International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 4, April - 2014

Building Personalised Recommendation System With Big Data and Hadoop Mapreduce

S. Vinodhini[1]
[1]Post Graduate Student, Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Chennai, India

V. Rajalakshmi[3]
[3]Assistant Professor, Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Chennai, India

B. Govindarajalu[2]
[2]Professor and Head, Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Chennai, India

Abstract **- Recommender systems are found in many e-commerce applications today. Recommender systems usually provide the user with a list of recommendations that they might prefer, or supply predictions on how much the user might prefer each item. Two common approaches for providing recommendations are collaborative filtering and content based filtering. By combining these two approaches, hybrid recommendation systems can be developed that considers both the ratings of the user and the item's feature to recommend the items to the user. The features of limited amount of data can be analyzed with the existing data analysis tools but when considering an e-book dataset of size in Terabytes, a big data analysis tool such as Hadoop is used. Hadoop is a software framework for distributed processing of large data sets. Hadoop uses MapReduce paradigm to perform distributed processing over clusters of computers to reduce the time involved in analyzing the item's feature (keywords of a book). The proposed system is reliable and fault tolerant when compared to the existing recommendation systems as it collects the ratings from the user to predict the interest and analyses the item to find the features. The system is also adaptive as it updates the rating list frequently and finds the updated interest of the user. Experimental results show that the proposed system is more accurate than the existing recommender systems.**

***Keywords: Recommendation System, Hadoop, Big Data, MapReduce, Keywords and stop words*.**

1. INTRODUCTION

Big data analysis is one of the upcoming disciplines in data mining where the large unstructured data that is very difficult to store and retrieve in an efficient manner. Big data doesn't refer not only to exabytes or petabytes of data. When the amount of data that is needed to be processed is greater than the capacity of the system, then it refers to Bigdata. The three perspectives of big data are volume, velocity and variety [1]. Volume refers to the amount of data that is being processed. It has moved to Zettabytes and Petabytes as of 2014 and expected to increase in future. Velocity refers to the speed at which the data can be processed with minimal error rate. Variety refers to all types of data starting from unstructured raw data to semi-structured and structured data which can be easily analyzed and used for the process of decision making and predictive analysis.

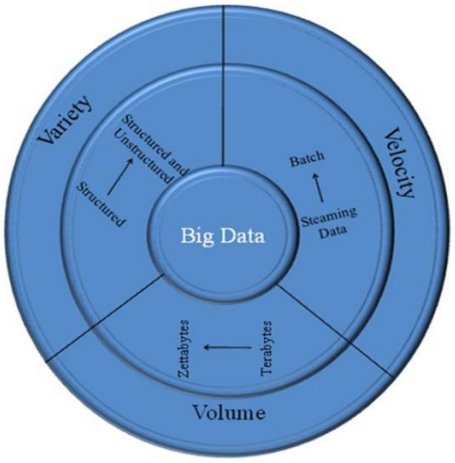

Fig. 1. Three Characteristics of Big Data

This exponential growth in data has lead to many vital challenges in business. Existing tools have become inadequate to process such large sets of data. In order to overcome this, Google introduced a programming model called MapReduce [2]. This system was considered as a great evolution in the field of data mining. Soon after, a tool called Hadoop was introduced. Hadoop is a tool used for analyzing large sets of data using distributed clusters. This tool can also be used for parallel programming. There are many big data analysis tools but the key terms that made Hadoop distinct from others are:

Accessible-Hadoop can run on large and distributed clusters of nodes or on some services of cloud computing such as Amazon's Elastic Compute Cloud (EC2).

Robust-Hadoop is architected with the capacity to withstand or tolerate hardware malfunctions such as shut

down or data loss. It can gracefully handle most such failures with the help of secondary Namenode.

Scalable-Hadoop can be scaled to add more nodes once the multi node cluster has been set up.

Simple- users can easily write parallel code with the help of Hadoop.

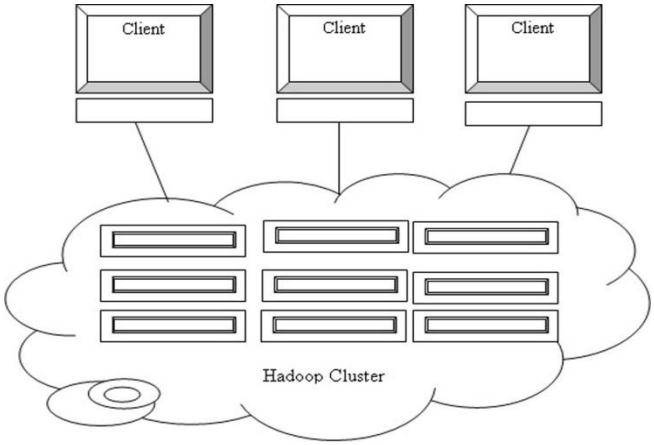

Fig. 2. Multinode Cluster

MapReduce is a programming model where large sets of data can be distributed among the nodes of a cluster and processed parallel. There are two types of node such as Master node and Slave node. Master node allocates the tasks to the slave and slave nodes carries out the job assigned to it. Master node then collects the results. This model has two main steps which are 1) Map - Distribute the job among the slaves and 2) Reduce – Collect the results.

Recommender systems have become popular from the last decade. Since the number of products has grown in number, the need for recommender systems has also increased. Recommender system tries to predict the interest of a user and recommend products that match their interest as accurately as possible. Also, e-commerce business will be profited by the increase of sales which will obviously occur when the user is presented with more items that he/she would likely found to match the interest. There are two common approaches in building a recommendation system. One is Collaborative filtering that builds a model from a user's past behavior as well as similar decisions made by other users to predict items that the user may have an interest in. The other is Content-based filtering where the characteristics of an item are analyzed to recommend additional items to the user.

The following sections are arranged as such chapter 2 includes the works related to the proposed system; chapter 3 includes the design of the system along with the modular description of the proposed system. Chapter 4 depicts the implementation setup and results obtained for the proposed system along with the performance evaluation. Chapter 5 gives a brief description about the proposed system and future extension that can be done.

II. LITERATURE SURVEY

Existing recommendation system recommends books to the user based on the book name and the ratings given by that user to the book or based on the number of views for that book. Fuzhi Zhang et al (2010), proposed a two-stage algorithm that uses location of the users to predict the interest. K-means algorithm is used to cluster the users based on the profile which is collected during the user sign up. But predicting the concept of a book only with the book name reduces the accuracy of the system. V. Mohanraj et al (2012) uses the concept of ontology to predict the interest of the user. The system was self adaptive and predicted the future browsing pattern of the user. Ozgur Cakir et al (2012) developed a recommendation system using association rules. Apriori algorithm is used to generate the rules for recommendation. The basket ratio which is the ratio between the number of items viewed to the number of items added to the shopping cart is increased in this method.

Boban Vesin et al (2012) developed a recommendation system termed as PROTUS (PRogramming TUtoring System) that recommended courses to the students. The courses are usually recommended to the students based on their age and domain of study but in this system semantic web technology concepts are used. Navigation patterns are obtained from the past history of the student and from that pattern, future recommendations are made. Konstantin Shvachko et al (2010) made a study on the Hadoop distributed File System. The study stated that by distributing the storage and computation across the machines of a cluster, the computational time can be reduced for analyzing big data when compared to single node processing.

Emmanouil Vozalis et al made an analysis on the types of recommendation algorithms that are in existence. Item-based recommendation is a method in which two users who have rated a item are separated and the similarity index is computed among them. When the similarity index is greater than the threshold, then similar items are recommended to them. A model which uses Collaborative filtering algorithm for supervised learning was developed. This model classifies even the new unseen item. According to this model, there are only two classes C1:like C2: dislike. Content-Boosted Collaborative Filtering utilizes Contentbased Filtering to fill in the missing ratings from the initial user-item matrix. It then employs classic Collaborative Filtering techniques to reach a final prediction.

CaiNicolas Ziegler et al (2005) proposed a recommendation system that considers a concept called topic diversification. According to this concept, the list of

top n recommendation will be balanced as the users's extended interest will also be taken into account. Thus the user will not be bored upon the similar kind of recommendations often made. The concept of User-based Collaborative filtering and Item-based Collaborative filtering are combined and the recommendations are made.

Brian McFee et al(2012) developed a recommendation system for music by learning the contenet similarity. It used content based similarity method initially and then collaborative similarity method is imposed on the results. It avoided the cold start problem and the overhead of query-to-answer technique.

III. SYSTEM DESIGN

The idea of this system is to develop a recommendation engine that can recommend books to the users with increased accuracy by analyzing the interest of the user and features of the books. A hybrid recommender system is developed that gets its input from the user in the form of ratings. This ratings list and the profile of the user are the key terms used to predict the interest of the user. The data set considered is a large set of books which is a big data. In order to analyze the features of the book set that is so large, we go for a tool named Hadoop.

MapReduce programs have been written to find the feature. Preprocessing tasks are also performed in order to eliminate the stop words and to generate the keywords for the book. The overall architecture of the developed system is given below. It can be divided into 4 modules.

Initially the data set which are the ebooks are collected from the website www.bookza.org and then they are preprocessed. Preprocessing task involved steps such as converting the pdf format of books to word, removing stop words, generating word count and finally extracting keywords from the word count file. These keywords are collected for each book and used while recommending books to the users.

An application is created to do all these preprocessing works. This application is created with JAVA and MapReduce. The recommendation system is developed which will recommend books to the user. The user must create an account in the system. During the creation of the account, a set of 10 books are given and user is asked to rate the books. The ratings given initially will be analyzed to provide further recommendations to the user. These recommendations will be provided when the user logs in with the password for the next time.

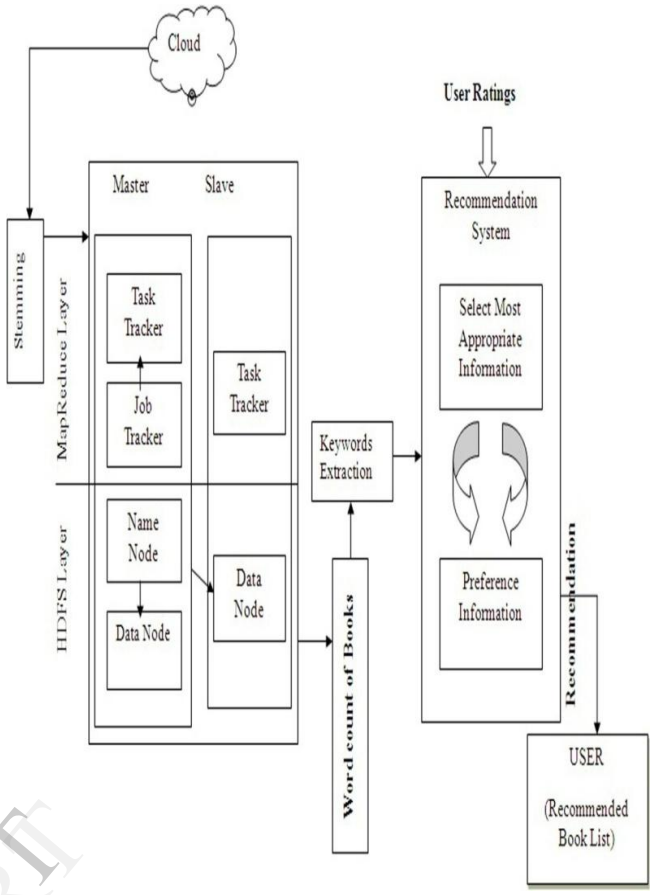

Fig. 3. Architecture diagram for proposed system

A) *Dataset Collection*

Big Data (i.e) a large set of books which is distributed among nearly 20 domains are collected. These books are collected from the website www.bookza.com. The domains with which the website is created are

TABLE 1. DOMAINS OF THE DATASET

DOMAINS OF THE DATASET (EBOOKS)	
COMPETITIVE EXAM	DATABASE MANAGEMENT SYSTEM
DATA STRUCTURES	WIRELESS SENSOR NETWORKS
HORROR	IMAGE PROCESSING
CRYPTOGRAPHY AND NETWORK SECURITY	SOFTWARE ENGINEERING
COMICS	DATA MINING
CHEMISTRY	FANTASY
FICTION	WEB TECHNOLOGY
SYSTEM SOFTWARE	COOKING
OPERATING SYSTEM	COMPUTER ARCHITECTURE

B) Preprocessing by Stop words removal

The initial input is set of books in the form of a pdf file. These pdf files must be converted into text files because Hadoop can read text files only. If it is a single book, any pdf to text converter tool can be used. But it is a large set of books. So a program that can convert the pdf files to text in reduced time period is written. The pseudocode of that program is given below

```
PROCEDURE:
To convert pdf to text

Create a pdf reader
pNum: number of pages in the document
for page=1 to pNum
        text : pdfTextExtractor.getTextFromPage(instance_of_pdfReader ,page);
        write the text to the text file
end for
```

The text file that is obtained from the above process is used to remove the stop words present in the file. The final objective is to generate keywords from the book where the existence of irrelevant words is not a good sign. Thus the stop words are removed from the text file. The pseudocode for removing stop words is given below

```
PROCEDURE:
Remove Stop words

tokenize(textFile);
word : word in the text file
StopWords[]={"is","was","how","has","had",…… ,"you"};
While(word.hasNext())
      for i=0 to StopWords.length
          if(StringCompare(word, StopWords))
                word.remove();
          end if
      end for
end while
```

C) Multi-node Cluster Setup for Hadoop

In order to run the MapReduce program parallel in more than 2 machines, we setup a Hadoop cluster with 5 nodes. This can be done by setting up Hadoop in Ubuntu by allocating an Hduser for Hadoop. But the better option was to go with HortonWorks Sandbox. HortonWorks is considered to be better because of its easy installation in Windows and also it's a complete package of all the pre-requisites that are needed to be installed before the installation of Hadoop. The sandbox includes the core Hadoop components (HDFS and MapReduce), as well as all the tools needed for data ingestion and processing. In order to run Hadoop in HDP (HortonWorks Data Platform) environment, some supporting tools like putty, WinSCP are needed.

1) Putty

Putty is an application used for transferring files between the master and slave. The master node provides the input data and instructs the slave to perform a task.

2) WinSCP

WinSCP is used for secure file transfer between a master and the slaves. Inorder to authenticate the slave that will connect to the master, a protocol named SSH (Secure SHell) protocol is needed. This protocol ensures secure login and logout between the master and the slaves.

The pseudocode that is written to generate word count is given below

```
PROCEDURE:
Map
For each book in bookset
{
    T=tokenize(book);
    For each token in T
    {
      WordCount[token]++;
    }
}
sendToReduceStep(WordCount);
```

```
PROCEDURE:
Reduce
For each WordCount received from
MapStep
{
Add(totalWordCount,WordCount);
}
```

D) Keywords Generation

The word count of the preprocessed book is stored in a text file. This text file is used to extract the keywords for that book. In order to do this, a threshold of the value in <key,value> pair is taken and the keys that have their values greater than that threshold is filtered out. The pseudocode is as follows

```
PROCEDURE:
Keyword generation from wordcount of Hadoop

Set string: word count file generated from Hadoop
if(string is not Null)
        String1=string.split("\t");
end if
if(string1[0] is not Null)
        if(string1[1]>threshold_value)
                map.put(string1[0], Double.parseDouble(string1[1]));
        end if
end if
for (Map.Entry<String, Double> entry : map.entrySet())
        write(getKey());
end for
```

E) Building Recommendation System

A recommendation engine is created as GUI to make the user interact with the system in an easy way. The user can login and logout of the system, can rate books, can view and download the books from the system. This recommendation system is created with two types of privileges 1) admin 2) user

PROCEDURE:
Admin's rights

- Authenticate the users who signed up new
- View, edit and download the books and the corresponding details
- Upload new books by specifying all the details

PROCEDURE:
User's rights

- To view and download the books recommended
- Rate the books recommended
- Search the books from the database with some keywords
- Specify the region and view books that belong to that region

Recommendation system that was developed has a special feature called Region Aggregation (RA). The user is asked to enter the details about the country, state and city.

If(country="India" && state="TamilNadu" && City="Chennai")→Book="Holy Thirukkural")
If(country="India" && state="AndhraPradhesh " && City="Hyderabad")→Book="Classical Telugu Poetry")

Users are clustered using K-means clustering algorithm. The profile of the users is considered to form the cluster. For example:

TABLE 2. TABLE FOR K-MEANS CLUSTERING OF USERS

Profile of Users					Preference		
User id	Age	Income	Gender	Marital Status	Service	Rice	environment
139****5685	25	2000	female	N	mid	low	high
139****5455	40	5000	female	Y	high	high	high
150****5679	24	3000	male	N	mid	mid	mid
133****5489	55	5500	male	Y	high	high	high

IV. IMPLEMENTATION RESULTS

This section explains the implementation that is done in the system. The implementation is done with tools such as Hadoop, HortonWorks Sandbox, Putty, WinSCP, VirtualBox and programming is done in java and MapReduce. Here a single book is taken as an input and the respective results for each module are shown. Initially a book in pdf format is taken as an input. This input file is converted into text with the help of the program for which the pseudocode is given above. Fig 4 describes the java application that was developed to convert a pdf file to text file, to remove stop words and to extract keywords from the book with the help of Hadoop MapReduce program. The path is specified and linked in the program between the various tasks.

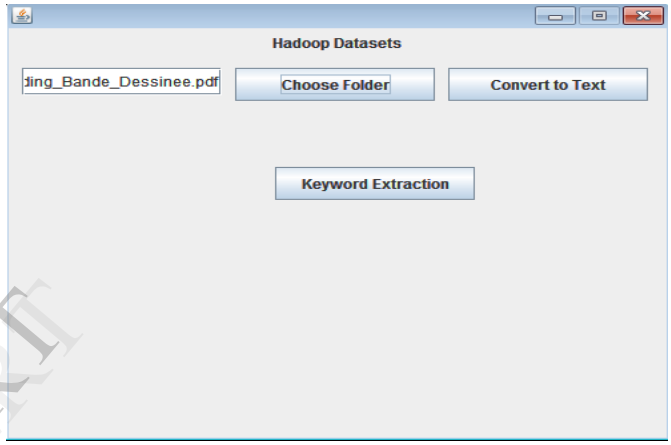

Fig. 4. The java application developed to generate keyword for a book

From the text file obtained, the word count is generated using the Hadoop MapReduce program. The output of the program will be in the format of <key,value> pair. A sample of the word count generated from a book on politics is given below

<community	146>
<citizens	74>
<divided	50>
<freedom	98>
<government	157>

The keywords are extracted from the word count file by setting a threshold and entered inside the keyword field of the recommendation system while uploading a book. Thus the keywords of the book are

Keywords: Community-citizens-freedom-government

Admin is responsible to upload new books or delete the outdated books from the database. The uploading process of books can be done via the following tab of GUI created

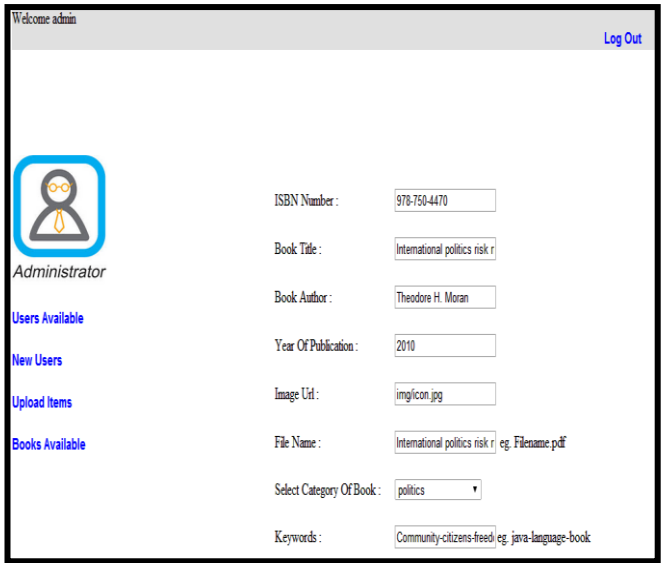

Fig. 5. Upload books

The recommendations to the user will be made in the following format

Fig. 6. View and download the recommended books

Region aggregation is implemented here where the comic book that has rights to be distributed in India and the book that is mostly read in Chennai is given as recommendation. Ratings are given out of 10. If the previous rating was 8 and the new rating by a new user was 4, then the rating of the book would change to 6. Average of the previous rating and new rating is taken.

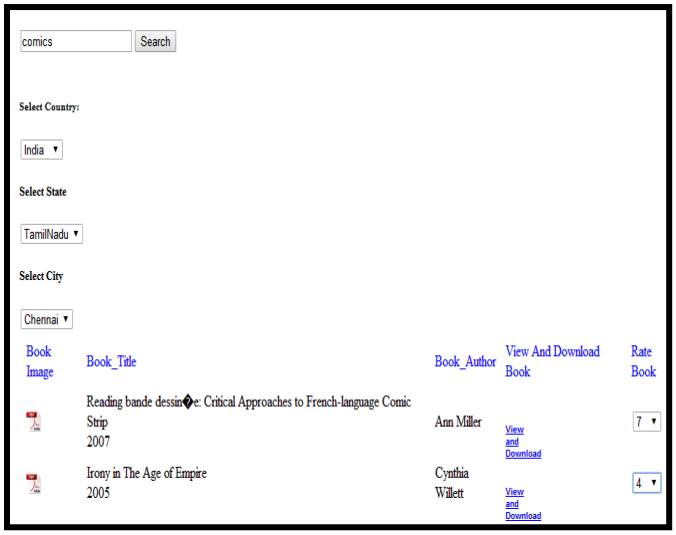

Fig. 7. Region aggregation and search by keyword

Performance Evolution

Basically performance of a recommender system can be measured using accuracy. In this work, performance of proposed system is evaluated in terms of calculating accuracy and precision These values can be calculated easily by forming a confusion matrix which is also known as contingency table. This confusion matrix contains True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). Precision refers positive prediction value and accuracy can be calculated with the following formula.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

$$\text{Precision} = \frac{TP}{(TP + FP)}$$

The following table describes the confusion matrix that is formed while considering a set of 100 books and when offline evaluations are made.

TABLE 3.CONFUSION MATRIX OF PROPOSED SYSTEM

CONFUSION MATRIX	Preferred	Non Preferred
Recommended	12	3
Not recommended	5	80

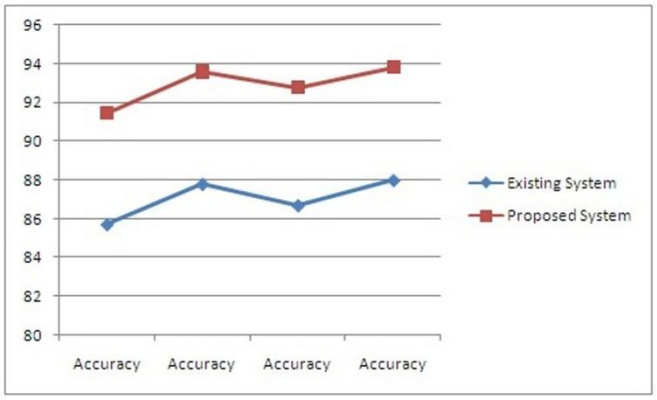

Fig. 8. Graph plotted to depict the accuracy variations in percentage

V. CONCLUSION

Along over two decades of research and commercial development, recommender systems have proved to be a successful technology to overcome the information overload that burdens users in modern online media. According to a survey, 62% of the customers who notice the recommendations purchase the recommended products. The key driver for this success is to provide more relevant recommendation by incorporating customer interest. These recommendations can be provided more accurately by analyzing the features of the product to be recommended and matching it with the interest of the user accordingly. This recommendation system is to be built for recommending the books to the users according to their interest. This work can be extended for movies recommendation, music recommendation, website recommendation etc. But while dealing with website recommendation, the total number of views for that website should also be considered as a metric for providing accurate recommendations.

REFERENCES

[1] Asela Gunawardana and Guy Shani , "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks", Journal of Machine Learning Research , Vol. 10, pp. 2935-2962, 2009.
[2] Boban Vesin., Mirjana Ivanovic., Aleksandra Klasnja-Milic and Zoran Budimac (2012), 'Ontology-based semantic recommendation in programming tutoring system', Journal on expert systems with applications, Vol. 39, pp 1229-12246.
[3] CaiNicolas Ziegle.R, Sean M. McNee., Joseph A. Konstan and Georg Lausen (2005), 'Improving Recommendation Lists Through Topic Diversification', International World Wide Web Conference Committee (IW3C2), ACM, pp. 5959-30-469.
[4] Feng Xie., Zhen Chen., Hongfeng Xu., Xiwei Feng and Qi Hou (2013), 'TST: Threshold Based Similarity Transitivity Method in Collaborative Filtering with Cloud Computing', IEEE Transactions on Tsinghua Science and Technology, Vol. 18, No. 3, pp 318-327.
[5] V. Mohanraja., M. Chandrasekaran., J. Senthilkumar., S. Arumugam and Y. Suresh (2012), 'Ontology driven bee's foraging approach based self adaptive online recommendation system', The journal of systems and software, Vol. 85, pp. 2439-2450.
[6] Ozgur Cakira and Murat Efe Aras (2013), 'Recommendation engine by using association rules', Journal of Social and Behavioral Sciences, Vol. 62, pp. 452 – 456.
[7] 'Hadoop', http://hadoop.apache.orgcore/docs/current/mapred_tutorial.html.
[8] 'Google dataset for book', http://books.google.com/ngrams/graph?content=Albert+Einstein%2CSherlock+Holmes%2CFrankenstein&year_start=1800&year_end=2000&corpus=15&smoothing.
[9] Fuzhi Zhang, Huilin Liu, Jinbo Chao, "A Two-stage Recommendation Algorithm Based on K-means Clustering In Mobile E-commerce", Journal of Computational Information Systems, Vol. 6, Issue 10, pp. 3327-3334, 2010.
[10] Taek-Hun Kim, Young-Suk Ryu, Seok-In Park, and Sung-Bong Yang, "An Improved Recommendation Algorithm in Collaborative Filtering", Department of computer science yonsei university.
[11] Konstantin Shvachko, Hairong Kuang, Sanjay Radia and Robert Chansler, "The Hadoop Distributed File System", IEEE , pp. 978-1-4244-7153-9/10, 2010.
[12] Emmanouil Vozalis, Konstantinos G. Margaritis, " Analysis of Recommender Systems' Algorithms", conference proceeding of IEEE.
[13] Brian McFee, Luke Barrington and Gert Lanckriet, "Learning Content Similarity for Music Recommendation" IEEE Transactions on Audio, Speech, and Language Processing, Vol. 20, No. 8, 2012.
[14] Paul C.Zikopolus and Chris Eaton, " Understanding Big Data Analytics for Enterprise Class Hadoop and Streaming Data", thesis, 2013.
[15] Chuck Lam, "Hadoop in Action", thesis, 2013.