ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

Vol. 3, Issue 5, May 2015

# Reduction of Online Execution Time for Big Data Application

P.S.Manikandan, M.Vimalan, Prof K.Ravikumar

M.E Final Year, Dept. of CSE., Rrase Engineering College, Chennai, India

Assistant Professor, Dept. of CSE., Rrase Engineering College, Chennai, India

H.O.D, Dept. of CSE., Rrase Engineering College, Chennai, India

**ABSTRACT**: An increasing number of services are emerging on the Internetby service computing and cloud computing. As a result, service-relevant data become too big to be effectively processed by traditional approaches. In view of this challenge, Reduction of online execution time for big data application is proposed in this paper, which aims at recruiting similar services in the same clusters to recommend services collaboratively. Technically, this approach is enacted around two stages. In the first stage, the available services are divided into small-scale clusters, in logic, for further processing. At the second stage, a collaborative filtering algorithm is imposed on one of the clusters. Since the number of the services in a cluster is much less than the total number of the services available on the web, it is expected to reduce the online execution time of collaborative filtering. At last, several experiments are conducted to verify the availability of the approach.

**KEYWORDS**: big data application, cluster, collaborative filtering, mash up.

## I. INTRODUCTION

Big data has emerged as a widely recognized trend, attracting attentions from government, industry and academia [1]. Generally speaking, Big Data concerns large-volume, complex, growing data sets with multiple, autonomous sources. Big Data applications where data collection has grown tremendously and is beyond the ability of commonly used software tools to capture, manage, and process within a "tolerable elapsed time" is on the rise [2]. The most fundamental challenge for the Big Data applications is to explore the large volumes of data and extract useful information or knowledge for future actions [3].

With the prevalence of service computing and cloud computing, more and more services are deployed in cloud infrastructures to provide rich functionalities [4]. Service users have nowadays encounter unprecedented difficulties in finding ideal ones from the overwhelming services. Recommender systems (RSs) are techniques and intelligent applications to assist users in a decision making process where they want tochoose some items among a potentially overwhelming set of alternative products or services. Collaborative filtering (CF) such as item- and user-based methods are the dominant techniques applied in RSs [5]. The basic assumption of user-based CF is that people who agree in the past tend to agree again in the future.

Different with user-based CF, the item-based CF algorithm recommends a user the items that are similar to what he/she has preferred before [6]. Although traditional CF techniques are sound and have been successfully applied in many e-commerce RSs, they encounter two main challenges for big data application: 1) to make decision within acceptable time; and 2) to generate ideal recommendations from so many services. Concretely, as a critical step in traditional CF algorithms, to compute similarity between every pair of users or services may take too much time, even exceed the processing capability of current RSs. Consequently, service recommendation based on the similar users or similar services would either lose its timeliness or couldn't be done at all. In addition, all services are considered when computing services‟ rating similarities in traditional CF algorithms while most of them are different to the target service. The ratings of these dissimilar ones may affect the accuracy of predicted rating.

Copyright to IJIRCCE
DOI: 10.15680/ijircce.2015.0305154

## II. RELATED WORK

Clustering methods for CF have been extensively studied by some researchers. Mai et al. [37] designed a neural networks-based clustering collaborative filtering algorithm in e-commerce recommendation system. The cluster analysis gathers users with similar characteristics according to the web visiting message data. However, it is hard to say that a user’s preference on web visiting is relevant to preference on purchasing. Mittal et al. [38] proposed to achieve the predictions for a user by first minimizing the size of item set the user needed to explore. K-means clustering algorithm was applied to partition movies based on the genre requested by the user. However, it requires users to provide some extra information. Li et al. [39] proposed to incorporate multidimensional clustering into a collaborative filtering recommendation model. Background data in the form of user and item profiles was collected and clustered using the proposed algorithm in the first stage. Then the poor clusters with similar features were deleted while the appropriate clusters were further selected based on cluster pruning. At the third stage, an item prediction was made by performing a weighted average of deviations from the neighbor’s mean. Such an approach was likely to trade-off on increasing the diversity of recommendations while maintaining the accuracy of recommendations. Zhou et al. [40] represented Data-Providing (DP) service in terms of vectors by considering the composite relation between input, output, and semantic relations between them. The vectors were clustered using a refined fuzzy C-means algorithm. Through merging similar services into a same cluster, the capability of service search engine was improved significantly, especially in large Internet-based service repositories. However, in this approach, it is assumed that domain ontology exists for facilitating semantic interoperability. Besides, this approach is not suitable for some services which are lack of parameters. Pham et al. [41] proposed to use network clustering technique on social network of users to identify their neighborhood, and then use the traditional CF algorithms to generate the recommendations. This work depends on social relationships between users. Simon et al. [42] used a high-dimensional parameter-free, divisive hierarchical clustering algorithm that requires only implicit feedback on past user purchases to discover the relationships within the users. Based on the clustering results, products of high interest were recommended to the users. However, implicit feedback does not always provide sure information about the user’s preference.

## III. PROPOSED ALGORITHM

### A. *Design Considerations:*

- Cluster Services.
- Compute Description Similarity and Functionality Similarity
- Compute Characteristic Similarity
- Cluster Services

### B. *Description of the Proposed Algorithm:*

Aim of the proposed algorithm is to maximize the network life by minimizing the total transmission energy using energy efficient routes to transmit the packet. The proposed algorithm is consists of three main steps.

Step 1.1: Stem Words:

Different developers may use different-form words to describe similar services. Using these words directly may influence the measurement of description similarity. Therefore, description words should be uniformed before further usage. In fact, morphological similar words are clubbed together under the assumption that they are also semantically similar. For example, „map‟, „maps‟, and „mapping‟ are forms of the equivalent lexeme, with „map‟ as the morphological root form. To transform variant word forms to their common root called stem, various kinds of stemming algorithms, such as Lovins stemmer, Dawson Stemmer, Paice/Husk Stemmer, and Porter Stemmer, have been proposed [13]. Among them, Porter Stemmer (http://tartarus.org/martin/PorterStemmer/) is one of the most widely used stemming algorithms. It applies cascaded rewrite rules that can be run very quickly and do not require the use of a lexicon [14]. In ClubCF approach, the words in $D_t$ are gotten from service Bigtable where row key = “$s_t$” and column family = “$Description$”. The words in $D_j$ are gotten from service Bigtable where row key = “$s_j$” and column family = “$Description$”. Then these words are stemmed by Porter Stemmer and put into $D_{t'}$ and $D_{j'}$, respectively.

Step 1.2:

Compute Description Similarity and Functionality SimilarityDescription similarity and functionality similarity are both computed by Jaccard similarity coefficient (JSC) which is a statistical measure of similarity between samples sets [15].

For two sets, JSC is defined as the cardinality of their intersection divided by the cardinality of their union. Concretely, description similarity between $s_t$ and $s_j$ is computed by formula (1): $D\_sim\ s_t, = D_{t'}\bigcap D_{j'}\ D_{t'}\bigcup D_{j'}$ (1) It can be inferred from this formula that the larger $D_{t'}\bigcap D_{j'}$ is, the more similar the two services are. Dividing by $D_{t'}\bigcup D_{j'}$ is the scaling factor which ensures that description similarity is between 0 and 1. The functionalities in $F_t$ are gotten from service Bigtable where row key = "$s_t$" and column family = "$Functionality$". The functionalities in $F_j$ are gotten from service Bigtable where row key = "$s_j$" and column family = "$Functionality$". Then, functionality similarity between $s_t$ and $s_j$ is computed using JSC as follow: $F\_sim\ s_t, = F_t\bigcap F_j\ F_t\bigcup F_j$ (2)

## IV. Pseudo code

Step 1: Search for the pair in the similarity matrix with the maximum similarity and merge them.
Step 2: Create a new similarity matrix where similarities between clusters are calculated by their average value.
Step 3: Save the similarities and cluster partitions for later visualization.
Step 4: Proceed with 1 until the matrix is of size $K$, which means that only $K$ clusters remains.

Let $K$=3 as the termination condition of Algorithm 1, the reduction steps are illustrated in TABLE VI~TABLE IX. As for reduction Step 1 as shown in TABLE VI, since the maximum similarity in the similarity matrix is $dC2,5$, $C2$ and $C5$ are merged into $C2,C5$ . And the similarity between $C2,5$ and other clusters is calculated by their average value. For example, $dC2,5$ ,$C3$= $dC2,C3$+$dC5,C3$ 2 = 0+0.063 2 ≅0.032. As for reduction Step 2 as shown in TABLE VII, since the maximum similarity in the similarity matrix is $dC3,4$, $C3$ and $C4$ are merged into $C3,C4$ . And the similarity between $C3,4$ and other clusters is calculated by their average value. For example, $dC2,5$ , $C3,C4$ = $dC$

## V. Simulation Results

To evaluate the accuracy of ClubCF, Mean Absolute Error (MAE), which is a measure of the deviation of recommendations from their true user-specified ratings, is used in this paper. As Herlocker et al. [36] proposed, MAE is computed as follow: MAE= $ra,−Pua,stni=1n$(8)

In this formula, $n$ is the number of rating-prediction pairs, $ra$, is the rating that an active user $ua$ gives to a mash up service $st$, ($ua$,$t$) denotes the predicted rating of $st$ for $ua$. In fact, ClubCF is a revised version of traditional item-based CF approach for adapting to big data environment. Therefore, to verify its accuracy, we compare the MAE of ClubCF with a traditional item-based CF approach (IbCF) described in [26]. For each test mash up service in each fold, its predicted rating is calculated based on IbCF and ClubCF approach separately. The mashup services published on Programmable Web focus on six categories which labeled with keywords: "photo", "google", "flash", "mapping", "enterprise", and "sms". Therefore, without loss of generality in our experiment, the value of $K$, which is the third input parameter of Algorithm 1, is set to 3, 4, 5, and 6, respectively. Furthermore, rating similarity threshold $\gamma$ is set to 0.1, 0.2, 0.3 and 0.4. Under these parameter conditions, the predicted ratings of test services are calculated by ClubCF and IbCF. Then the average MAEs of ClubCF and IbCF can be computed using formula (8). The comparison results are shown in Fig. 2 (a), (b), (c) and (d), respectively.

DOI: 10.15680/ijircce.2015.0305154

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

Vol. 3, Issue 5, May 2015

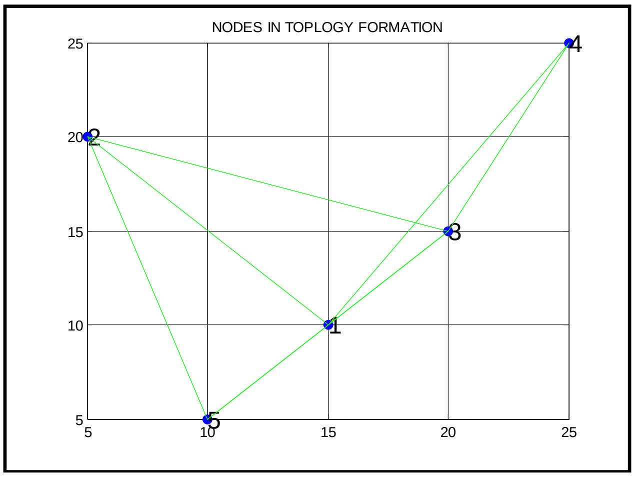

Fig.1.Ad Hoc Network of 5 Nodes

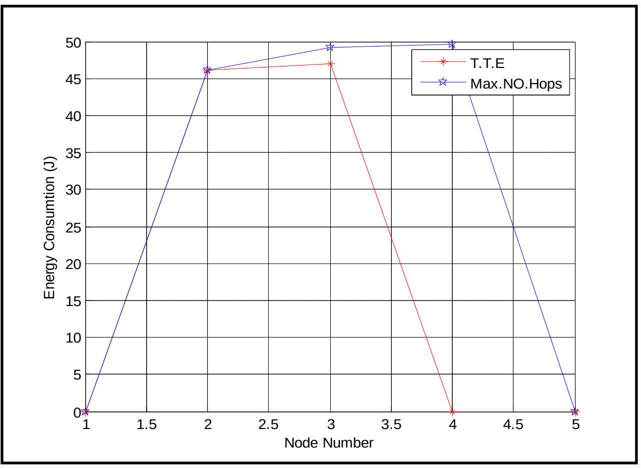

Fig. 2. Energy Consumption by Each Node

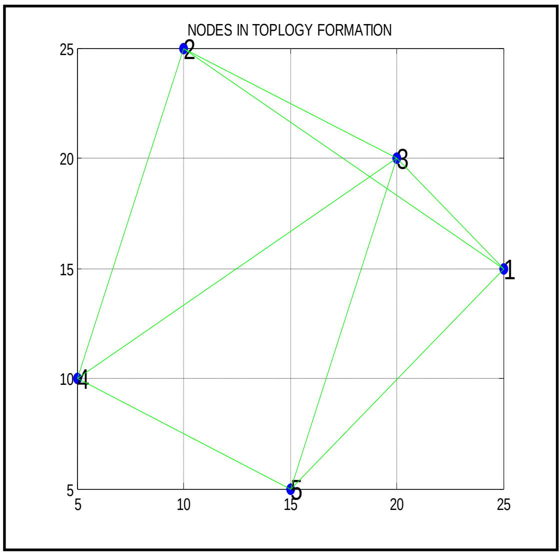

Fig. 3.Ad Hoc Network of 5 Nodes

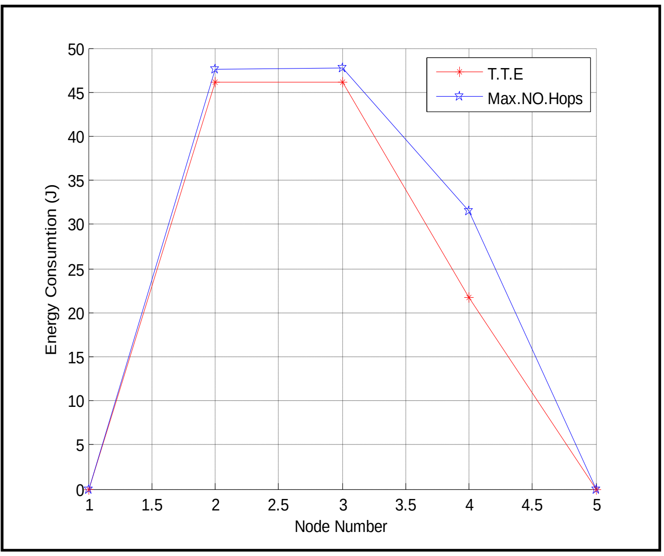

Fig 4. Energy Consumption by Each Node

## VI. Conclusion and Future Work

The Similarity of the services which interact with big data Hadoop,Bigtable are handled. And we implemented the web services to retrieve the values and not storing in any of the system. And also we are providing Search text for the services to increase the performance.Semantic-similar services needs to be clustered together and more analysis to be done to retrieve the algorithm which will increase the coverage of recommendations

## References

1. 1.M. A. Beyer and D. Laney, "The importance of "big data": A definition," Gartner, Tech. Rep., 2012.
2. X. Wu, X. Zhu, G. Q. Wu, et al., "Data mining with big data," IEEE Trans. on Knowledge and Data Engineering, vol. 26, no. 1, pp. 97-107, January 2014.
3. A. Rajaraman and J. D. Ullman, "Mining of massive datasets," Cambridge University Press, 2012.
4. Z. Zheng, J. Zhu, M. R. Lyu. "Service-generated Big Data and Big Data-as-a-Service: An Overview," in Proc. IEEE BigData, pp. 403-410, October 2013.
5. A. Bellogín, I. Cantador, F. Díez, et al., "An empirical comparison of social, collaborative filtering, and hybrid recommenders," ACM Trans. on Intelligent Systems and Technology, vol. 4, no. 1, pp. 1-37, January 2013.

W. Zeng, M. S. Shang, Q. M. Zhang, et al., “Can Dissimilar Users Contribute to Accuracy and Diversity of Personalized Recommendation?,” International Journal of Modern Physics C, vol. 21, no. 10, pp. 1217-1227, June 2010.

6. T. C. Havens, J. C. Bezdek, C. Leckie, L. O. Hall, and M. Palaniswami, “Fuzzy c-Means Algorithms for Very Large Data,” *IEEE Trans. on Fuzzy Systems*, vol. 20, no. 6, pp. 1130-1146, December 2012.
7. Z. Liu, P. Li, Y. Zheng, et al., “Clustering to find exemplar terms for keyphrase extraction,” in *Proc. 2009 Conf. on Empirical Methods in Natural Language Processing*, pp. 257-266, May 2009.
8. X. Liu, G. Huang, and H. Mei, “Discovering homogeneous web service community in the user-centric web environment,” *IEEE Trans. on Services Computing*, vol. 2, no. 2, pp. 167-181, April-June 2009.
9. H. H. Li, X. Y. Du, and X. Tian, “A review-based reputation evaluation approach for Web services,” *Journal of Computer science and technology*,

## BIOGRAPHY

**P S Manikandan**is a M.E Final year student in the Computer science department, Rrase College of engineering, Chennai. He received Master of Bachelor of Engineering degree in 2006 from Annauniversity,Chennai,India. His research interests are Datawarehousing,BusinessAnalytics,Big data Hadoop, etc.

DOI: 10.15680/ijircce.2015.0305154