in the preference keyword set of user *u'* is defined as

$$TF = N_{pki} / \sum_g Npki \qquad (3)$$

Where is the number of occurrences of the keyword *pki* in all the keyword sets of the reviews commented by the same user *u'*, *g* is the number of the keywords in the preference keyword set of the user *u'*.

The inverse document frequency (*IDF*) is obtained by dividing the number of all reviews by the number of reviews containing the keyword *pki*.

$$IDF = \log |R'| / |r': pki \in r'| \qquad (4)$$

Where $|R'|$ is the total number of the reviews commented by user *u'*, and $|r': pki \in r'|$ is the number of reviews where keyword *pki* appears. So the *TF-IDF* weight of the keyword *pki* in the preference keyword set of user *u'* can be decided by the following function:

$$w_{pki} = TF \times IDF = (N_{pki} / \sum_g Npki) \times (\log |R'| / |r': pki \in r'|) \qquad (5)$$

Then the similarity based on the cosine-based approach is defined as follows:

$$Sim(APK, PPK) = \cos(w_{AP}, w_{PP}) = \frac{\vec{w}_{AP} \cdot \vec{w}_{PP}}{\|\vec{w}_{AP}\|_2 \|\vec{w}_{PP}\|_2}$$

$$= \frac{\sum_{i=1}^{n} \vec{W}AP,i \times \vec{W}pp,i}{\sqrt{\sum_{i=1}^{n} \vec{W}2\ AP,i} \sqrt{\sum_{i=1}^{n} \vec{W}2\ pp,i}} \qquad (6)$$

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 1, Issue 2, pp: (79-88), Month: October 2014 – March 2015, Available at: www.paperpublications.org

→ →

Where w_{AP} and w_{PP} are respectively the preference weight vectors of the active user and a previous user. Vector $w_{AP,i}$ is the *i*-th dimension of vector w_{AP} and represents the weight of the keyword *ki* in preference keyword set *APK*,vector $w_{pp,i}$ is the *i*-th dimension of vector w_{pp} and represents the weight of the keyword *ki* in preference keyword set *PPK*.

Algorithm 2, SIM-ESC, illustrates the functionality of the exact similarity computation method.

Algorithm 2: SIM-ESC (Exact Similarity Computation)

Input: The preference keyword set of the active user *APK*

The preference keyword set of a previous user *PPKj*.

Output: The similarity of *APK* and *PPKj*,

$sim_{ESC}(APK,PPK_j)$.

1**: for** each keyword *ki* in the keyword-candidate list

2: if $k_i \epsilon$ APK **then**

3: get → by formula(2)

$W_{AP,i}$

4: else →

$W_{AP,i} = 0$

5: end if

6: if $k_i \epsilon$ PPK_j **then**

7**:** get → by formula (5)

$W_{pp,i}$

8: else →

$W_{PP,i} = 0$

9: end if

10: end for

11: get $sim_{ESC}(APK,PPK_j)$*.by formula(6)*

12: **return** the similarity of *APK* and *PPKj*

$sim_{ESC}(APK,PPK_j)$.

(3) Calculate personalized ratings and generate recom-mendations:

Based on the similarity of the active user and previous us-ers, further filtering will be conducted. Given a threshold δ, if sim(*APK*,PPK_j)< δ, the preference keyword set of a pre-vious user *PPKj* will be filtered out, otherwise *PPKj* will be retained.

Finally, a personalized service recommendation list will be presented to the user and the ser-vice(s) with the highest rating(s) will be recommended to him/her.

We use a weighted average approach to calculate the personalized rating *pr* of a service for the active user.

$pr = \bar{r} + k \sum_{ppkj \epsilon R\ cap} \mathrm{sim}(APK, PPKj) \times (rj - \mathrm{r\ bar})$

$k = 1/\sum_{ppkj \epsilon R\ cap} \mathrm{sim}(APK, PPKj)$

where sim(*APK*,*PPK*$_j$) is the similarity of the preference keyword set of the active user and the preference keyword set of a previous user *PPKj*; multiplier *k* serves as a normalizing factor, ˆR(R cap) denotes the set of the remaining preference keyword sets of previous users after filtering; *rj* is the rating of the corresponding review of *PPKj*, and r bar is defined as the average ratings of the candidate service.

we can calculate the personalized ratings of all candidate services for the active user. Then we can rank the services by the personalized ratings and present a personalized service recommendation list to him/her. we assume that the services with higher ratings are more preferable to the user. So the service(s) with the highest rating(s) will be recommended to the active user. we can recommend the Top-K services to the user.

Algorithm 3: Basic Algorithm of KASR

Input: The preference keyword set of the active user *APK*

The candidate services WS={ws$_1$,ws$_{2,}$……ws$_N$}

The threshold δ in the filtering phase The number *K* .

Output: The services with the Top-K highest ratings

{t*ws1*, t*ws2*, …, t*wsK*}.

1: **for** each service ws$_i$ ϵ WS

2: ˆR = Φ sum = 0, r = 0

3: **for** each review *Rj* of service *wsi*

4: process the review into a preference keyword set *PPKj*

5: if PPKj ∩ APK = Φ then

6: insert *PPKj* into ˆR

7: end if

8: end for

9: **for** each keyword set PPK j ϵ ˆR

10: sim$(APK, PPKj) =$ SIM$(APK, PPKj)$

11:if sim$(APK, PPKj) < \delta$ then

12: remove *PPKj* from ˆR

13: **else** *sum=sum+1*, $r=r+r_j$

14: end if

15: end for

16: $\bar{r}$= r/sum

17: get pr_i by formula (7)

18: end for

19: sort the services according to the personalized ratings pr$_i$

20: **return** the services with the Top-*K* highest ratings

{tws_1, tws_2… tws_k}

Algorithm 3 illustrates the basic algorithm of KASR. The input contains the preference keyword set of the active user *APK*, the candidate services WS={ws$_1$,ws$_2$,…ws$_N$} the threshold δ in the filtering phase, and the number K. In line 2, ˆR is used to store the remaining preference keyword sets of previous users, and *sum* is to record the number of the remaining

preference keyword sets of previous users. Line 3 to line 8 is used to process each review of the previous users into the corresponding preference keyword sets, and then do a simple filtering to filter out the reviews unrelated with the active user's preferences. Line 9 to line 15 are to calculate the similarity of *APK* and *PPKj*, and then filter out the keyword set *PPKj* whose similarity with *APK* is less than the threshold δ.

In this paper, there are two methods to calculate the similarity: approximate similarity computation (see Algorithm 1) and exact similarity computation (see Algorithm 2). Line 16 to line 18 is to calculate the personalized ratings of the candidate services for the active user. Finally, line 19 and line 20 is to sort the candidate services according to the personalized ratings and recommend the services with the Top-K highest ratings to the active user.

V. RESULTS AND DISCUSSION

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 datasize). Fig. 2 shows the speedup of KASR (Here, KASR-ESC method is adopted in the scalability experiment). From Fig. 2, 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 datasize 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 idealspeedup. The experimental result shows that KASR on Map-Reduce in Hadoop platform has good scalability over Big Data and performs better with larger dataset.

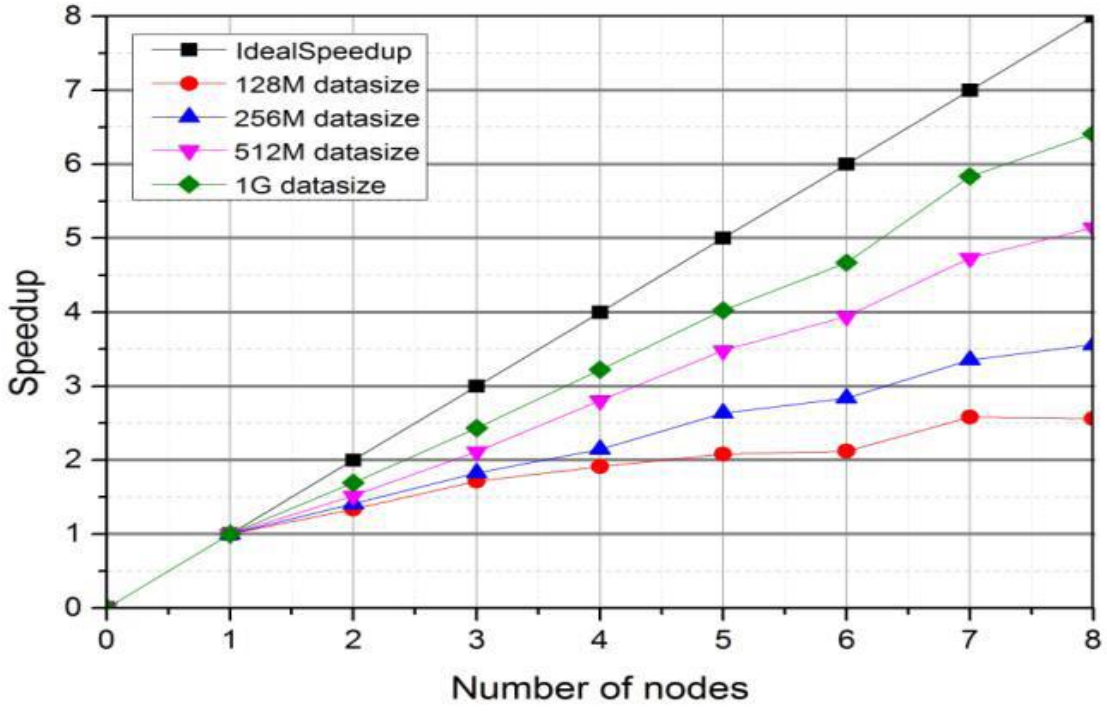

Fig. 2 Speedup of KASR

Overall, these experimental results show that KASR per-forms well in accuracy and KASR on Mapreduce framework has good scalability in Big Data environment.

VI. CONCLUSION

In this paper 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.

VII. FUTURE SCOPE

Implement Collaborative filtering algorithm on mapreduce is adopted to generate appropriate recommendations. To improve the scalability and efficiency of Keyword Service Recommendation Method in Bigdata Environment,to distinguish the positive and negative preferences of the users from their reviews to make the predictions more accurate.

REFERENCES

[1] Key Word Aware Service Recommendaion Method.

[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.

[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 Mining People Attributes and Travel Group Types From Community-Contributed Photos". IEEE Transactions on Multimedia, Vol. 25, No.6, pp. 1283-1295, 2012.www.ijecse.org.

Author's Profile:

Dayanand Bhovi completed the bachelors degree in Informaton Science &Engineering from University of VTU and presently pursuing Masters in Engineering in Computer Science & Engineering at Mangalore Institute of Technology, Mangalore.

Ashwin kumar completed bachelors and masters degree in Computer Science and Engineering. Currently working as Sr. Assistant professor in Mangalore Institute Technology, Mangalore.