PAPER • OPEN ACCESS

Enhanced K-means clustering with encryption on cloud

To cite this article: Iqjot Singh *et al* 2017 *IOP Conf. Ser.: Mater. Sci. Eng.* **263** 042057

View the article online for updates and enhancements.

Enhanced K-means clustering with encryption on cloud

Iqjot Singh, Prerna Dwivedi, Taru Gupta and P. G. Shynu

VIT University, Vellore-632014, Tamil nadu, India

Email: pgshynu@vit.ac.in

Abstract. This paper tries to solve the problem of storing and managing big files over cloud by implementing hashing on Hadoop in big-data and ensure security while uploading and downloading files. Cloud computing is a term that emphasis on sharing data and facilitates to share infrastructure and resources.[10] Hadoop is an open source software that gives us access to store and manage big files according to our needs on cloud. K-means clustering algorithm is an algorithm used to calculate distance between the centroid of the cluster and the data points. Hashing is a algorithm in which we are storing and retrieving data with hash keys. The hashing algorithm is called as hash function which is used to portray the original data and later to fetch the data stored at the specific key. [17] Encryption is a process to transform electronic data into non readable form known as cipher text. Decryption is the opposite process of encryption, it transforms the cipher text into plain text that the end user can read and understand well. For encryption and decryption we are using Symmetric key cryptographic algorithm. In symmetric key cryptography are using DES algorithm for a secure storage of the files. [3]

1. Introduction

Traditionally, to store and compute a huge amount of data on cloud was difficult because of querying, transferring data etc. was insufficient to deal because they need a software to run simultaneously on millions of servers to get the desired output. To get the output in less measure of time big data came into existence. Big-data is a term used for large data sets that are examined and reckoned to give a fruitful result. Cloud is an abstract term for internet that allows us to store our data on the internet rather than on our system. [8]

In cloud, enormous amount of data is to be stored and in distributed data centres which makes data processing a complex and a time consuming operation. The data and services that a user wants to use are stored at different geographical areas. To provide the data and services to the authenticated user is a challenge for the cloud service provider [5]. Because of cloud computing flexibility and on demand services, the expansion of cloud is inevitable.

Hadoop is a service distributed all over the cloud and is used by the organizations to furnish cloud computing services. It stores data using HDFS (Hadoop Distributed File System) which is based on master-slave architecture. HDFS allows parallel computing throughout several nodes in a cluster. [10]

K-means is one of the most elementary algorithms used for unsupervised learning that is used to solve the clustering problem. Clustering is the procedure of segmenting a group of data points into a minor number of clusters. It is used to calculate distance in cloud to calculate distance between the centroid of a cluster and its data points. [17]

Hashing is a technique to convert a group of data points into indexes. It stores the data item in such a way that helps to locate it uniquely. A hash table is a collection of items which stores data in a way that will be easy to locate in future.

Encryption is a process to transform electronic data into non readable form known as cipher text. Decryption is the opposite process of encryption, it transforms the cipher text into plain text that the end user can read and understand well [3].

Symmetric key algorithms are cryptographic algorithms which use same secret key for both encryption and decryption. Symmetric key encryption can be used for either stream cipher or block cipher. [4] Stream cipher encrypts the digits of the message one at a time and block cipher takes a group of bits and encrypt it as a complete group.

2. Literature Survey

Keji Hu and Wensheng Zhang [1], Leak of customer's sensitive data from cloud framework stays to be an issue in spite of advancement of cloud technology. One of the significant reasons for such holes to happen is the absence of effective encryption verification plans. For a cloud computing framework, computational quality of server must not be yielded while encryption and verification is applied, which includes certain level of trouble in building up an efficient verification process.

In this paper an effective triggered method to confirm the information encryption on server side. This method is most suitable and faster than the other encryption methodology. Other than the efficiency, this strategy can be applied to both the file information and altered information. The above are significant elements that make our plan to be more practical and appropriate than past solutions. The issue this paper focuses at is to give an arrangement that provide advantage on cloud server to save the information in encrypted way and clients can verify their data that is really presence in the form of encrypted at the servers storage. It is very difficult to keep the data secure if clients encrypt and decrypt the data by own and keep the keys. A cloud server has very effective processing power that each customer may long to use.

K.Brindha, N.Jeyanthi [2], Cloud computing model permits the clients to get the secured data from cloud environment without the assistance of utilizing the equipment of system. For the successful use of the information from the cloud supplier, the person who has authority on data, they encrypts the information and then deploy the information on cloud server. To secure the information in the cloud we have managed through the security issues prior. In this paper we have proposed the thought how the information could be made more secured utilizing cryptographic methods .This model not just builds the trust among the clients to send their data with full dependability and furthermore helps in the authorization of the clients. In the cloud computing environment Cloud storage is the virtual space where we store the client information. The major issue is that client has no control on data. Cloud provider has the control over client information. So there is no privacy of client data on server, this generates the major issue on cloud. The expression "Information privacy" characterizes that exclusive approved individuals can utilize the information; "Information integrity" refers to the information that has not been modified. Information accessibility signifies to the accessibility of the information when required in time. We are using cryptography technique at the user end so client has privacy on data.

B.Harikrishna, Dr.S.Kiran, R.Pradeep kumar Reddy [3] gave a survey about the available cryptography schemes for sensitive data on cloud. As we know that in today's era cloud is highly demanded technology because it is like an outsourcing of IT services and communications. It helps in facilitating the services and resources which are required at that instance of time and is paid for that only, that is cloud follows the "pay on use" scheme. Sometimes cloud fails to provide the security to the sensitive information uploaded on the cloud, so to overcome this situation the paper proposed an algorithm which is completely different from the other existing algorithms. Information is only said to be secured when it is confidential, integrated and available for the authorized and authenticated user only. The paper discussed about various cryptography algorithms (key oriented and keyless oriented) in order to maintain the security in the cloud. The paper also discussed about Third Party Auditor (also known as TPA) which has rights to check the integrity of the files of the behalf of the user and can release an appraisal report to the user. The proposed algorithm provides the security to the stored data and to the keys as well. There is always a chance for an insider attacker or an outsider attacker to attack the files on the cloud, so security is very important and the keyless algorithms are proved to be better than key based algorithms because they do not go against to the characteristics or the features of the cloud, namely authentication, demand access, integrity services and so on.

Esteves, Rui Maximo, Rui Pais, and Chunming Rong [7], described that clustering is a technique to group data items on the basis of distance from other data points. For calculating distance between data points we use Euclidean distance formula. K-means clustering is a widely used concept in the world of clustering. This algorithm comes with a disadvantage that it is not suitable for big data as it does give high performance when applied to a small data set and not to a large data set. Mahout is proved to be best algorithm because it is an inexpensive solution and also a promising one to the problems created by big data, but there is a lack of study in this field, so no one can promise that this test will give high performance. The Mahout project is still developing and at the moment there is no promising result for clustering. On applying Mahout in cloud we saw that when the numbers of nodes are increased, the CPU usage falls and the network usage increases.

3. Proposed Work

Cloud computing opens a new world for the users where they can work on software that are not present in their machine, and use the infrastructure and services that their machine can't handle. Cloud lets you to access your data from any part of the world. With so many facilities there comes a drawback in the name of security. Cloud provides enormous amount of data storage to store your data, but with providing storage it needs to provide security too. With the vast usage of cloud it is not easy to keep data secure because various servers are connected to the cloud and can be able to crack the security and view the data, moreover for searching a normal file in data in HDFS system it is time consuming so we introduces an enhanced k-means clustering algorithm with an addition of security on cloud. The K means clustering algorithm is enhanced by adding hashing to it so that we can be able to reach to that specific data node in comparatively less time.

3.1 Hadoop

Some text. With the fast development in computer science field, we get to deal with a huge amount of data which is mostly in unstructured format. All these data are really important for analysing and taking valuable decisions which is important for an organization and its growth. But the main problem is where to store this huge data which can be counted in TBs or even in PBs, because disk storage is very expensive and not a convenient way to store and at the same time the data has to be distributed among the systems. So to overcome all these problem, hadoop came into the scene. Hadoop is an open source platform provided to us which uses java programming framework to compute the operations on the data stored in it and facilitates the fast and rapid data transfer rates among various nodes present and still helps in allowing the system to operate even in case of any node failure. [17]

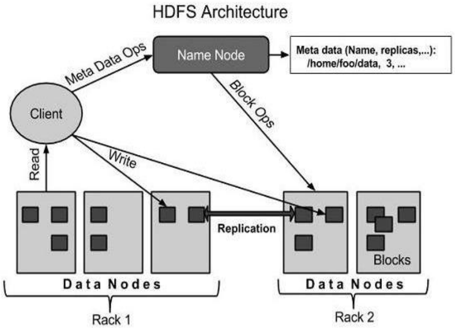


Fig. 1. HDFS structure

3.2 K Means Clustering

K means clustering is a clustering algorithm which divides the huge dataset in k clusters. By the use of MapReduce framework we can implement K mean clustering on java framework which can work on huge and vast data. The main aim of this type of clustering is to decrease the intra cluster space and increase the inter cluster space. With the help of this clustering we can find a suitable structure for an unstructured data. It is an efficient and easy technology to be understood and gives the best result when data are distinct and separated from each other. Since clusters form here are non hierarchal so they don't overlap with each other.

3.2.1 K Means Clustering Algorithm

Step 1: Randomly select k centroids

Step 2: The input file should contain the real data and the centroid too.

Step 3: With the help of "configure" function present in mapper class, we can read the data and store it in a data structure of our choice.

Step 4: Mapper reads the data file and outputs the closest centroid with the node to the reducer (known as intermediate output).

Step 5: Now its reducer work to collect all the data and compute the new respective centroid and emit.

Step 6: Under job configuration, both files are read and checked that if the difference between the old and new centroid is small than 0.1 then the convergence is attained else repeat from step 2.

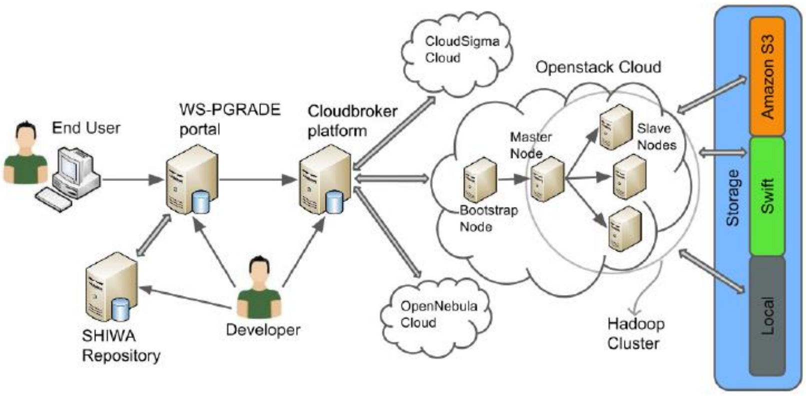


Fig. 2. Method implementation

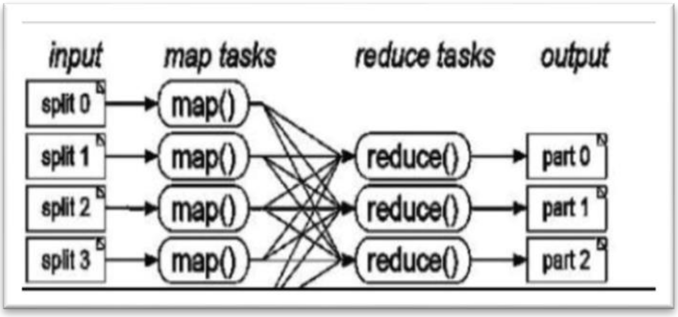


Fig. 3. K Mean Clustering process

3.3 Hashing

Hashing is a widely used technique for searching a data in a huge dataset. This technology uses a hash table and hash key value to compute. Hash table is a linear table in which the items are stored in such way that it is easier to allocate them and find them. Each position within the table is known as slot which is named by an integer value that starts with 0. The process of mapping between slots and item is known and hash function. This hash function accepts any value and returns the slot value from the range 0 to m-1. There are various types of hash functions such as mod m method, folding method and so on. Each of these methods has their own pros and cons depending upon the dataset.

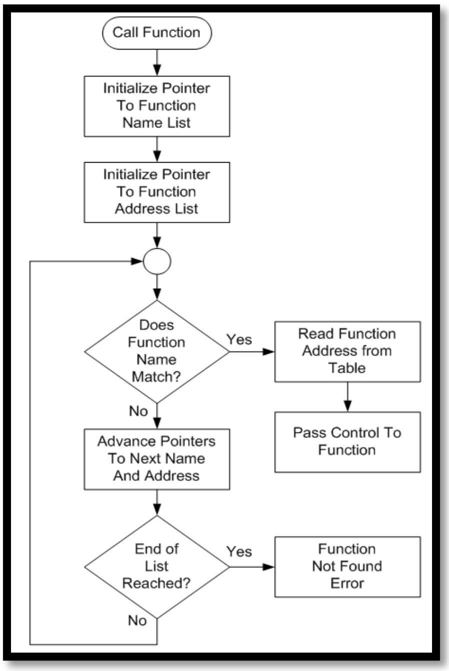

Call Function
Initialize Pointer To Function Name List
Initialize Pointer To Function Address List
Does Function Name Match?
Yes
Read Function Address from Table
Pass Control To Function
No
Advance Pointers To Next Name And Address
End of List Reached?
Yes
Function Not Found Error
No

Fig. 4. Hashing flowchat

In our proposed system we are using hash function known as remainder method. In this method the item is divided by the total range and then it is stored at the remainder position. It is the basic and easy to understand algorithm and hash clashes can be handled easily. Hash clash is a situation in which the two or more items get the same slot. SO to prevent the system from Hash clash we use linear chaining, in which the data sharing same slots are kept together by the help of linked list, so that the data can be retrieved faster and easily.

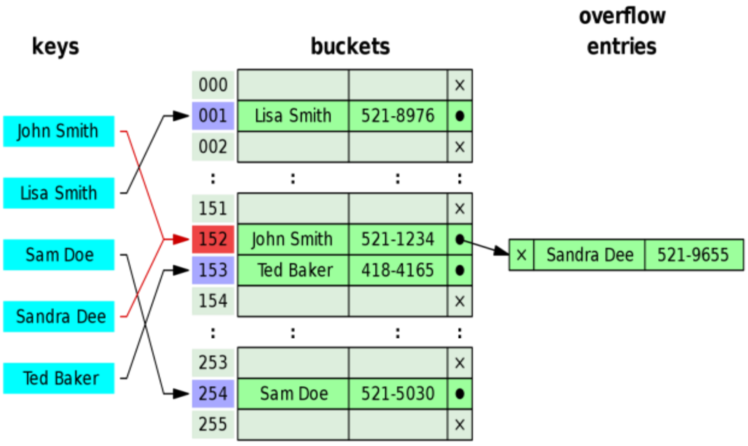


Fig. 5. Hash clash example

3.4 Symmetric Key Algorithm

There are many cryptographic algorithms available, but none of them is cent-percent safe. The hackers find out the loop holes in the algorithm and try to breach the security. To make the security on cloud strong we use Symmetric key algorithm.[1] This algorithm does not produce separate keys for encryption and decryption which finally saves time to a great extent, memory and cost related to key generation. It even provides security against elusive failure, server clouding and data alteration attacks.

Symmetric algorithm is an algorithm that is used for keeping the data secure and safe. This algorithm is used for cryptography that uses one secret key which is known by both sender and receiver.[3] It converts the plain text to cipher text and decrypts the cipher text to plain text using that key. The key is maintaining the privacy between both parties. In symmetric we are using two types of ciphers.

- In stream ciphers whole data is encrypted one at a time.
- In block ciphers we are not taking whole data one at a time, cipher take a number of bits and encrypt that as a one unit.

In this paper we are using Symmetric stream cipher, our whole data is encrypted at once.

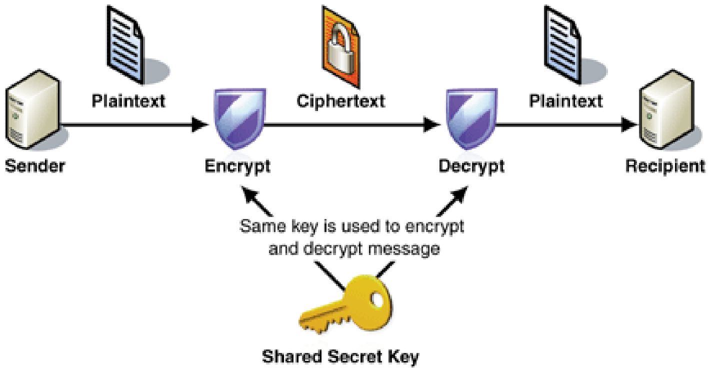


Fig. 6. Symmetric Key Architecture

To maintain the security in the hadoop we used DES algorithm, which can take 64-bit of plain text and generates 56 bit cipher key. The reason for using this algorithm is that, this algorithm covers the two main features of security, namely, avalanche effect and completeness.

Completeness means that each and every bit within the cipher text is fully dependent on the bits in plain text.

Avalanche effect states that if there is any minute change within the plain text, then there will be a huge change in the cipher text.

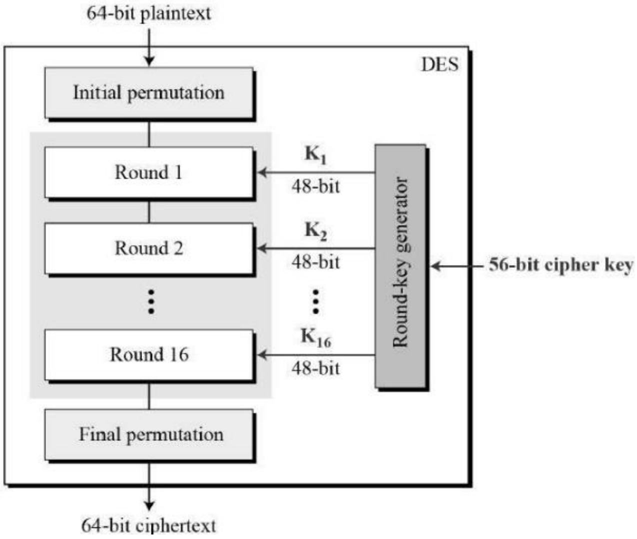


Fig. 7. Data Encryption Standard Architecture

3.5 Proposed Architecture

After discussing all these powerful algorithms and techniques we came with a structure that would be efficient and powerful to work along. In our system the data is stored in an encrypted form, with the help of symmetric algorithm and then it is retrieved by the means of enhanced k means clustering algorithm.

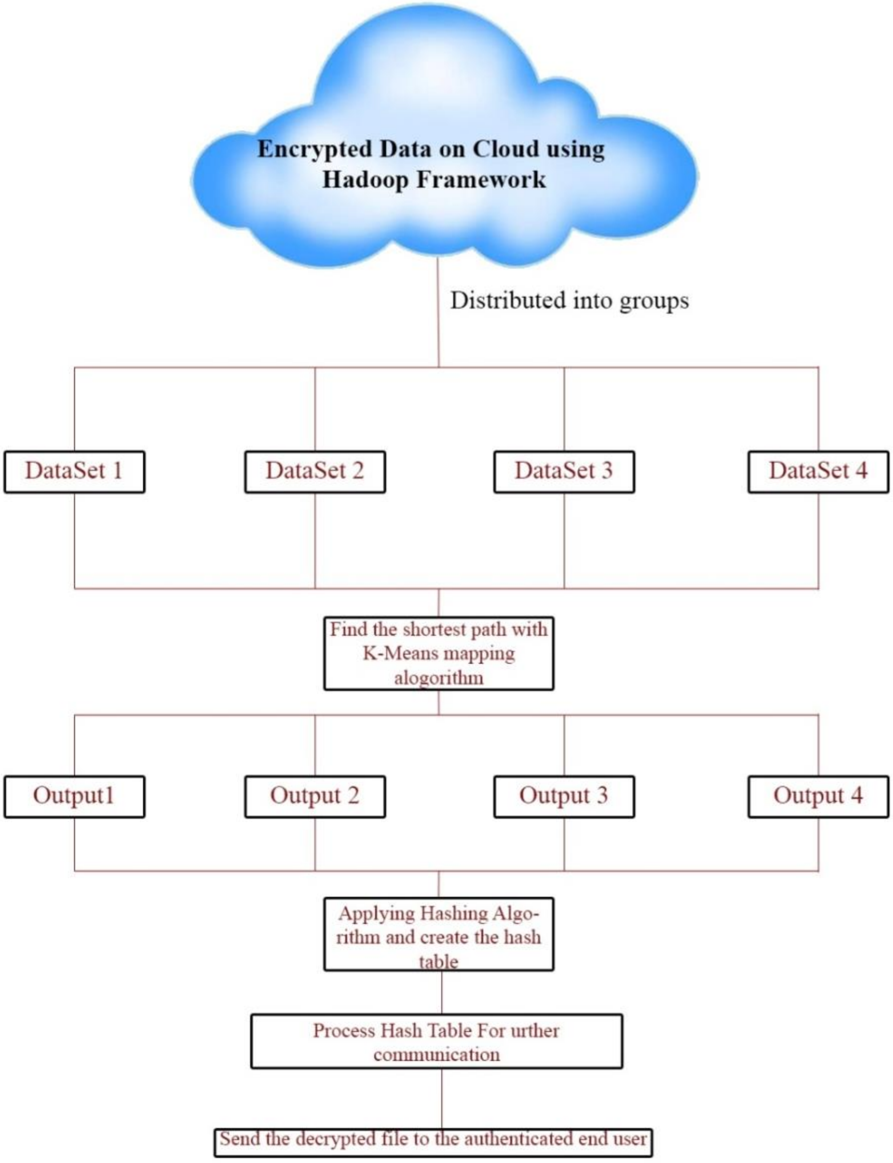

Encrypted Data on Cloud using Hadoop Framework
Distributed into groups
DataSet 1
DataSet 2
DataSet 3
DataSet 4
Find the shortest path with K-Means mapping alogorithm
Output1
Output 2
Output 3
Output 4
Applying Hashing Algo-rithm and create the hash table
Process Hash Table For urther communication
Send the decrypted file to the authenticated end user

Fig. 8. Design of the Proposed Model

Here is the flow chart of our proposed architecture which shows how the data will go and get the required file

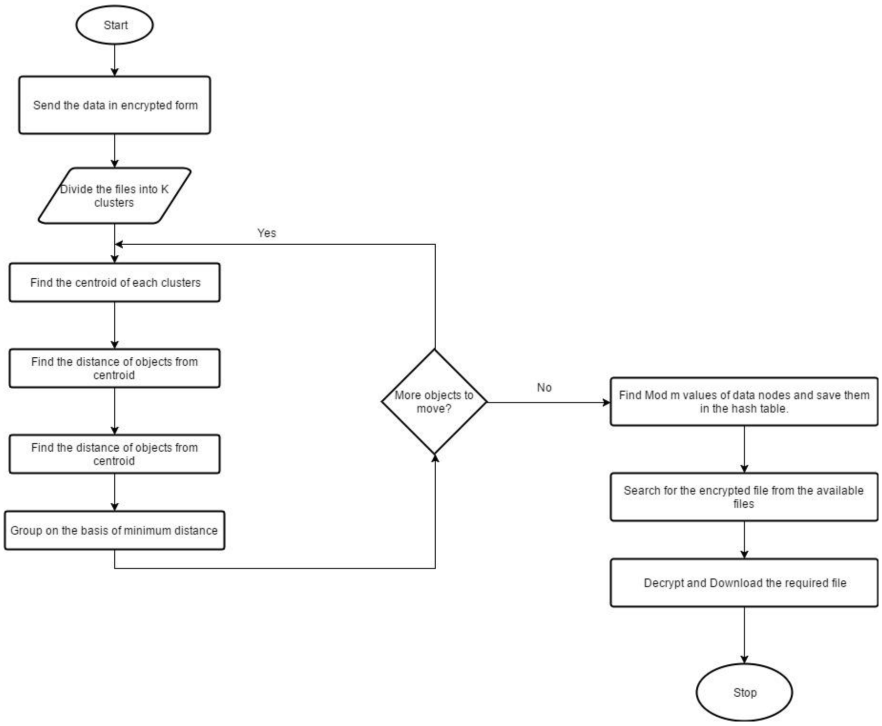

Start
Send the data in encrypted form
Divide the files into K clusters
Yes
Find the centroid of each clusters
Find the distance of objects from centroid
Find the distance of objects from centroid
Group on the basis of minimum distance
More objects to move?
No
Find Mod m values of data nodes and save them in the hash table.
Search for the encrypted file from the available files
Decrypt and Download the required file
Stop

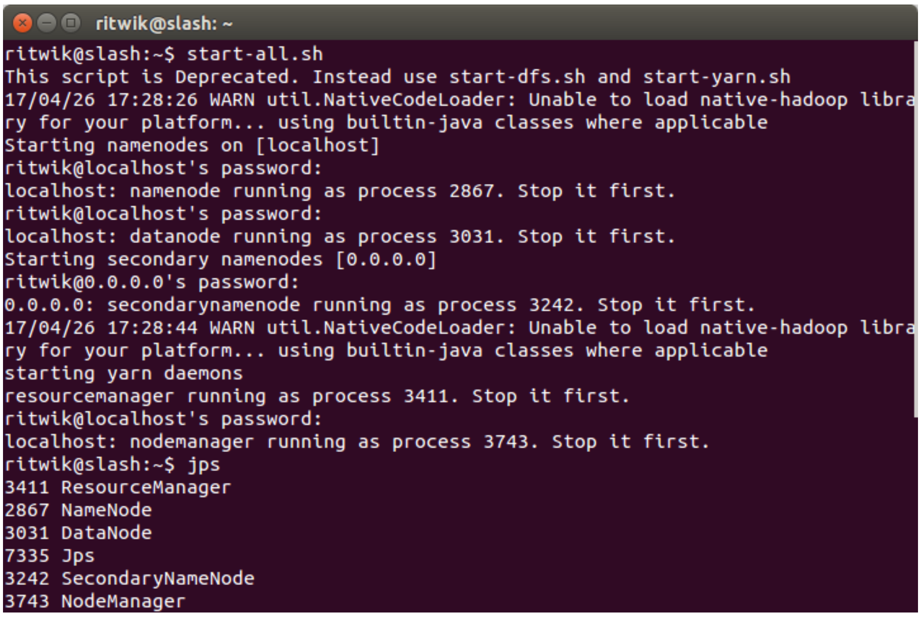
Fig. 9. Flow chart of the proposed design

4. Result Analysis

We used Hadoop platform to access big data and perform the operation, that we had proposed above.

Here is the implementation of the proposed work.

Starting Hadoop and checking whether all nodes are active or not.



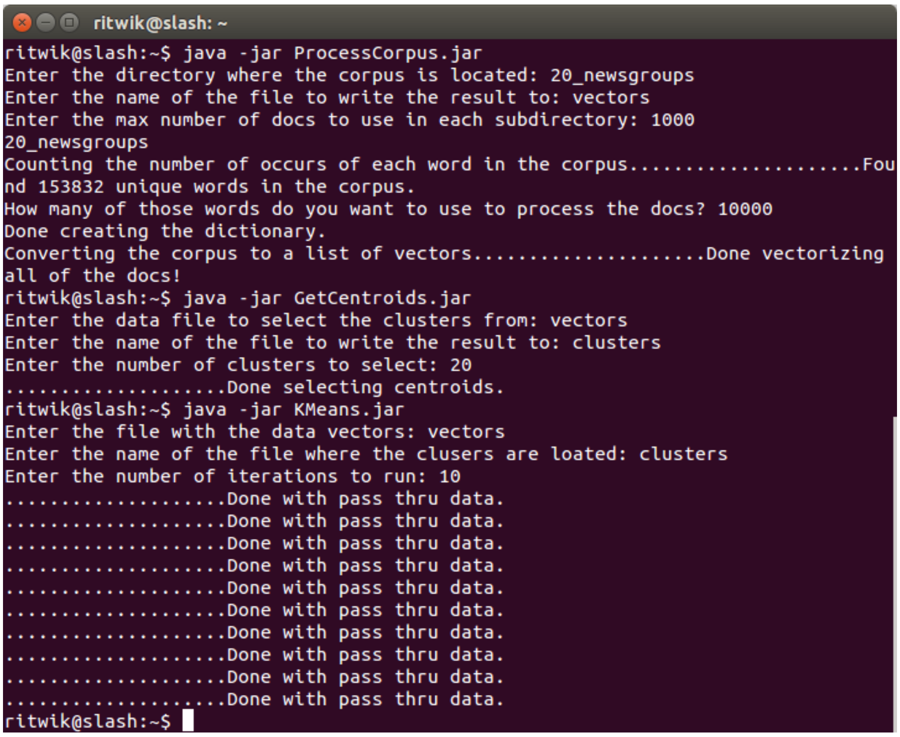
```
ritwik@slash:~$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
17/04/26 17:28:26 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
ritwik@localhost's password:
localhost: namenode running as process 2867. Stop it first.
ritwik@localhost's password:
localhost: datanode running as process 3031. Stop it first.
Starting secondary namenodes [0.0.0.0]
ritwik@0.0.0.0's password:
0.0.0.0: secondarynamenode running as process 3242. Stop it first.
17/04/26 17:28:44 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
starting yarn daemons
resourcemanager running as process 3411. Stop it first.
ritwik@localhost's password:
localhost: nodemanager running as process 3743. Stop it first.
ritwik@slash:~$ jps
3411 ResourceManager
2867 NameNode
3031 DataNode
7335 Jps
3242 SecondaryNameNode
3743 NodeManager
```

Fig. 10. Outputscreen_1

First of all we will start our process by running "ProcessCorpus.jar", which will take a whole of the items present in unstructured data and then convert them into vectors form. This will create a file named vector which contains 20*1000 lines with a dictionary size of 10000 words.

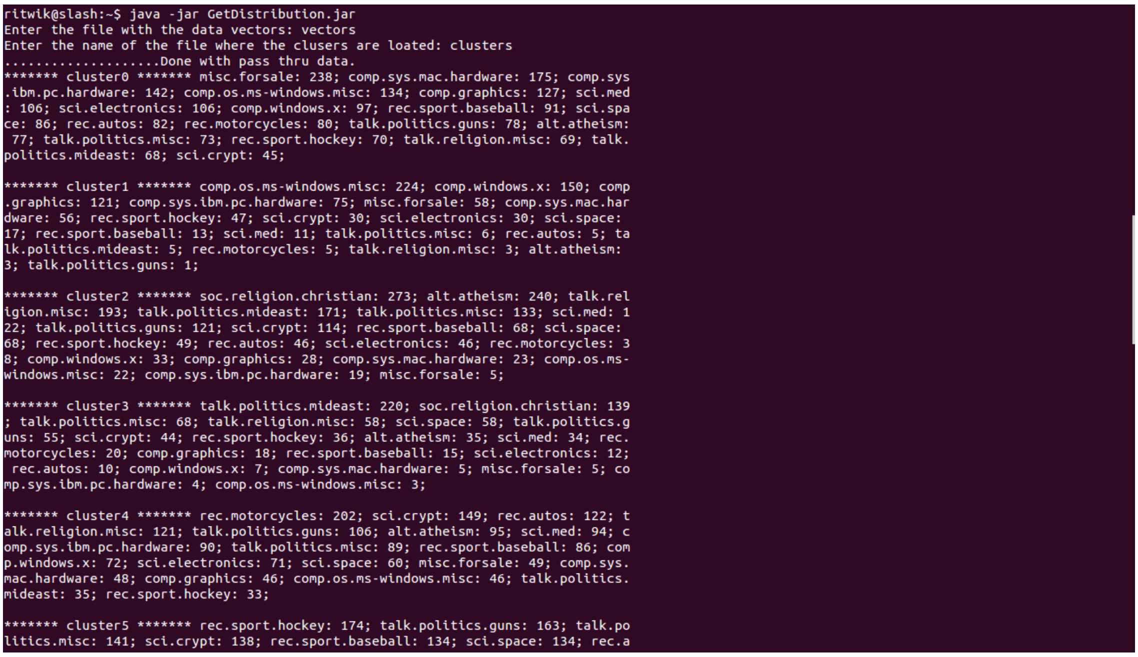
Then we will choose the initial set of centroids with the help of “GetCentroid.jar”. We create a file named “clusters” that consists of 20 lines, each of which describes a centroid of a cluster. The initial set of centroid is chosen randomly from “vectors” file.

Then we will “GetDistribution.jar” which will go off and count how many of each of the 20 types of newsgroups was put into each cluster. If the clustering where ineffective, you would expect an even distribution of newsgroups into clusters. But what you should find it that the clusters tend to group the postings together in meaningful (and interesting!) ways.



```
ritwik@slash:~$ java -jar ProcessCorpus.jar
Enter the directory where the corpus is located: 20_newsgroups
Enter the name of the file to write the result to: vectors
Enter the max number of docs to use in each subdirectory: 1000
20_newsgroups
Counting the number of occurs of each word in the corpus....................Fou
nd 153832 unique words in the corpus.
How many of those words do you want to use to process the docs? 10000
Done creating the dictionary.
Converting the corpus to a list of vectors....................Done vectorizing
all of the docs!
ritwik@slash:~$ java -jar GetCentroids.jar
Enter the data file to select the clusters from: vectors
Enter the name of the file to write the result to: clusters
Enter the number of clusters to select: 20
....................Done selecting centroids.
ritwik@slash:~$ java -jar KMeans.jar
Enter the file with the data vectors: vectors
Enter the name of the file where the clusers are loated: clusters
Enter the number of iterations to run: 10
....................Done with pass thru data.
....................Done with pass thru data.
....................Done with pass thru data.
....................Done with pass thru data.
....................Done with pass thru data.
....................Done with pass thru data.
....................Done with pass thru data.
....................Done with pass thru data.
....................Done with pass thru data.
....................Done with pass thru data.
ritwik@slash:~$ 
```

Fig. 11. Outputscreen_2



```
ritwik@slash:~$ java -jar GetDistribution.jar
Enter the file with the data vectors: vectors
Enter the name of the file where the clusers are loated: clusters
...................Done with pass thru data.
******* cluster0 ******* misc.forsale: 238; comp.sys.mac.hardware: 175; comp.sys
.ibm.pc.hardware: 142; comp.os.ms-windows.misc: 134; comp.graphics: 127; sci.med
: 106; sci.electronics: 106; comp.windows.x: 97; rec.sport.baseball: 91; sci.spa
ce: 86; rec.autos: 82; rec.motorcycles: 80; talk.politics.guns: 78; alt.atheism:
 77; talk.politics.misc: 73; rec.sport.hockey: 70; talk.religion.misc: 69; talk.
politics.mideast: 68; sci.crypt: 45;

******* cluster1 ******* comp.os.ms-windows.misc: 224; comp.windows.x: 150; comp
.graphics: 121; comp.sys.ibm.pc.hardware: 75; misc.forsale: 58; comp.sys.mac.har
dware: 56; rec.sport.hockey: 47; sci.crypt: 30; sci.electronics: 30; sci.space:
17; rec.sport.baseball: 13; sci.med: 11; talk.politics.misc: 6; rec.autos: 5; ta
lk.politics.mideast: 5; rec.motorcycles: 5; talk.religion.misc: 3; alt.atheism:
3; talk.politics.guns: 1;

******* cluster2 ******* soc.religion.christian: 273; alt.atheism: 240; talk.rel
igion.misc: 193; talk.politics.mideast: 171; talk.politics.misc: 133; sci.med: 1
22; talk.politics.guns: 121; sci.crypt: 114; rec.sport.baseball: 68; sci.space:
68; rec.sport.hockey: 49; rec.autos: 46; sci.electronics: 46; rec.motorcycles: 3
8; comp.windows.x: 33; comp.graphics: 28; comp.sys.mac.hardware: 23; comp.os.ms-
windows.misc: 22; comp.sys.ibm.pc.hardware: 19; misc.forsale: 5;

******* cluster3 ******* talk.politics.mideast: 220; soc.religion.christian: 139
; talk.politics.misc: 68; talk.religion.misc: 58; sci.space: 58; talk.politics.g
uns: 55; sci.crypt: 44; rec.sport.hockey: 36; alt.atheism: 35; sci.med: 34; rec.
motorcycles: 20; comp.graphics: 18; rec.sport.baseball: 15; sci.electronics: 12;
 rec.autos: 10; comp.windows.x: 7; comp.sys.mac.hardware: 5; misc.forsale: 5; co
mp.sys.ibm.pc.hardware: 4; comp.os.ms-windows.misc: 3;

******* cluster4 ******* rec.motorcycles: 202; sci.crypt: 149; rec.autos: 122; t
alk.religion.misc: 121; talk.politics.guns: 106; alt.atheism: 95; sci.med: 94; c
omp.sys.ibm.pc.hardware: 90; talk.politics.misc: 89; rec.sport.baseball: 86; com
p.windows.x: 72; sci.electronics: 71; sci.space: 60; misc.forsale: 49; comp.sys.
mac.hardware: 48; comp.graphics: 46; comp.os.ms-windows.misc: 46; talk.politics.
mideast: 35; rec.sport.hockey: 33;

******* cluster5 ******* rec.sport.hockey: 174; talk.politics.guns: 163; talk.po
litics.misc: 141; sci.crypt: 138; rec.sport.baseball: 134; sci.space: 134; rec.a
```

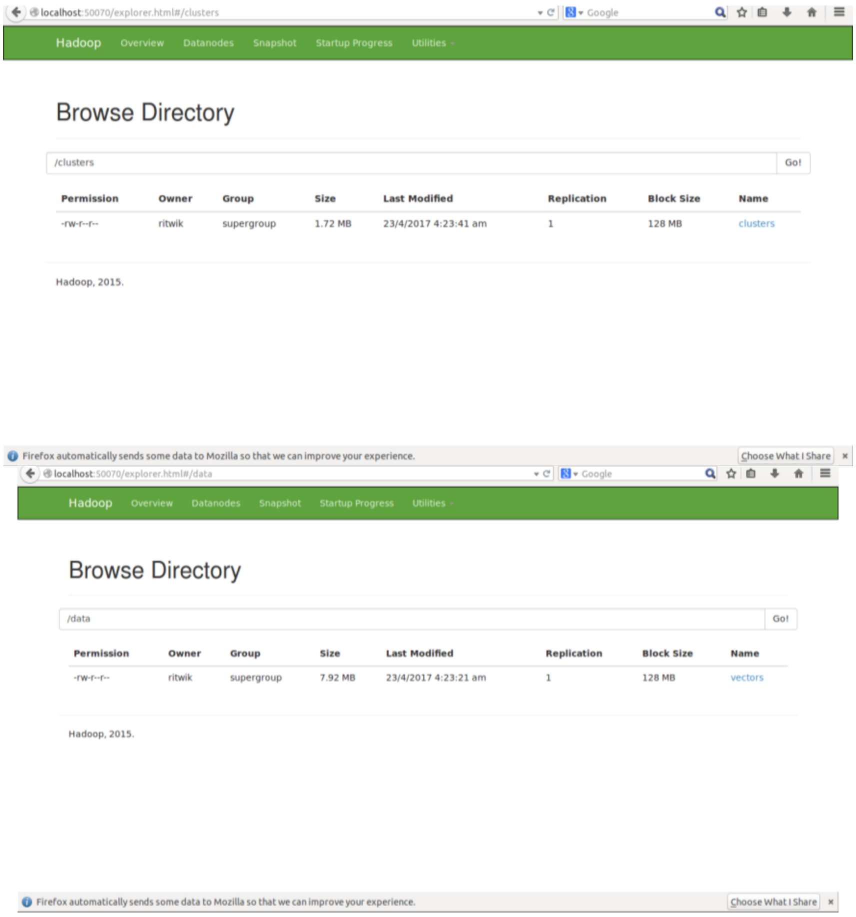
Fig. 12. Outputscreen_3

Now we will send the files “vector” & “clusters” to HDFS by following commands
hdfs dfs -mkdir /data

hdfs dfs -mkdir /clusters

hdfs dfs -copyFromLocal vectors /data

hdfs dfs -copyFromLocal clusters /clusters



Browse Directory

/clusters

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	ritwik	supergroup	1.72 MB	23/4/2017 4:23:41 am	1	128 MB	clusters

Hadoop, 2015.

Firefox automatically sends some data to Mozilla so that we can improve your experience.

Browse Directory

/data

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	ritwik	supergroup	7.92 MB	23/4/2017 4:23:21 am	1	128 MB	vectors

Hadoop, 2015.

Firefox automatically sends some data to Mozilla so that we can improve your experience.

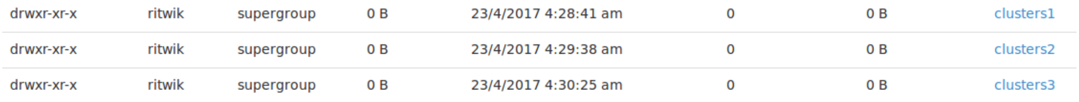
Fig. 13. Outputscreen_4

Turn the master to secure shell and then run the K-Means code over hadoop by typing the following command :

hadoop jar MapRedKMeans.jar KMeans /data /clusters 3

This command will help in running our proposed KMeans algorithm on the clusters and data formed. It pass three parameters, */data*, is the directory present in hadoop file system that contains te data set (20,000 items) ; */custers* is the file in Hadoop file system that contains the initial clusters ; *3(n)*, it tells the number of iterations to be done. *Kmeans* is the file where our proposed algorithm is coded.

Each new file formed will be named as 'clusters1','cluster2'………'clusters_n'



drwxr-xr-x	ritwik	supergroup	0 B	23/4/2017 4:28:41 am	0	0 B	clusters1
drwxr-xr-x	ritwik	supergroup	0 B	23/4/2017 4:29:38 am	0	0 B	clusters2
drwxr-xr-x	ritwik	supergroup	0 B	23/4/2017 4:30:25 am	0	0 B	clusters3

Fig. 14. Outputscreen_5

The following are the screenshots for securing the files. The algorithm used is DES algorithm which converts thee plain text to cipher text and on entering the correct key it will convert the text back to plain text.

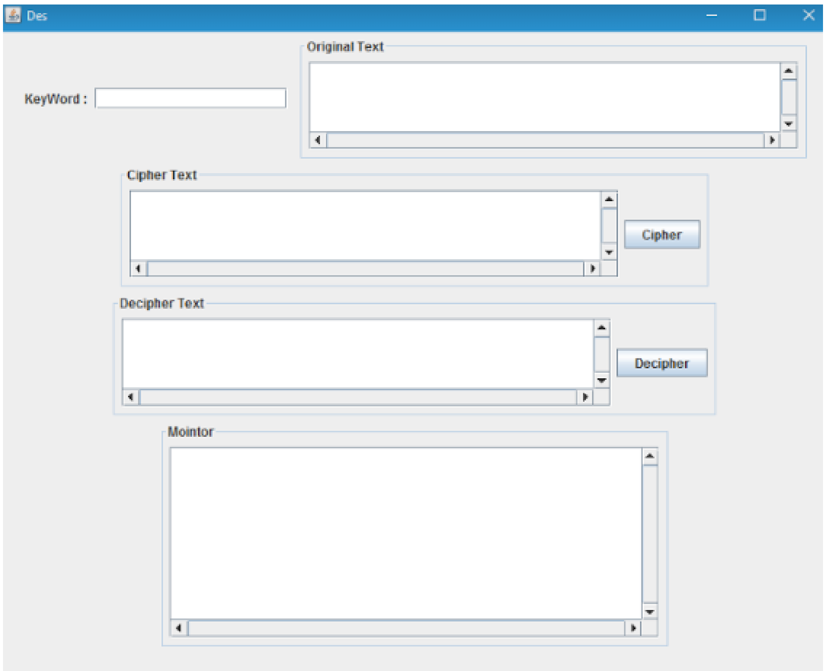
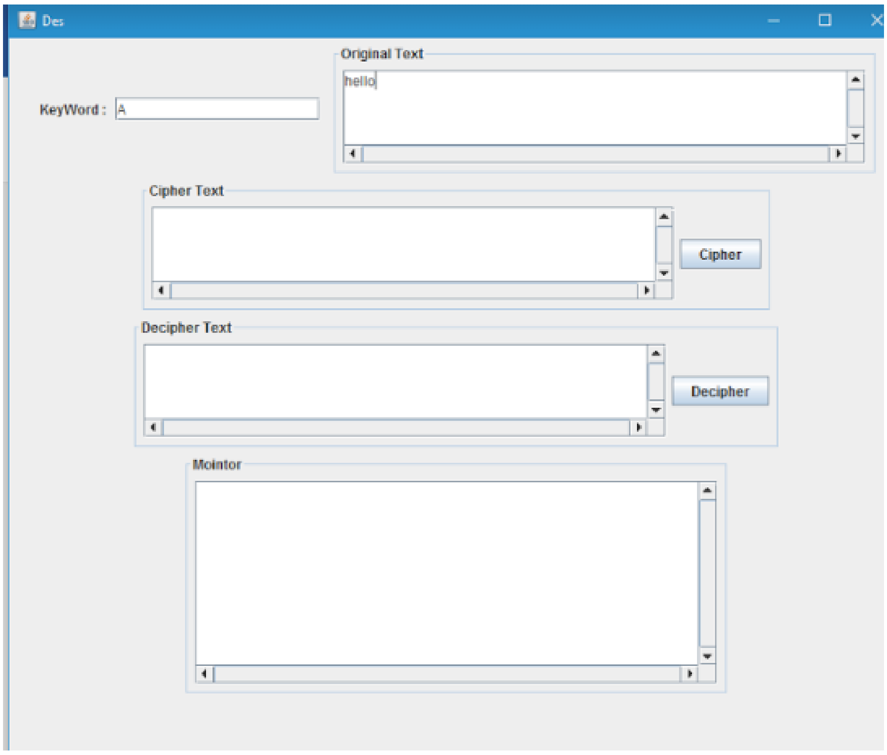


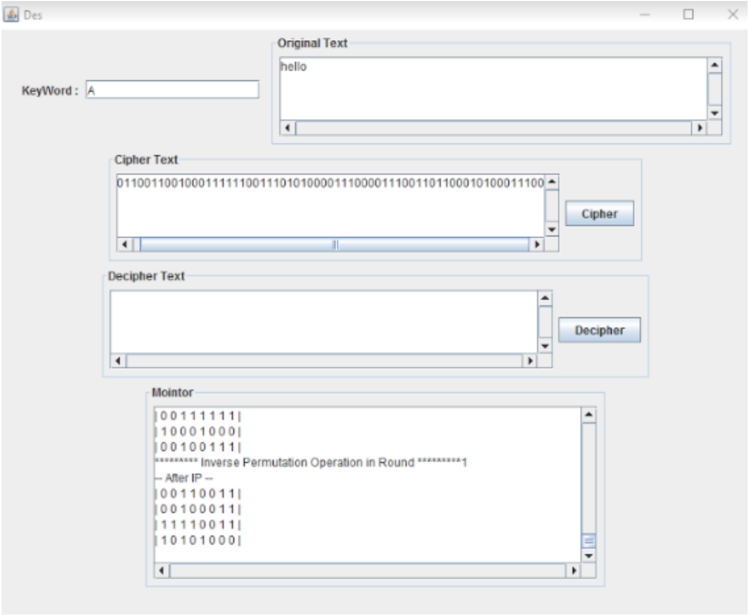
Fig. 15. Outputscreen_6



KeyWord : A

Original Text
hello

Fig. 16. Outputscreen_7



KeyWord : A

Original Text

hello

Cipher Text

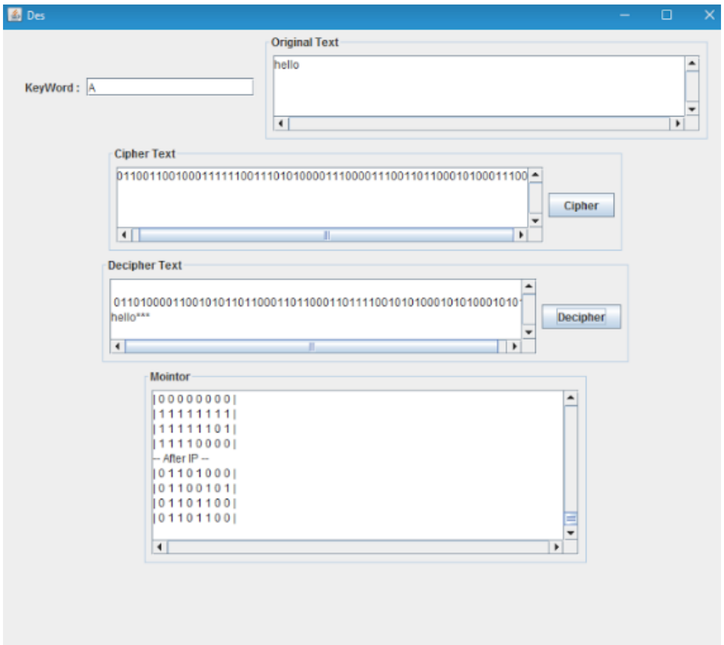
```
01100110010001111110011101010000111000011100110110001010001110O
```

Decipher Text

Mointor

```
|00111111|
|10001000|
|00100111|
********* Inverse Permutation Operation in Round *********1
-- After IP --
|00110011|
|00100011|
|11110011|
|10101000|
```

Fig. 17. Outputscreen_7



KeyWord : A

Original Text

hello

Cipher Text

```
01100110010001111110011101010000111000011100110110001010001110O
```

Decipher Text

```
0110100001100101011011000110110001101111001010100010101000101O
hello***
```

Mointor

```
|00000000|
|11111111|
|11111101|
|11110000|
-- After IP --
|01101000|
|01100101|
|01101100|
|01101100|
```

Fig. 18. Outputscreen_8

5. Result Analysis

After performing the above mentioned architecture, we analyzed the result and got know that by adding hashing we are able to access files faster and with the help of encryption technology the data stored in the HDFS is safe and secure. This architecture can help in maintaining the trust between the user and the system. Over the years, the cloud is facing the problem of security and by these techniques we can give an efficient and safe output in less period of time.

This algorithm will not create load on the system and CPU and smooth accessing is enabled to the user through which he can get the desired and applicable output.

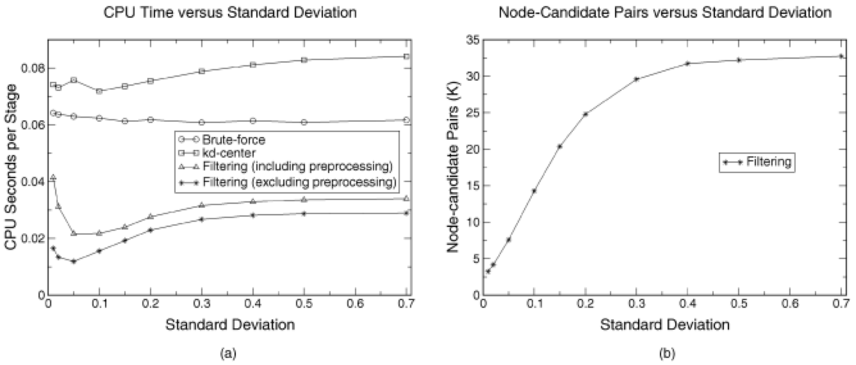


Fig. 19. Average CPU times and node-candidate pairs per stage versus cluster standard deviation for n =10 000, k=20.

References

[1] Hu, Keji, and Wensheng Zhang 2014. "Efficient verification of data encryption on cloud servers." *Privacy, Security and Trust (PST), 2014 Twelfth Annual International Conference on*. IEEE.

[2] Brindha, K., and N. Jeyanthi 2015 "Securing cloud data using visual cryptography." *Innovation Information in Computing Technologies (ICIICT), 2015 International Conference on*. IEEE.

[3] Shynu, P. G., and K. John Singh.2016 "A Comprehensive Survey and Analysis on Access Control Schemes in Cloud Environment." Cybernetics and Information Technologies 16(1), pp.19-38.

[4] Jang, Miyoung, et al. "Clustering-Based Query Result Authentication for Encrypted Databases in Cloud,2014" *High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC, CSS, ICESS), 2014 IEEE Intl Conf on*. IEEE.

[5] Maitri, Punam V., and Aruna Verma.2016 "Secure file storage in cloud computing using hybrid cryptography algorithm." *Wireless Communications, Signal Processing and Networking (WiSPNET), International Conference on*. IEEE.

[6] Zhu, Hongliang, et al.2016 "Based on the character of cloud storage string encryption and cipher text retrieval of string research." *Cloud Computing and Intelligence Systems (CCIS), 2016 4th International Conference on*. IEEE.

[7] Esteves, Rui Maximo, Rui Pais, and Chunming Rong,2011 "K-means clustering in the cloud--a Mahout test." *Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference on*. IEEE.

[8] Kim, SungYe, et al. 2015 "Power efficient mapreduce workload acceleration using integrated-gpu." *Big Data Computing Service and Applications (BigDataService), 2015 IEEE First International Conference on*. IEEE.

[9] Gugnani, Shashank, and Tamas Kiss.2015 "Extending Scientific Workflow Systems to Support MapReduce Based Applications in the Cloud." *Science Gateways (IWSG), 2015 7th International Workshp on*. IEEE.

[10] Saxena, Ankur, et al.2016"Implementation of cloud computing and big data with Java based web application." *Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on*. IEEE.

[11] Yetis, Yunus, et al.2016 "Application of Big Data Analytics via Cloud Computing." *World Automation Congress (WAC),* IEEE.

[12] Buyya, Rajkumar, et al.2015 "Big Data Analytics-Enhanced Cloud Computing: Challenges, Architectural Elements, and Future Directions." *Parallel and Distributed Systems (ICPADS), 2015 IEEE 21st International Conference on*. IEEE.

[13] Lu, Huang, Chen Hai-Shan, and Hu Ting-Ting 2012 "Research on hadoop cloud computing model and its applications." *Networking and Distributed Computing (ICNDC), 2012 Third International Conference on*. IEEE.

[14] Kanungo, Tapas, et al.2002 "An efficient k-means clustering algorithm: Analysis and implementation." *IEEE transactions on pattern analysis and machine intelligence* (24) 7,881-892.

[15] Adnan, Muhammad, et al.2014 "Minimizing big data problems using cloud computing based on Hadoop architecture." *High-capacity Optical Networks and Emerging/Enabling Technologies (HONET), 2014 11th Annual*. IEEE.

[16] Kuzu, Mehmet, Mohammad Saiful Islam, and Murat Kantarcioglu.2012 "Efficient similarity search over encrypted data." *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*. IEEE.

[17] Iqjot singh, Prerna dwivedi, Taru gupta, Shynu P.G.2016 “An enhanced k-means clustering algorithm for big data in cloud.” */International Journal of Pharmacy & Technology,* (8)4.