I.J. Information Technology and Computer Science, **2017, 10, 59-67**
Published Online October 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2017.10.06

An Efficient Framework for Creating Twitter Mart on a Hybrid Cloud

[1]Imran Khan
[1]FTK-CIT, Jamia Millia Islamia,New Delhi-25, India
E-mail: imrankhan.ee2531@gmail.com

[1]S. Kazim Naqvi, [2]Mansaf Alam, [3]Mohammad Najmud Doja and [4]S. Nasir Aziz Rizvi
[2] Department of Computer Science, Jamia Millia Islamia, New Delhi-25, India
[3]Department of Computer Engineering,Faculty of Engineering & Technology, Jamia Millia Islamia, New Delhi-25, India
[4]Dept. of Mathematics, Jamia Millia Islamia, New Delhi-25, India
E-mail: sknaqvi@jmi.ac.in, malam2@jmi.ac.in, mdoja@jmi.ac.in, dr.snarizvi@gmail.com

Received: 19 June 2017; Accepted: 13 July 2017; Published: 08 October 2017

***Abstract*—**The contemporary era of technological quest is buzzing with two words - Big Data and Cloud Computing. Digital data is growing rapidly from Gigabytes (GBs), terabytes (TBs) to Petabytes (PBs), and thereby burgeoning data management challenges. Social networking sites like Twitter, Facebook, Google+ etc generate huge data chunks on daily basis. Among them, twitter masks as the largest source of publicly available mammoth data chunks intended for various objectives of research and development. In order to further research in this fast emerging area of managing Big Data, we propose a novel framework for doing analysis on Big Data and show its implementation by creating a '*Twitter Mart*' which is a compilation of subject specific tweets that address some of the challenges for industries engaged in analyzing subject specific data. In this paper, we adduce algorithms and an holistic model that aids in effective stockpiling and retrieving data in an efficient manner.

***Index Terms*—**Cloud Computing, Big Data, Hadoop, Twitter.

I. Introduction

The *Big Data and Cloud Computing* have been creating waves everywhere. With the huge volumes of digital data flooding in GBs, TBs to PBs, the compliance of effective data management is still a hard nut to crack. Big giants to the likes of Google, Yahoo and Microsoft, etc. have huge repositories of data generated through processes and customized processes to analyze and provide sententious insight into that data. Google has diverse projects running that deals with innovative way of manoeuvring Big Data like Google MapReduce, Google Bigtable, Google Borg, Google Chubby, Google Dremel etc. [1]. Big Data custodian solutions catering to these projects need are Apache Hadoop, Apache Hbase, Apache Zookeeper etc.

Numerous extractions from over the Internet like server logs, sensor data, mobile app data, e-commerce site data and data from social networking sites like Facebook, Twitter, Google+, Instagtram etc face congested traffic on a periodic basis and spawn data periodically in such gigantic volumes that fulfill all the 3V typical of Big Data i.e. volume, velocity and variety[2]. Online social network data has exhibited exponential growth in various sectors[3]. Present count of more than 1.8 million active users of Facebook[4] alone is an indicator of the size of the problem. Data from such sites serve multiple purposes in various organizations. For instance - to gauge the popularity of any product, to fathom global news in real time, to expand the knowledge base of a scientific community etc. The data generated by these social networking sites cannot be hoarded or shelved at a single place. Also the data cannot be managed by the traditional database solutions and they exhibit variety of limitations in meeting current Big Data analytics and management [36]. Hence distributed computing is the only viable solution for them. Managing data of these social networking sites is a ponderous issue and it includes stretching the data in such a way that it will be easily stored, highly available and scalable.

Cloud computing has arrived as a new paradigm for stockpiling data in a cost effective, scalable and location independent manner. This is the emerging technology and is highly significant in both business and academic environments [37]. Cloud computing presents umpteen services and users can avail them on a *just-pay-as-you-use* basis. Many organizations are using cloud ERPs (Enterprise Resource Planning) that are supported by SAAS (Software-as-a-Service) and these ERPs can be wangled by the user browser over the Internet without actually restricting them to users site(s) [5]. However data outsourcing to a Cloud service provider possesses new challenges of data portability, reliability and security etc. Although Public Clouds [6] are becoming more

prominent due to services offered freely or with the nominal *pay-per-use* charges. In addition to this, some may use Private Clouds or an amalgamation of Private and Public Cloud commonly known as Hybrid Cloud.

Twitter is the largest source of candid publicly available mammoth data chunks which can be of significant use in various sectors of research and development. Devising a system that stores such data on a Hybrid Cloud is a mammoth challenge for Cloud service providers. As the data in Hybrid Cloud is largely unrecompensed or offered at minimal cost to the users, hence with availability and scalability of the solution the cost of lodging data is a paramount issue that need to be taken care of by any cloud service provider or the client.

In this paper, we propose algorithms and an overall model that aids in effectively stockpiling and retrieving data in the most conducive manner known as twitter mart. The framework essentially rattles out data from twitter and store them on a platform of subject specific clusters. The framework uses Hybrid Cloud for data storage in a cost effective manner.

Section 2 of the paper focuses on Literature Survey, Section 3 explains the Problem Statement, Section 4 briefly explains cost factors, the cloud storage model and the analysis is given in Section 5. Cost reckoning is done in section 6 and Section 7 shows the experiments and inferences. Discussion and Conclusion are presented in in Section 8 and 9 respectively. References are mentioned in the last section.

II. Related Work

With decreasing cost of storage, digital data is increasing in colossal amounts. The ongoing research work in cloud infrastructure will further improve efficiency in stockpile data competently in the cloud so that the read/write operations attain further optimality and the data availability is clinched. Based on contrastive scenarios assorted solutions have been proposed in anterior studies. In prevailing literature in order to ensure data availability the main focus was in drafting few more nimble policies for data stockpiling and data counterfeiting. As in [7], three schemes have been proposed for data storage in DOSNs. The three schemes were: the cloud based scheme, the desktop based scheme and the hybrid scheme which mainly amalgamates the cloud based and desktop based scheme. In the first scheme, data will be stored in cloud servers. In the second scheme, either data replicas are encrypted before stockpiling them towards virtual hosts or the user take the practicality of trust that has been embedded in the social network in order to store its duped data on confirmation and established friends. But the approaches are not competent enough as the complexity and the overhead is altitudinous in monitoring the encryption keys and trust. Although some other systems have used varied strategies for proposing different frameworks for different case studies. As [8] has proposed a cloud platform for fully coherent car system. A similar but more amplified architecture has been proposed in [9] and is known as *Cloudthings*.

Although cloud computing provides a wide range of services and advantages to the clients such as gratuitous services, easy to use etc., still clients resist to provide their confidential data to the cloud service provider[10]. It is because of lack of trust on the cloud service provider and losing the direct control on their confidential data. Cloud service providers uses methods like firewall, encryption, virtualization etc., to provide security on the data stored but still the methods are insufficient. Cloud service providers need more unassailable and robust methods to ensure the privacy and confidentiality of the data stored [11,12]. The most conventional issues that have been involved in cloud security have been discussed in [10]. One solution for cloud security has been proposed by [13] SecCloud. The solution proposes protocols for ensuring protection of the data in the cloud. In [14] a simple privacy preserving identity management has been proposed for cloud environment (SPICE). It provides a user centric authentication by clinching unidentified authentication access control. [15] proposed an identity management framework to disburse access control across multiple cloud service providers. One of the security deportment with the cloud is breach of trust, so service level agreements need to be countersigned from the client and service provider. In [16] authors proposed a scheme that will ensure and react on violation of service level agreements.

As the online social network (OSN) data are very prominent and a user stores his personal details onto the social networks the data on these networks need to be armed. Generally the frameworks that have been used by the OSN service providers are typically concentred where the data of the OSN users is stored. Since the OSN service provider have all the personal details of users stored at a central server that may be utilized by the service provider like to know the interestedness of a user on various things or some hackers may mug private information or in worst case the service provider may allegedly sell that information to some other third party organization. Therefore, the current centralized online social networks (COSNs) has raised the serious threats to privacy [17,18,19].

Among various clarifications to address the privacy concern in case of OSN data on centralized server the most popular solution is to encrypt users data [20,21]. In this approach the user data is first encrypted using some secret key and the same secret will again be encrypted with the corresponding friend's public key. Once the encrypted data is revived by the user's friend it also receives a secret key. So it in turn decrypts the secret key first with its own private key and then the encrypted data will be decrypted with the secret key. However it is popular approach to ensure the privacy of the user's data but there are some disadvantages also for this approach. One of the detriments of this approach is managing the friends of a user as a user may have large number of friends which may add or delete over a period of time. So it is quite cumbersome to manage the keys of multifarious friends. Secondly encryption and decryption

of user's data with the secret key is a overhead for the system.

Generally when we store data storage in cloud, low cost is the most important and imperative issue [22]. Several improvements have been made in the past to optimize the cost effectiveness of data stored in cloud. As in [23, 24] they primarily focus on storing the provenance of data and will restore the data on demand. [25, 26, 27] have proposed some distinction that tries to find out the effective trade-off between stored data and the cost of computation for the data. Some other impends of cost curtailment may include supervising the data redundancy in the cloud cluster. As in [28, 29] different algorithms have been proposed for intermittently and non-intermittently used data.

With the increasing emergence of Big Data, Clouds Infrastructure-As-A-Service (IAAS) leverages Big Data by actualizing the computation on virtual machines. Hadoop [30] installed on virtual machines is being used as one of the common solution to be used by many organizations by conjoining the services of Hadoop's map/reduce architecture in Cloud. A similar approach of monitoring Big Data in cloud is synchronic in [31]. A framework named as Sailfish was harbingered which is being used as a new map/reduce environment for supervision of Big Data.

III. Problem Statement

Twitter concedes to access and stockpile its data publicly using their API's (restful and streaming) which could be used for multifarious sorts of analysis and research findings. Although multitudinous twitter analysis systems are procurable that offer services to analyze the twitter data for users. Some of them are using apportioned servers for real time as well as archived twitter data for succouring to their user needs. But the cost of accessing and stockpiling twitter data is quite high and it has cocksure limitations too (like rate limit etc.). The data congregated is not subject specific as twitter allows annexing its data based on a particular keyword, language or geo-location. Suppose a political party is predisposed in analyzing its popularity in a particular region they need gather all the data from that region and then perform analysis on the whole dataset which involves high storage and analysis cost. In order to address this problem we have introduced the concept of "Twitter Mart". Twitter mart is a congregation of twitter data based on specific subject or topic. This will expedite the users to percolate analysis on the data of specific domain only and help in diminishing the computation cost.

Nowadays, the cloud computing and their services are so much in vogue that it is being perpetually used as a storage platform by many organizations. Especially in case of Twitter Mart, cloud storage will be the best choice as it allows scaling up and down the capacity on demand.

Keeping all the above stated problems in mind we have contemplated a Hybrid Cloud based storage framework for Twitter Mart that will assist the service providers in tapering down their computation and storage cost and provides an efficacious storage.

IV. Cost Affecting Factors in Cloud

Earlier organizations used to build their own private cloud due to destitution of trust and privacy and with the limited storage of data. Nowadays many organizations have started to outsource their data to various public cloud service providers due to advancements in various services and securities in their cloud infrastructure. Although a majority of the organization are opting for Hybrid Cloud storage that consists of in-house resources as well as the public resources. Hence Hybrid Cloud storage is the most commonly used architecture nowadays. But when we deliberate the factors that affect cost in the cloud it may vary from Public, Private as well as in Hybrid Cloud. Some of the most customary factors in Hybrid Cloud cost calculation are as follows [7]:

Cost for In-house Resources: This involves the cost of various in-house resources like electricity, labour cost, and cost of acquiring and managing separate data centre. Apart from this it also incur software licence cost and also some additional cost of planning and strategy making for the data centre.

Cost for Public Resources: Since the public resources are owned by some cloud service provider so it involves the cost of pricing model, time period, data size etc. Some of the adjacent cost may involve the cost of computation, data communication etc.

Public Private Cloud Interaction Cost: In Hybrid Cloud some of the data and processing is done on in-house resources and some is done on public resources, so cost of partitioning data in public and private cloud and the cost of workload distribution will also an important factor for consideration.

Miscellaneous Cost: Other costs that plays an important role in overall cost calculation are costs related to decision making for cloud adoption and the selection of cloud service provider, cost of data migration, portability and deployment etc. It also incurs the cost of training, maintenance and support.

In addition to the above factors the cost of data storage in cloud may embrace some other factors like organizational factors, environmental factor, various usage and services patterns etc.

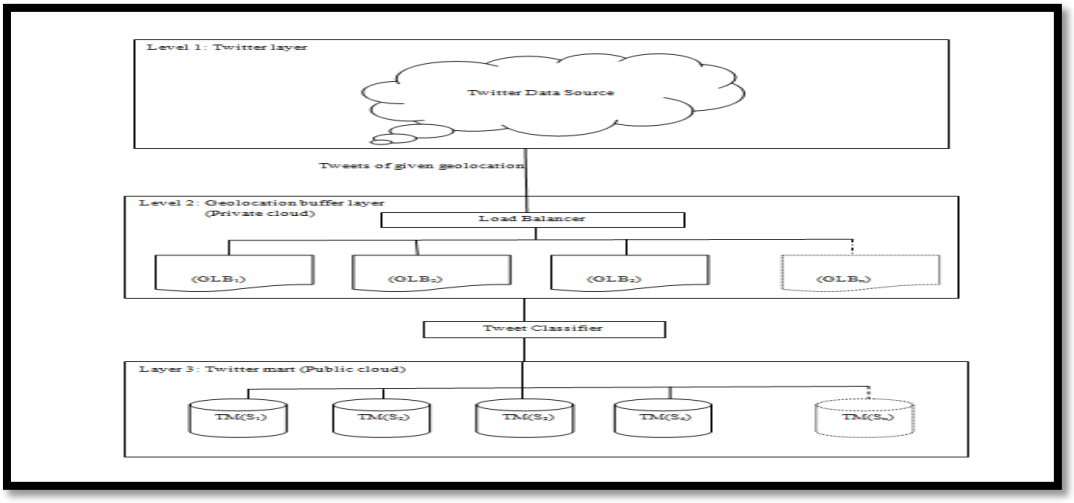

Fig.1. Cloud Storage Model

V. Proposed Cloud Storage Model

We are proposing a three layered distributed cloud architecture that will store data from twitter into pre-classified clusters known as "twitter mart". Three layers are described below:

Layer 1 is the twitter layer from where we are extracting tweets based on some pre-defined geolocation.

Layer 2, is the geolocation buffer layer where the tweets extracted from layer 1 are stored. Geolocation buffer layer consists of a load balancer and multiple storage units. The load balancer manages the high velocity of tweets extracted from layer 1 and the storage units will store the tweets of specific locations. The data stored in this layer is used for segregation in the next layer. We are storing the data of layer 2 in private cloud so that once the data is classified in the next layer we will purge the remaining data to optimize the storage cost of private cloud. Layer 3 is the twitter mart layer where we store the classified data from specific location based on some predefined subjects. This layer is in public cloud so that it can be accessed by any user for various purposes like sentiment analysis, popularity finding etc.

Suppose *CL*=(*CDC*, *L*, G, *S*) be the Cloud that will store data from the Twitter, where *CDC* = Cloud Data Centres, *L* = Communication Links, G = location and *S* = Subject Clusters (like. politics, sports, education, entertainment etc). The twitter data source allows us to get the data from twitter by using some API's like streaming API of twitter that allows to collect real-time tweets from the twitter[32]. Now the streaming API offers many request parameters to collect data [5], for example language, track, location etc. We are using location parameter to get the data from a particular geolocation. Following steps are involved in the whole storage process (Fig.1.):

1. Collect tweets from different geolocations on demand.
2. Send the data to geolocation buffers GLB_i in each cluster and an index of location parameter so that we can manage the tweets data specific to different geolocations.
3. Extract Subject of tweets from the geolocation buffer, classify them according (subject corresponds to the topic to which the particular tweet belongs for example politics, sports, education etc) and store them into public cloud twitter mart $TM(S_i)$.
4. Create a new subject cluster within the existing cluster (optional in case when the subject clusters are not known in advance).

For classification of tweets, we use Naive Bayes algorithm that work in two steps one for training and one for testing. So, first we train Naive Bayes classifier and determine the requisite parameters for our testing.

Algorithm
// Suppose *S*= set of subjects
// *GLB*(*G*) = Geolocation Buffer from location G
// S_k = Subject Cluster k.
// *T*(*t*) = Tweets from a specific geolocation.
// *T* = <T_i, S> Tuple consist of training tweet with corresponding subject

Step 1 Training

Training (*S*, *T*)

1) $V \leftarrow ExtractVocabulary\ (T)$
2) $N \leftarrow CountDocs\ (T)$
3) for each $s \in T$
4) do $N_c \leftarrow CountDocsInclass(T, S)$
5) $prior[c] \leftarrow N_c / N$
6) $text_c \leftarrow ConcatenateText(T, S)$
7) for each $t \in V$
8) do $T_{ct} \leftarrow CountTokenTerm(text_c, t)$
9) for each $t \in V$
10) do $condprob[t][c] \leftarrow \dfrac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$

return *V*, *prior*, *condprob*

Once the training is done we can use our cluster creation algorithm to form the Twitter Mart.

Cluster Creation

1) for each tweet $t_i \in T(t)$
2) if $\prod_{location}(t_i)$=G
3) send t_i to $GLB(G)$
4) $\forall t_i \in GLB(G)$
5) s_i = *ExtractSubjectUsingNaiveBayes*(t_i)
6) if $s_i \in S_k$
7) Store t_i in $TM(S_k)$
8) Else:

Purge t_i from $GLB(G)$

ExtractSubjectUsingNaiveBayes(t_i)

1) $W \leftarrow ExtractToken(V, t_i)$
2) for each $c \in S$
3) do $score[c] \leftarrow \log prior\,[c]$
4) for each $t \in W$
5) do $score[c] += logcondprob[t][c]$
6) return $\text{argmax}_{c \in C}\, score\,[c]$

A. Analysis

Let N be the total number of tweets in the collection and n be the number of clusters/classes. Each tweet will be extracted and analyzed by the Naive Bayes algorithm. So the total time complexity will be the sum O(N) and the complexity of Naive Bayes algorithm. The time complexity to extract the algorithm will be O(N).

For Naive Bayes classification, the total complexity will be sum of training the classifier and then testing it. For training the complexity will be dependent on $n^*|V|$ and $N^*|L|$ where, $|V|$ is the vocabulary length, $|L|$ is the number of words in the tweets.So total time complexity for training is $n^*|V| + N^*|L|$.

For testing, the time complexity depends upon the number of token N_t and the type of classes n. So the total time complexity of Naive Bayes algorithm is O($n^*|V|$) + O($N^*|L|$)+ O(n^*N_t) which is linear. Also the time complexity to create a new cluster is O(n)

Hence, the total time complexity of our proposed algorithm is O(N) + O($n^*|V|$) + O($N^*|L|$)+ O(n^*N_t) + O(n) which is linear order of time..

VI. Cost Calculation

This model there are two costs involved. One is the cost of storage length of data in both public and private cloud and the second is the computation cost for the analysis.

As we are using the cost model for hybrid cloud proposed in [22]. Consider the storage function $C(t) \rightarrow R$ that maps storage as a function of time and maps it is with the requisite resources R. As we know the storage need will increase with time so $C(t)$ will be an increasing function and it is assumed to be positive. Since we are considering the cost of hybrid cloud, so it will include the cost of private as well as public cloud [21].

Suppose C_1 is the cost of private cloud and C_2 is the cost for public cloud and both C_1 and C_2 are different so we will calculate both the costs in order to calculate the total cost.

Now cost calculation for Private cloud (C_1) depends upon various factors such as initial cost of acquisition and maintenance of data(C_i), also as the storage increases with time we need to estimate the increase in storage capacity for a time interval t say ($p_o(c(t))$).So, total cost for private storage is:

$$C_{1=}\ C_i\ + c(t)p_o c(t)\ t \tag{1}$$

Where $c(t)$ is the estimate of storage capacity needed for the future during acquisition time interval. Since we are estimating the capacity and it may contain some erroneous component as well so form [21]

$$c(t) = k_e k_r c(t)$$

$$C_{1=}\ C_i\ +\ k_e k_r c(t)\ p_o(c(t))\ t \tag{2}$$

$$\frac{d}{dt}C_1\ =\ 0 + k_e k_r p_o(\frac{d}{dt}c(t)t + c(t))$$

Since, $\frac{d}{dt}C_i =$, as C_i is the constant cost.

Now C_1 will increase as the t increases since k_e, k_r, $c(t)$ and t all are positive. Similarly for public cloud the factors are cost of charging period(C_k) and cost of acquisition for the time period t. So,

$$C_2 = C_k + \int_1^t c(t)p_1(c(t))dt$$

Also $C_2 > 0$ since $\frac{d}{dt}C_2 > 0$

So the combined cost of hybrid cloud is:

$$C = C_1 + C_2$$

$$C = k_e k_r c(t)\ p_o(c(t))\ t + \int_1^t c(t)p_1(c(t))\ dt$$

$$C=\ k_e k_r p_o c(t)\ t + p_1 \int_1^t c(t)dt \tag{3}$$

In our framework we are analysing our data before storage so during data pre-processing a large portion of data will be removed in the form of noise. So the total amount of data left will be much smaller than the expected storage volume.

Suppose X% of twitter data is removed from the main corpus as noise.

So the total length of data will be reduced by X%.

$$c(t)= c(t) - \frac{X}{100}c(t)$$

$$c(t) = \frac{(100 - X)\ c(t)}{100}$$

from eq. (3).

$$\textbf{Storage Cost} = k_e k_r p_o \frac{(100 - X)\ c(t)}{100} t + \frac{(100 - X)\ c(t)}{100} p_1 \int_1^t c(t)dt$$

As we know a tweet is of maximum length of 140 characters and it contains many insignificant words like stopwords, URL's or slangs which will be removed during pre-processing. From our experiments we came to a conclusion that the value of X may lie form 30 to 40% and it also varies from place to place.

Computation cost includes the CPU cycles used in order to perform read/write operations. In our case the cost of read and write cost will be different since we are analysing our data before writing to disk so writing cost include the analysis cost as well. Let the CPU executes I instructions per second and it the cost C_I .

$$\text{Write cost} = C_I * [O(N) + O(n*|V|) + O(N*|L|) + O(n*N_t) + O(n)]$$

If the disk reading cost per byte if 'd' and the total bytes to read are 'B', then total cost is dB. So,

$$\textbf{Computation cost} = C_I * [O(N) + O(n*|V|) + O(N*|L|) + O(n*N_t) + O(n)] + dB$$

Overall cost = Storage cost + Computation cost.

VII. Experimentation and Results

We have used python as our programming language to congregate data from the Twitter source using Twitter streaming API and we have used the geolocation parameter of streaming API. For our experiment we have gathered the data of Delhi region. Data is collected on some text initially before it is scrutinized then data is moved to the virtual machines having Apache Spark installed on it. The convened data is analyzed with the classification algorithm in order to extract and classify tweets.

Experiment 1: For this experiment, we have used our own training data set of 4500 tweets which consists of 1500 tweets from each category (i.e. politics, sports and entertainment) and then we have taken 5000 tweets from a single geolocation buffer GLB_1 and classified the tweets into three classes Politics, Sports and Entertainment using Naive Bayes text classification algorithm and made the Twitter Mart. We have seen that after pre-processing only 4300 tweet remain for further classification out of which only 629 qualified for twitter mart storage and remaining tweets were purged. The results can be seen in Fig. 2.

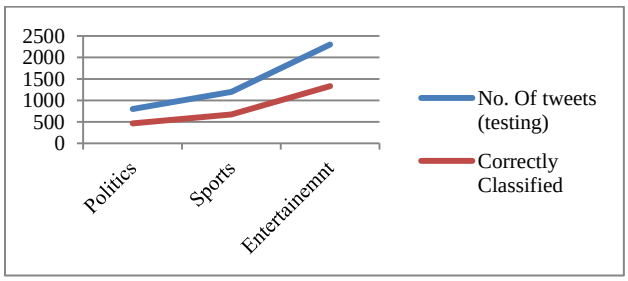

Fig.2. Result Naive Bayes

The accuracy of the classier is found to be around 56% which is not very good. However, this can be improved by using some good classifier whose accuracy is more. The following table (Table 1) illustrates the size of testing and training data:

Table 1. Training and testing data

	Politics	Sports	Entertainment
No. Of tweets (Training)	1500	1500	1500
No. Of tweets (testing)	800	1200	2300
Classified	464	672	1334

Once the data is regimented, we have made three clusters of each class based on geolocations . Also it can be seen that the size of twitter mart is very small as compared to the original data (Fig.3) in the geolocation buffer which reduces the storage cost. It may be noted that in case we decide to classify tweets directly from twitter source, the pre-requisite is to have trained Naive Bayes classifier and also even with such a classifier the program that extracts the tweets crashes with good frequency and hence the problem is not addressed properly. In contrast our proposed model handles this scenario without any performance issues.

***Experiment 2*:** Since the performance of Naive Bayes classifier is low, so we have experimented our model on another classifier proposed in [35]. For this classifier we have used three different word lists (for politics, sports and entertainment) each having 200 words initially and perform iterations to classify tweets. Also the training data form Naive Bayes classifier remains same (i.e. total 4500 tweets for training 1500 from each category).

Table 2. Comparison of Naive based and Modified Naive bayes Algorithm

total tweets	1000	2000	3000	4000	5000	6000
Naïve bayes	550	1150	1790	2500	2670	3300
Modified Naive bayes Algorithm	30	80	160	240	300	390

We have applied both of the algorithms on our proposed model and we have find out the results recorded in Table 2. From fig.4 it can be seen that the performance of modified Naive Bayes classifier is better than the Naive Bayes classifier. So, if accuracy is the only

concern we can use the modified Naive Bayes classifier for creating the Twitter Mart.

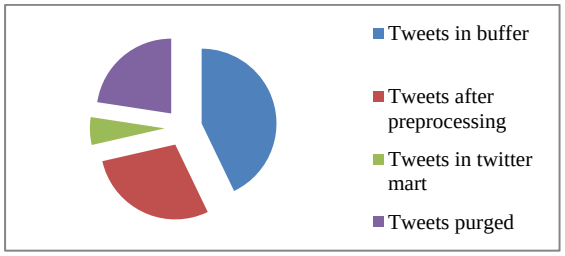

Fig.3. Twitter Mart

Once the Twitter Mart is built we can use it for further analyses like to find various trends on a particular region on a particular subject. As we can see in Table 3 that the tweets in the three cluster permeates to three different subjects which can be directly accessed if somebody is interested in knowing about a particular subject at a particular instance of time from a given geolocation.

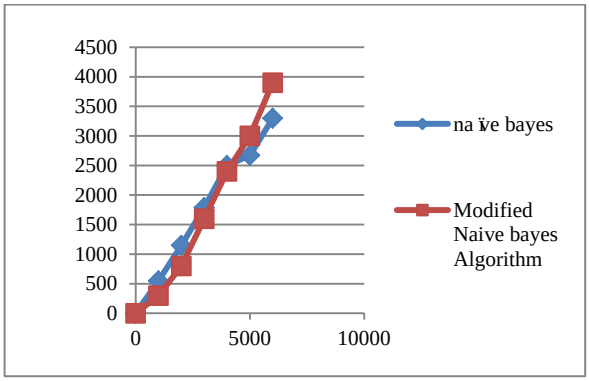

Fig.4. Comparison of Naive bayes and Modified Naive bayes[35]

Table 3. Twitter Mart

Cluster Id	Custer Name	Tweet Mart
1	Politics	1. The only thing that Modi is stopping these moron filmakers crying emergency-emergency from doing is dancing naked on 2. constitution do not allow you to insult the honour given by the Government just to make happy your political friends 3. pm promised ganaga be cleaned in 18 months whats the progress? 4. The conspiracy seems too deep being hatched by congress. After all this was the party which ensured US visa ban for @narendramodi 5. The conspiracy seems too deep being hatched by congress. After all this was the party which ensured US visa ban
2	Sports	1. Dear #ICICI team, according to your team, they have fixed the date for my issues regarding fraud it was 28th Oct. wt next. 2. Victor Valdes hopes to stay in the Premier League if Manchester United will allow him to leave in January, his agent has said. 3. @englandcricket does he know how to win matches? 4. BCCI is looking to give veteran batsman VirenderSehwag a formal send-off in the fourth Test in Delhi starting December 3. 5. Afghanistan's rise in International Cricket is to be lauded, without any adequate resources; funds they keep on performing; improving.
3	Entertainment	1. A new show on @sabtv where An ALTO Car wil help Finding solutions #ChaltiKaNaamGaadi 2. Song of nature by #davidgerstein #brunoart #charlesfazzino 3. All the best selling #books, they even have an #airport edition chart! #India #travel #delhi 4. Perfect! That is NikolinaNikoleski Dance Company. #superb #lufthansa 5. A college dropout giving lectures in IIT. Indian Education, in its current form, is all about making sophisticated servants.

VIII. Discussion

Novelty of Framework: Our framework is novel in the sense that we are proposing a 3-layered architecture where we are using buffers for load balancing and ephemeral storage for overseeing high velocity data. The data is then gathered into Tweet Marts. Also as a common practice in most of the frameworks, the clusters are made on the basis of size or in some cases may be on the geolocation but in our case our clusters are made on the basis of a particular subject so whenever required we can directly query the pre-built cluster rather than sending the query to every cluster and analyzing all clusters to find the answer of the query. As you can see in Table 2 three subjects has been identified i.e. Politics, Sports and Entertainment, so we have created three clusters representing each subject respectively. Now these three clusters can be used for further analysis. For example, to find the acclaim of a political leader at a particular instance of time in a specific region or to find the popularity of a particular sports etc. Our framework uses only two algorithms one is to find the subject of any tweet and second is to create a cluster based upon the specific subject. For the first algorithm our framework is flexible to allow any available algorithm such as Naive Bayes, Support Vector Machine, and Passive Aggressive Algorithm etc. So depend on the type and resource availability one case use any of the approach.

Cost Reduction: Majority of available frameworks use store as such and retrieve all approach [33, 34], they basically allow stockpiling all data in any of the cluster and then they apply analysis during data retrieval which is quite extravagant. Since the data is so huge it will take a lot of time during data retrieval which leads to poor response time of user's query and also the cost of storing that unprocessed data is too large. In our framework the cost of storage is relatively small as compared to the

conventional systems. Also the running time of our proposed solution is of linear order.

IX. Conclusion

Social networking sites like Facebook, Twitter, Google+ etc. generate humongous amount of data which can be used for analysis. Twitter is the most popular source of publicly available Big Data but the data provided by twitter is not subject or domain specific which incur high cost of storage and computation. As the volume of such data is very huge so it requires proper panels and cloud computing is a viable solution for the storage of such huge amount. In order to stockpile the data in cloud various factors are involved which affect the cost of storage and analysis in the cloud. In this paper we have proposed a framework for storing data in the cloud efficiently and in subject specific clusters on VMs (Virtual Machines). The results shows that once we classify the data and store them according to the subject cluster then the cluster will be used for further analysis like sentiment analysis, popularity finding etc.

References

[1] MAPR Article: "5 Google Projects That Changed Big Data Forever": (accessed Mar 27, 2017) https://www.mapr.com/blog/5-google-projects-changed-big-data-forever

[2] Sqluthority.com Blog : "Big Data – What is Big Data – 3 Vs of Big Data – Volume, Velocity and Variety – Day 2 of 21": (accessed Mar 27, 2017)https://blog.sqlauthority.com/2013/10/02/big-data-what-is-big-data-3-vs-of-big-data-volume-velocity-and-variety-day-2-of-21/

[3] Keywebmetrcs article “How Big Data drives Facial Recognition”: (accessed Mar 27, 2017): http://www.keywebmetrics.com/2013/08/big-data-drives-social-graph/

[4] Article on "Global social media research summary 2017": (accessed Mar 27, 2017): http://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/

[5] Ali M, Nasr ES, Geith M, Benefits and Challenges of Cloud ERP Systems- A Systematic Literature Review, Future Computing and Informatics Journal (2017), doi: 10.1016/j.fcij.2017.03.003

[6] Article on “Public Cloud Definition” (accessed Mar 27,2017)”http://searchcloudcomputing.techtarget.com/definition/public-cloud

[7] A. Shakimov, A. Varshavsky, L.P. Cox, et al., Privacy, cost, and availability tradeoffs in decentralized OSNs, in: The 2nd ACM Workshop on Online Social Networks, ACM, 2009, pp.13–18.

[8] Y. Ding, M. Neumann, D. Gordon, T. Riedel, T. Miyaki, M. Beigl, W. Zhang, L. Zhang, A platform-as-a-service for in-situ development of wireless sensor network applications, in: Networked Sensing Systems (INSS), 2012 Ninth International Conference on, 2012, pp. 1–8.

[9] J. Zhou, T. Leppanen, E. Harjula, M. Ylianttila, T. Ojala, C. Yu, H. Jin, L. Yang, Cloudthings: A common architecture for integrating the internet of things with cloud computing, in: Computer Supported Cooperative Work in Design (CSCWD), 2013 IEEE 17th International Conference on, 2013, pp. 651–657

[10] A Study on Data Storage Security Issues in Cloud Computing Naresh vurukonda1, B.Thirumala Rao2

[11] Fabregas, Aleta C., Bobby D. Gerardo, and Bartolome T. Tanguilig III. "Enhanced Initial Centroids for K-means Algorithm." (2017).

[12] C. Wang, Q. Wang, K. Ren, N. Cao, W. Lou, Toward secure and dependable storage services in cloud computing, IEEE Trans. Services Comput. 5 (2)(2012) 220–232.

[13] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, A.V. Vasilakos, Security and privacy for storage and computation in cloud computing, Inform. Sci. 258 (2014) 371–386.

[14] S.M.S. Chow, Y. He, L.C.K. Hui, S.M. Yiu, Spicesimple privacy-preserving identity-management for cloud environment, in: Applied Cryptography and Network Security, Springer, Berlin, Heidelberg, 2012, pp. 526–543

[15] R.D. Dhungana, A. Mohammad, A. Sharma, I. Schoen, Identity management framework for cloud networking infrastructure, in: IEEE International Conference on Innovations in Information Technology (IIT), 2013, pp. 13–17.

[16] M.L. Hale, R. Gamble, Risk propagation of security SLAs in the cloud, in: IEEE Globecom Workshops (GC Wkshps), 2012, pp. 730–735.

[17] B. Krishnamurthy, C.E. Wills, Characterizing privacy in online social networks, in: Proceedings of the First Workshop on Online Social Networks, ACM, 2008, pp.37–42.

[18] C. Zhang, J. Sun, X. Zhu, et al., Privacy and security for online social networks: challenges and opportunities, IEEE Netw. 24(4) (2010).

[19] B. Krishnamurthy, C.E. Wills, Privacy leakage in mobile online social networks, in: Proceedings of the 3rd Conference on Online Social Networks, USENIX Association, 2010.

[20] K. Graffi, C. Gross, P. Mukherjee, et al., LifeSocial.KOM: a P2P-based platform for secure online social networks, in: 2010 IEEE Tenth International Conference on Peer-to-Peer Computing (P2P), IEEE, 2010.

[21] S. Buchegger, D. Schi öberg, L.H. Vu, et al., PeerSoN: P2P social networking: early experiences and insights, in: The Second ACM EuroSys Workshop on Social Network Systems, ACM, 2009, pp.46–52.

[22] Laatikainen, Gabriella, OleksiyMazhelis, and PasiTyrvainen. "Cost benefits of flexible hybrid cloud storage: Mitigating volume variation with shorter acquisition cycle." Journal of Systems and Software 122 (2016): 180-201.

[23] Borthakur, Dhruba. "HDFS architecture guide." Hadoop Apache Project 53 (2008). http://hadoop.apache.org/docs/r1.2.1/hdfs_design.pdf

[24] Adams, Ian F., et al. "Maximizing Efficiency by Trading Storage for Computation." HotCloud. 2009.

[25] Yuan, Dong, et al. "A cost-effective strategy for intermediate data storage in scientific cloud workflow systems." Parallel & Distributed Processing (IPDPS), 2010 IEEE International Symposium on. IEEE, 2010.

[26] Yuan, Dong, et al. "On-demand minimum cost benchmarking for intermediate dataset storage in scientific cloud workflow systems." Journal of Parallel and Distributed Computing 71.2 (2011): 316-332.

[27] Yuan, Dong, et al. "A data dependency based strategy for intermediate data storage in scientific cloud workflow systems." Concurrency and Computation: Practice and

Experience 24.9 (2012): 956-976.
[28] Mao, Bo, et al. "Read-performance optimization for deduplication-based storage systems in the cloud." ACM Transactions on Storage (TOS) 10.2 (2014): 6.
[29] Clements, Austin T., et al. "Decentralized Deduplication in SAN Cluster File Systems." USENIX annual technical conference. 2009.
[30] White, Tom. *Hadoop: The definitive guide*. " O'Reilly Media, Inc.", 2012.
[31] S. Rao, R. Ramakrishnan, A. Silberstein, M. Ovsiannikov, D. Reeves, Sailfish: A framework for large scale data processing, in: Proceedings of the Third ACM Symposium on Cloud Computing, SoCC '12, ACM, New York, NY, USA, 2012, pp. 4:1–4:14.
[32] Twitter Developer Documentation :(accessed on Mar 27, 2017): ttps://dev.twitter.com/streaming/overview
[33] Fazio, Maria, et al. "Big data storage in the cloud for smart environment monitoring." Procedia Computer Science 52 (2015): 500-506
[34] Vinay, A., et al. "Cloud based big data analytics framework for face recognition in social networks using machine learning." Procedia Computer Science 50 (2015): 623-630.
[35] Khan, Imran, et al. "An efficient framework for real-time tweet classification." International Journal of Information Technology: 1-7.
[36] Venkatraman, Sitalakshmi, et al. "SQL Versus NoSQL Movement with Big Data Analytics." International Journal of Information Technology and Computer Science (IJITCS) 8.12 (2016): 59.
[37] Diaby, Tinankoria, and Babak Bashari Rad. "Cloud Computing: A review of the Concepts and Deployment Models." International Journal of Information Technology and Computer Science (IJITCS) (2017).

Authors' Profiles

Imran Khan is a phd research scholar in FTK-Centre for Information Technology, Jamia Millia Islamia, New Delhi, India. His current research works are in Big Data and Cloud Computing. He received BSc (Hons) Chemistry degree and MCA (Master of Computer Science and Application) degree from Aligarh Muslim University (AMU), Aligarh, India.

How to cite this paper: Imran Khan, S. Kazim Naqvi, Mansaf Alam, Mohammad Najmud Doja, S. Nasir Aziz Rizvi, "An Efficient Framework for Creating Twitter Mart on a Hybrid Cloud", International Journal of Information Technology and Computer Science(IJITCS), Vol.9, No.10, pp.59-67, 2017. DOI: 10.5815/ijitcs.2017.10.06