iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** IX **Month of publication:** September 2015

DOI:

Volume 3 Issue IX, September 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Cloud Based Mobile Video Recommendation System with User Behaviour

Ms. Latika R.Gaddam[1], Mrs. Pratibha S. Yalagi[2]
[1]Computer science and engineering, Solapur University

***Abstract*— Nowadays, Countless online video services are available Users usually waste lot of time to obtain their interested videos in browsing and watching those videos on mobile. The existing result shows the poor service quality of video streaming over mobile network such long buffering and interrupt happen in the streaming video. The proposed system is a video recommendation system which can speed up the recommendation process and reduce network overhead. The mobile video streamed by using the cloud. For video-sharing mobile application Mobile properties are collected for context aware recommendation. The Cloud based recommendation system is created with Mahout Machine learning library. Core algorithms of Mahout are used for classification, cluster and collaborative Moreover filtering. We use Mahout's user and item based recommendation and user behaviour. The System collects user, item, rating, genre and viewing time for creating clusters of user profiles to generate implicit preferences video recommendations. Mobile properties such as location, time and network types are used to generate recommendation rules. User context and profile clusters are used for finding video recommendation from the massive amount of video collection on cloud. The proposed system can recommend desired services with high recall, high precision and low response delay.**
***Keywords*— Mahout Machine learning library, Cloud storage, video recommendation, user behaviour analysis, Collaborative filtering.**

I. INTRODUCTION

Due to the explosive growth of multimedia content, e-commerce and online environment has made it quite difficult for people for search information that is pertinent to their interest and needs. One solution to this problem is a **recommendation system** which is a useful tool which offers an automated mechanism to seek out relevant as well as new information. Now a day people are fully depends on their social network on the internet. All the information regarding products, music, video, films, television programs, web pages, news, restaurant and blog available on the internet. Therefore, different recommendation systems have been proposed. Most of the systems deploy a large number of context collectors at terminals and access networks. However, Context collecting and exchanging results heavy network overhead and huge computation requirement. The solution to network overhead is **Cloud service.** The internet provides several services in the day to day activities of the technology world. A Cloud service is one among such services that plays major role in the networking environment. As we know that cloud is upcoming technology in the world. It provides several types of services for the user, one of them is Storage as a Service. Our project is related to the Saas services of the cloud. In recent year the video sharing websites (e.g. YouTube) has achieved great success. The video content may be similar or quite different in billions of video sharing websites. Users are always facing a problem of finding their useful videos in less time. For mobile users the situation is worse because of low bandwidth and screen limit. How to help mobile users obtain their desired content lists from billions of web pages in a short time is a very challenging issue. Some websites provide searching is based on the keywords. In that case Mobile users do not have any keyword when they process the search. It's very difficult to find there interested video. So in big data environment improve its scalability and efficiency with user preferences and user-based Collaborative Filtering algorithm to provide recommendation.

In this paper, we proposed mahout technology for implementing a recommendation system with easier and faster. **Mahout** is a scalable machine-learning library which is implemented on top of Apache Hadoop and using the MapReduce paradigm. It does not restrict contributions to Hadoop based implementations [11]. The Mahout's core algorithms are used classification, clustering and collaborative filtering. In addition Mahout is self managing and can easily handle hardware failures as well as scaling up or down its deployment without any change to the codebase.

In the proposed system, the video sharing application is used to find the context information from a user's smartphone. Initially after logging into the system we provide user based recommendation. Mahout recommender requires interactions between users and items. Mostly user after watching complete video cannot give rating we use the view count to find similarity betwwn them. The userid, Itemid genre, ratings and view counts are used to find similarity between the two users. The user and item based recommendation of mahout with the collaborating filtering algorithm. User contexts and the profiles will be delivered collectors and then user context clusters according to the clustering rules and mahout cluster. Mahout creates a user profile cluster by using clustering rules. Then Mahout Recommender function is use data model, user similarity and user neighboring for recommendation rules. The context used to create restrictions rules for playing videos in specific context.

Recommendation rules are reordered to improve scalability and real-time recommendation. Existing recommendation systems always recommend a ranked list for users after training from some given data. However, if the content changes or a new keyword appears, a fixed list is always provided.

II. RELATED WORK AND SYSTEM ARCHITECTURE

A. Recommendation System

There are various specific domains for Recommendation systems. For Example Google News provides personalized news recommendation services for a substantial amount of online readers. The Amazon has achieved great success in recent years. Amazon uses the recommender system to help users find their desired products. YouTube uses user watching history to predict and recommend videos for users.

1) *Content-Based Recommendation System:* In a content-based recommender system, keywords or attributes are used to describe items. A user profile is built with these attributes Items are ranked by how closely they match the user attribute profile, and the best matches are recommended. [14] The systems make recommendation based on the similarities of content titles, tags, or descriptions. Some systems find user-interested items based on user's individual reading history in term of content [12].
2) *Collaborative Filtering Recommendation System:* The systems make recommendation based on sufficient user transaction histories and content popularity. In the systems, individual user's interests are predicted by a group of similar users. To obtain the content rating and users' similarity, statistics and feedback methods are used [8], [9]. Collaborative filtering (CF) is a technique used by some recommender systems. Most of the e-commerce websites use CF to help users find their interested goods [4].
3) *Context-Aware Recommendation System:* Context-aware recommendation systems user context sensed on Smartphone and long-time user profile to assist the user in selecting better services and videos dynamically. However, it is also important to the capture contextual information into the recommendation process in order to get better recommend items to users in certain circumstances. Context is a difficult concept to capture and such as fuzzy ontologies and semantic reasoning are used to augment and enrich the description of context [1], [7],[8].
4) *Graph-Based Recommendation System:* Graph-based recommendation system is to calculate the correlation between recommendation objects and the node selection problem on a graph. Incorporating conversion of the content and contextual information, links on video pages are converted to undirected weighted graph. Users' Co tag behaviors and the graph described with friendships in social network and random walk restarts is applied to the social graph to recommend items. [2], [3].

As a stateless programming model, MapReduce cannot directly express Collaborative Filtering so it is performed on Hadoop,[6] KASR is implemented on Hadoop platform. To improve its efficiency and scalability in the big data environment, a widely-adopted distributed computing platform using the MapReduce parallel processing system [9]. With the huge increase of user numbers, user contexts, video contents and the user profiles, recommendation systems require more and more computation capacity. To resolve the huge computation requirements, a CF algorithms Recommendation system based on user behavior is proposed

B. Cloud Based Recommendation System

The proposed system is a video sharing application includes two parts: recommendation training and real time recommendation. The Mahout Recommendation system uses COLLABORATIVE FILTERING for collecting clusters recommendation rules according to the user profile information. Real time recommendation components will extend requests to recommendation rules and will return the recommendation lists with optimized rules of Mahout. The framework of the proposed system is illustrated in fig 1.

Volume 3 Issue IX, September 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

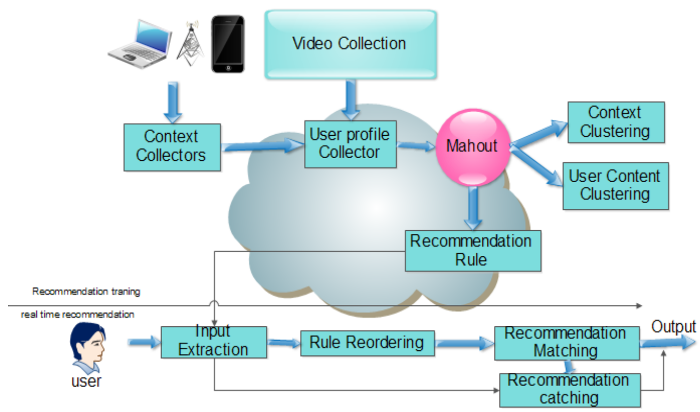


Fig 1: Conceptual overview of proposed system

The Video sharing application is used to finding their context information from user Smartphone. User Context and access preference are collected to context collector based on time, location, network type and screen resolution etc
User based recommendation are provided when initially logged into the system. The User Profile collector is created with watching history, ratings, genre and viewing time. Then Mahout Recommender function is use data model, similarity and neighboring for recommendation rules.
While user content for the search item as Item based recommendations are provided by user profile cluster by using clustering rules of mahout.
Recommended videos are collected from Cloud and shown to end user.
User will select video to play. Rules are extracted based contextual information for restricting video for that context.

In the Proposed system Cloud of Eucalyptus is free and open source computer software for building Amazon web Services (AWS).The dataset is created with Userid, Itemid, Rating and view count in mysql database. Apache Mahout is a project of the apache software foundation to produce free implementations of distributed environments.

The four major components in our framework are described as follows [1].

1) *Data Model:* Dataset is a collection of data. Dataset is user, item with specific to rating, genre, view count. In the recommender training dataset created for calculating similarities of user. For testing data send for particular user and item dataset.
2) *User And Item Based Recommender:* The idea behind this system is that when we want to compute recommendations for particular users, we look for other users with a similar taste and pick the recommendations from their items. For finding similar users are clustered by comparing their interactions. There are several methods for doing this. The collaborating filtering algorithm recommendation based on abundant user transaction histories and content popularity. The Mahout recommender is expecting the interaction between user and item as input. The User based recommender Function use the userid, itemid and rating to provide recommendations The user similarly and threshold of user neighborhood is used to create Recommendation rules with user context clusters and profile clusters
3) *User Behavior Collection:* Mahout's recommenders use an interface called *Data Model* to handle interaction data. User interest is vary according to time, location, emotions etc. User always watches the videos with specific types in particular context which indicates that user has unique access preferences. User Context is collected from android Smartphone with context of Network type, location, device screen resolution etc. User profile is created with the help of mahout recommender and data model of cloud storage. And also watching history, rating and viewing time is used to create the user profiles..
4) *Real-Time Recommendation:* The real-time component accepts the user's requests and returns the recommendation lists to the user. For new user the recommendation System creates cold start problem. To overcome cold start problem we provide fixed list recommendation. We have proposed a User and Item based recommender system for videos based on the Mahout platform. Distinguishing with other recommender systems, we have stored recommendation rules instead of recommending lists. Mahout will provide more efficient recommendation then manually preparing recommendation rules with different nodes by Hadoop

and MapReduce technology.

III.USER BEHAVIOR CONSTRUCTION

A. Mahout Recommendation Systems

1) Collaborative Filtering: In collaborative filtering, items are recommended according to the people having similar tastes, interests and according to their browsing history. These systems identify users who are having same preferences with the present user and propose which the user are interested in but has not yet seen. Generally, CF uses two main approaches, namely, user-based CF and item-based CF.

They are shown in table1.

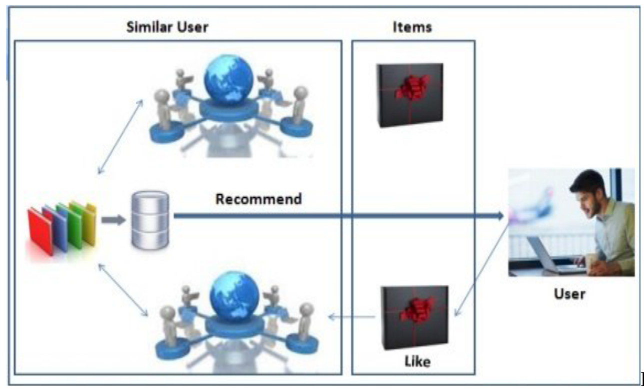
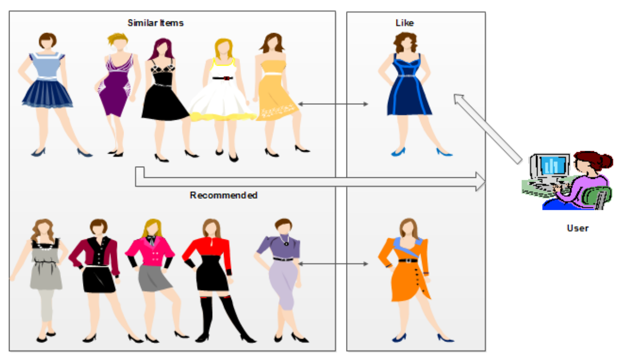
a) User-based Recommendation: In this category, recommendation is based on the highly correlated user by using the rating Preferences of other users to find highly correlated users. This category is described in fig2.	 FIG2
b) Item-based Recommendation: In this category, recommendation is based on the highly correlated item by using the rating preferences of other users to find highly correlated users This category is described in fig3.	 FIG3

Table1: User and Item Based Recommendation:

2) User And Item Based Recommendation: The Data Model is nothing but dataset which is stored on sever. The dataset is collection Bollyhood Movies. Mahout's recommenders use an interface called *DataModel* to handle interaction data. It fastByIdSet, map created for user, item and rating of the movie.

DataModel model = new FileDataModel(Map);

we want to compute recommendations for a particular users, we take consideration for other users with a similar taste and pick the recommendations from their items. There are several methods for doing For Finding Similarity for user we created a function with rating genre and viewing time.

UserSimilarity similarity = new cosineSimilarity(model);

we'll use all that have a similarity greater than *0.1*

UserNeighborhood neighborhood = new NearestUserNeighborhood(0.1, similarity, model);

Now we have all the pieces to create our recommender:

UserBasedRecommender recommender = new GenericUserBasedRecommender(model, neighborhood, similarity);

However, Item-based recommendation is derived from how similar *items* are to *items*, instead of *users* to *users*. In Mahout, this means they are based on an Item Similarity implementation instead of User Similarity [11].
It deploys a GenericItemBasedRecommender rather than GenericUserBasedRecommender, Different and simpler set of dependencies are required. CosineMeasureSimilarity still works here because it also implements the ItemSimilarity interface, which is entirely similar to the UserSimilarity interface. It compares series of preferences expressed by many users, for one item, rather than by one user for many items. GenericItemBasedRecommender is simpler. It only needs a DataModel and Item- Similarity—there's no ItemNeighborhood. You might wonder at the apparent asymmetry.

3) *Similarity:* We are concerned with three kinds of video properties Genre, Rating, and View Group. Based on the connection relationship, we construct a weighted graph like Fig. 4. The graph is composed of three kinds of sub graph. The Genre sub graph is constructed from user adds in this group when a rating is given for same genre, the two users are linked together. If two users take each other as same genre and same rating there are two edges between users. Whenever user view the video does not give a rating for videos then the implicit viewing count is created. The sub graph with green link is plotted based on user interested groups. If a user joins an interested group, he is connected with the group. It is complex to recommend items based on the graph directly.

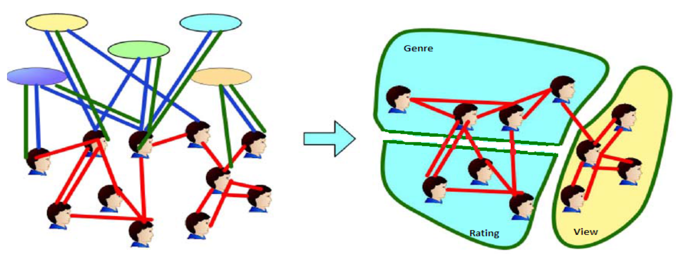


Fig 4 User Profile similarity

Sim(*A*,*B*) = SimGenre(*A*,*B*) * SimRating (*A*,*B*)* Simview(*A*,*B*);

where *A* and *B* are two user profiles. We have to calculate similarity of same genre video seen by 2 users A,B and we have to calculate similarity of same rating video given by 2 users A,B and we have to calculate similarity of same view count of video by 2 users A,B.

4) *Cosine Similarity:* Similarity in terms of Genres, Rating and View count are calculated by using cosine similarity function. Using the formula given below we can calculated the similarity between any two Users d1 and d2 where d[0] to d[n] are number of item related to that user.

Cosine Similarity (d1, d2) = Dot product(d1, d2) / ||d1|| * ||d2||

Dot product (d1,d2) = d1[0] * d2[0] + d1[1] * … * d1[n] * d2[n]

||d1|| = square root(d1[0]2 + d1[1]2 + ... + d1[n]2)

||d2|| = square root(d2[0]2 + d2[1]2 + ... + d2[n]2)

We need to understand this is that the cosine value is always between –1 and . The cosine of a small angle is near 1, and large angle near 180 degrees is close to –1. This is good, because small angles should map to high similarity, near 1, and large angles should map to near –1. When two users are similar, they'll have similar ratings, and so will be relatively close in space. They will be in roughly the same direction from the origin. The angle formed between these two lines will be relatively small. In contrast, when the two users are dissimilar, the distant points will be, and likely in different directions from the origin, forming a wide angle.

B. *User Behavior Calculation*

It is verified that user contexts are essential to provide users right services in ubiquitous networks. The contextual information such as time and location is more likely to change, user contexts have to be reported in time. initial users or contract users collect basic

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

user contexts, such as network type, time , and location . The contexts are reported to the collectors at the server side. When a user requests videos, the plug-ins will collect user contexts and it will check in restriction rules.

1) *Per Fetch Process:* If the Network type wifi then all videos are directly Streaming is possible from cloud If Network type the mobile device, then there are different range for different networks it varies from 14kbps to 23 mbps. For eg:
TelephonyManager.NETWORK_TYPE_GPRS:
Return false; // ~ 100 kbps

If the user selected the video to play in specific area which is restricted for that location then we cannot take video from cloud and send message. However, some videos we cannot play in the night time like after12 pm like that. If the user selected the video to play it is in restricted time period of that video then we cannot take video from the cloud and send message.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

Our recommender system includes two parts: one part is mobile application, which is implemented on the Android platform, collects user contexts, and calculating according to the context restriction rule. The core part is the recommender at the server side, which is implemented on the Mahout platform, includes user group partition, user profile clustering, recommendation rule generation, and real-time recommending. We deploy the recommender system on a small-sized cloud platform. To evaluate the performance of the system, training data and test data should be gathered.

A. Precision And Recall:

Precision is also known called positive predictive value and it is the fraction of retrieved instances that are relevant, while **recall** is also known as sensitivity and it is the fraction of relevant instances that are retrieved. After gathering the data, we classify the user profiles into two sets. A set of profiles is used as training data and denoted as T, and the other set of profiles is used as verified data and represented as V. Our recommender system recommends a list with no more than 20 items, which is denoted as R. Referring to other works, we define two metrics. Precision(R, V) and Recall(R, V) as (1) to evaluate the performance of our work

$$\text{Precision}(R,V) = \frac{\text{size}(R \bigcap V)}{\text{size}(R)}$$
$$\text{Recall}(R,V) = \frac{\text{size}(R \bigcap V)}{\text{size}(V)}. \quad (1)$$

It shows the comparison results of different profile cluster numbers. From Fig. 9, we observe that, as the number of user profile clusters k increases from 1 to 32, precision and recall improve consistently. Further increase of k from 32 results in less improvement of recommendation quality, but user profiles can be clustered into more small groups. The small groups benefit from real-time recommendation. The figure also shows that a very large cluster number leads to unimaginable quality degradation. For example, a recommendation quality of k = 300 is even lower than that of k = 1. The reason is that a few profiles in the small group do not cover user interest [13].

The proposed system can recommend desired services with high recall, high precision and low response delay.

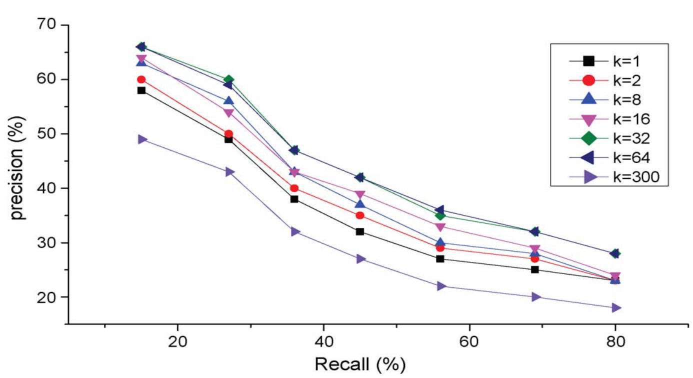


Fig5. Comparison results of different profile cluster numbers.

Volume 3 Issue IX, September 2015
ISSN: 2321-9653

B. Clustering Latency

Comparing the latency of three clustering algorithms with cloud and without cloud, we plot clustering latency with different cluster sizes in Fig. 12. From the figure, we can conclude that the cloud platform does not reduce clustering latency when the cluster number is small, but it helps in improving the performance of the system with the increment of the cluster number.

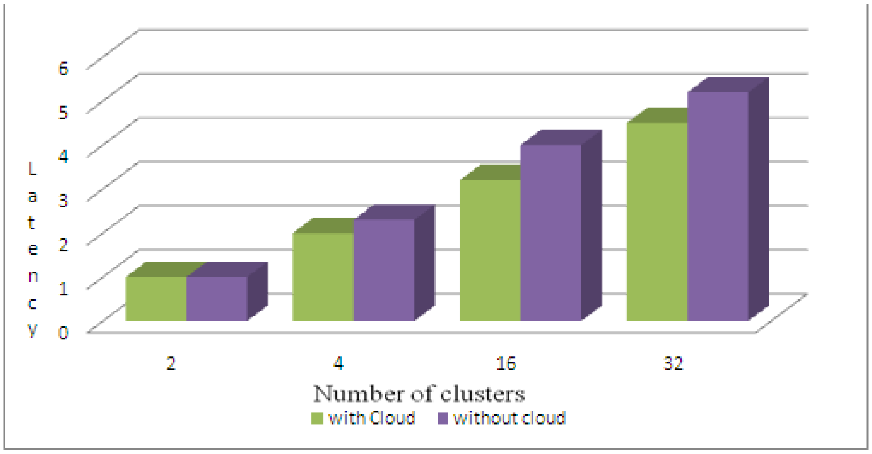


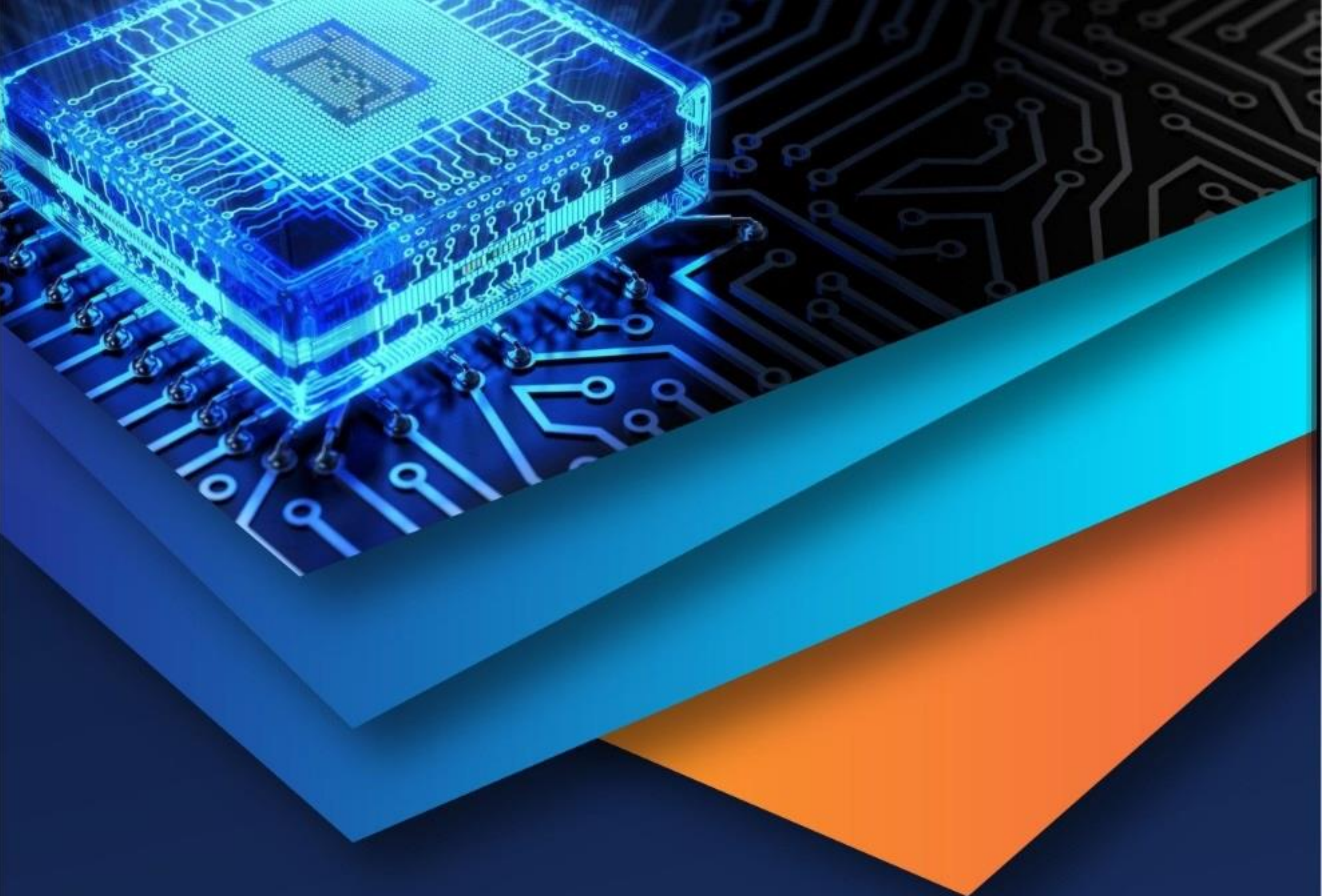
Fig6. Clustering latency with different cluster sizes with cloud and without cloud

V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a cloud based recommender system for videos based on the Mahout platform. Distinguishing with other recommender systems, we have stored recommendation rules instead of recommending lists. Cloud storage is used which can reduce network overhead and speed up the recommendation process to obtain useful recommendation in less time. Mahout will provide more efficient recommendation then manually preparing recommendation rules for user and item based recommendation. Cloud Platform does not reduce clustering latency when cluster number is small but it helps in improving the performance of the system with increment of the cluster number. Another important point that should be studied is designing a distributed recommendation cache to improve recommending mahout process. The cache can also reduce computation pressures caused by the amount of concurrent rule reordering and executions. The further implementation can be done in hadoop and mapreduce to recommendation process.

REFERENCES

[1] Latika Gaddam, Pratibha Yalagi "Mobile Video Recommendation System on Cloud with User behavior".
[2] A. C.M.Costa, R. S. SGuizzardi,G.Guizzari, and J. G. P. Filho " COReS:Context-aware, ontology-based recommender system for service recommendation,"
[3] X.Wu, Y. Zhang, J. Guo, and J. Li, "Web video recommendation and long tail discovering," in Proc. IEEE ICME, 2008, pp. 369–372.
[4] G. Go, J. Yang, H. Park, and S. Han, "Using onlinemedia sharing behavior as implicit feedback for collaborative filtering,"
[5] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly, "Video suggestion and discovery for YouTube: Taking random walks through the view graph,"
[6] Z.-D. Zhao and M.-S. Shang "User-based collaborative-filtering ecommendation algorithms on Hadoop
[7] Moon-Hee Park, Jin-Hyuk Hong, and Sung-Bae Cho "Location-Based Recommendation System Using Bayesian User's Preference Model in Mobile Devices"
[8] A. C. G. C. A. Cimino, B. Lazzerini, and F. Marcelloni, "Situation-aware mobile service recommendation with fuzzy logic and semantic web,"
[9] Shunmei Meng, Wanchun Dou, Xuyun Zhang, Jinjun Chen, "KASR: A Keyword-Aware Service Recommendation Method on MapReduce for Big Data Applications".
[10] Yijun Mo, Jianwen Chen, Xia Xie, Changqing Luo, and Laurence Tianruo Yang," Cloud-Based Mobile Multimedia Recommendation System With User Behavior Information".
[11] Sean Owen, Robin Anil, Ted Dunning and Ellen Friedman "Mahout in Action" book.
[12] Ramya. S, Mrs Anitha. M.L "Recommendation System for Videos in Cloud Based on User Behavior"
[13] Yijun Mo, Jianwen Chen, Senior Member, IEEE, Xia Xie, Changqing Luo, and Laurence Tianruo Yang, Member, IEEE "Cloud-Based Mobile Multimedia Recommendation System With User Behavior Information".
[14] M. J. Pazzani and D. Billsus,"Content-based recommendation systems," in The Adaptive Web. Berlin,Germany: Springer-Verlag, 2007, pp. 325–341.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY