Comparison of Keyword based Clustering of Web Documents by using OPENSTACK 4J and by Traditional Method

Shiza Anand
P.G. student, Department of Computer Science & Engineering, Meerut Institute of Engineering & Technology, Meerut, Uttar Pradesh, India
13 Rama Kutir, Tilak Road
Meerut, 250001

Pradeep Pant
Head Of, Department, Department of Computer Science & Engineering,Meerut Institute of Engineering & Technology, Meerut, Uttar Pradesh, India

Mukesh Rawat, PhD
Associate Professor, Department of Computer Science &Engineering,Meerut Institute of Engineering & Technology, Meerut, Uttar Pradesh, India

ABSTRACT
As the number of hypertext documents are increasing continuously day by day on world wide web. Therefore, clustering methods will be required to bind documents into the clusters (repositories) according to the similarity lying between the documents. Various clustering methods exist such as: Hierarchical Based, K-means, Fuzzy Logic Based, Centroid Based etc. These keyword based clustering methods takes much more amount of time for creating containers and putting documents in their respective containers. These traditional methods use File Handling techniques of different programming languages for creating repositories and transferring web documents into these containers. In contrast, openstack4j SDK is a new technique for creating containers and shifting web documents into these containers according to the similarity in much more less amount of time as compared to the traditional methods. Another benefit of this technique is that this SDK understands and reads all types of files such as jpg, html, pdf, doc etc. This paper compares the time required for clustering of documents by using openstack4j and by traditional methods and suggests various search engines to

adopt this technique for clustering so that they give result to the user queries in less amount of time.

Keywords
Clustering, openstack4j, K-Means, centroid based, document-matching

1. INTRODUCTION
There are many limitations of the the traditional clustering techniques. Therefore, OpenStack4j is introduced which overcomes all the weaknesses of the traditional clustering techniques. These traditional clustering techniques are very time consuming. Java is the mainly used programming language these days but it cannot read html, jpg, pdf, format files directly. Thus, conversion need to take place for clustering to occur. This process takes a lot of time and as a result these methods of clustering are slow. On the other hand, OpenStack4j overcomes this drawback as it contains some built-in functions that are capable of reading and understanding files of all formats and transfers them to their specific containers in a very less amount of time.OpenStack is a set of software tools and techniques for making and managing cloud computing platforms for publicly available as well as private clouds. Supported by few of the biggest companies in software development, as well as most of individual community members, many suppose and think that OpenStack is the future of cloud computing. OpenStack is managed by the OpenStack Foundation, a non-profit that for looks oversees both development and community-building around the project.OpenStack is a cloud computing operating system that controls big pools of computation, storage, and networking related resources throughout a database(datacentre),all managed through a dashboard that gives administrators full control while providing their users to access resources through a web interface.The OpenStack community collaborates around a six-month, time-based release cycle with frequent development milestones. During the planning phase of each product release, the community collects for an OpenStack Design Summit to facilitate programmer (developer) working periods(sessions) and to combine & assemble plans.

1.1 What is Openstack4j?
Anyone may be an OpenStack user and might not even be aware of it. As more and more companies begin to adopt OpenStack as a part of their cloud computing toolkit, the whole universe of apps running on an OpenStack backend is always expanding.OpenStack4j is an open source library that provides the facility of managing an OpenStack deployment process. It is a fluent based Application Programming Interface giving complete and full control over different services of OpenStack.

1.1.1 Why to use OpenStack4j?
OpenStack is a huge and a large system to maintain and manage. It is made it easy by providing a simplistic fluent API and intelligent error handling process.

Clustering
Clustering or cluster analysis is the task of collecting and grouping a set of objects in such a manner that objects in the same group (called a cluster) are similar (in some sense or another) to each other than to those in other groups (clusters). This is a major task of exploratory data mining, and a more common technique for statistical data analysis, used in many fields, pattern recognition, including machine learning image analysis, bioinformatics, information retrieval, data compression and computer graphics.

K-Means Clustering

It is a partitioning technique which finds mutual exclusive clusters of spherical in shape. It produces (generates) a specific number of disjoint, flat (non-hierarchical) clusters. Statistical method can be used to group (cluster) to assign rank values to the cluster categorical data. Categorical data have been converted into numeric by assigning rank value [115]. K-Means algorithm arranges (organizes) objects into k – partitions where each partition represents a cluster (Group). We initially start with initial set of means and classify cases based on their distances to their centres. Then, we compute the cluster means again, using the cases that are assigned to the clusters, then, we regroup and reclassify all cases based on the new set of means. We keep repeating this step till cluster means don't change between all successive steps. Lastly, we calculate the means of cluster once again and assign the cases to their permanent groups or clusters.

1.2 Document Clustering

Document clustering is the way of collecting similar documents into bins, where similarity is some function on a document.Document clustering relates to the use of descriptors and descriptor extraction. Descriptors are sets of words that describe the contents within the cluster. Document clustering is considered to be a centralized process. Examples of document clustering include web document clustering for search users.

2. RELATED WORK

"Web Document Clustering and Ranking using Tf-Idf based Apriori Approach [01]","The dynamic web has increased exponentially in the past few years with more than thousands of files and documents related to a subject available to the user now. Most of the web documents are not structured and not in an organized manner and therefore user is facing more difficulty in finding necessary and relevant documents. A more useful and efficient mechanism is combining clustering (grouping) with ranking, where clustering can group the similar documents at one place and ranking can be applied to each cluster for viewing the top documents at the initial beginning. Besides the particular clustering algorithm, the different term weighting functions applied to the selected features to represent that web document is a main aspect in clustering. Keeping this task in mind, we propose a new mechanism called Tf-Idf based Apriori for clustering the web based documents. Then we rank the documents in each cluster using Tf-Idf and similarity factor of documents based on the user queries. This approach will help the user to get all the relevant documents at one place and can restrict the search to few top documents of choice. For experimental purpose, we have taken the Classic3 and Classic4 datasets of Cornell University having more than 10,000 documents and using Gensim toolkit to carry out the work. We have compared our approach with traditional apriori algorithm and found that our approach is giving much more better results for higher minimum support. Our ranking mechanism is also giving a good F-measure of 78%."

"Clustering Web Documents using Hierarchical Method for Efficient Cluster Formation [02]", "Document clustering aims to group, in an unsupervised way, a given document set into clusters such that documents within each cluster are similar between each other than those in different clusters. It is an enabling technique for a wide range of information retrieval tasks such as efficient organization, browsing and summarization of large volumes of text documents. Cluster analysis aims to plan and organize a collection of patterns into clusters (groups) based on similarity. Clustering has its existence (root) in many fields, like mathematics, computer science, statistics, economics and biology. In different application domains, a large variety of clustering techniques have been developed, depending on the methods used to represent data, the techniques for grouping data objects into clusters and the measures of similarity between data objects. The first part is a document index model, the Document Index Graph, which allows for incremental construction of the index of the document set with an emphasis on efficiency, rather than depending (relying) on single-term indexes only. It provides reliable (efficient) phrase matching that is used to judge the similarity between documents. This model is flexible in that it could revert to a compact representation of the vector space model if we choose not to index phrases. The second part is an incremental document clustering algorithm based on maximizing the closeness (tightness) of clusters by carefully watching the pair-wise document similarity distribution inside clusters. Both the phases are based upon two algorithmic models called Gaussian Mixture Model and Expectation Maximization. The combination of these two components creates an underlying model for robust and accurate document similarity calculation that leads to much improved results in Web document clustering over traditional methods."

"Web Search Result Clustering - A REVIEW [03] ", "The ever-increasing information on the web with its heterogeneity and dynamism needs an information retrieval system which serves searcher's ambiguous, ill-formed, short queries with relevant result in a precise way. Web search result clustering has been emerged as a method which overcomes these drawbacks of conventional information retrieval (IR) systems. It is the clustering of results returned by the search engines into meaningful, thematic groups. This paper gives a succinct overview and categorizes various techniques that have been used in clustering of web search results."

"Matching Similarity for Keyword-based Clustering [04]","Semantic clustering of objects such as documents, web sites and movies based on their keywords is a challenging problem. This requires a similarity measure between two sets of keywords. We present a new measure based on matching the words of two groups assuming that a similarity measure between two individual words is available. The proposed matching similarity measure avoids the problems of traditional measures including minimum, average and maximum similarities. We demonstrate that it provides better clustering than other measures in location-based service application."

"Learning to Cluster Web Search Results [05]", "Organizing Web search results into clusters facilitates users' quick browsing through search results. Traditional clustering techniques are inadequate because they don't generate clusters with highly readable names. In this paper, we re-formalize the clustering problem as a salient phrase ranking problem. Given a query and the ranked list of documents (typically a list of titles and snippets) returned by a certain Web search engine, our method first extracts and ranks salient phrases as candidate cluster names, based on a regression model learned from human labelled training data. The documents are assigned to relevant salient phrases to form candidate clusters, and then the final clusters are created and generated by merging these candidate clusters. Experimental results verify our method's effectiveness and feasibility."

2.1 Algorithm for Cluster Generation using OpenStack4j

2.1.1 Used these credentials from Bluemix by going to the Object Storage service, and clicking on Service Credentials:

```
private static final String USERNAME = "gfc27miet@gmail.com";

private static final String PASSWORD = "ayushmangfc27";

private static final String DOMAIN_ID = "localhost";

private static final String PROJECT_ID = "Websearch1";
```

1. Object Storage uses a WSGI model to provide for a middleware capability that not only provides general extensibility but is also used for authentication of end-point clients. The authentication provider defines what roles and user types exist. Some use traditional user name and password credentials while others may leverage API key tokens or even client-side x.509 certificates. Custom providers can be integrated in using custom middleware. Object Storage comes with two authentication middleware modules by default, either of which can be used as sample code for developing a custom authentication middleware.

Code:

```
ObjectStorageService
    authenticateAndGetObjectStorageService() {

String OBJECT_STORAGE_AUTH_URL = "https://identity.open.softlayer.com/v3";

Identifier domainIdentifier = Identifier.byName(DOMAIN_ID); }
```

2.1.2 Overriding Methods

```
protected void doDelete(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

ObjectStorageServiceobjectStorage=authenticateAndGetObjectStorageService();

System.out.println("Deleting file from ObjectStorage...");

String containerName = request.getParameter("container");

String fileName = request.getParameter("file");

if(containerName == null || fileName == null){ //No file was specified to be found, or container name is missing

response.sendError(HttpServletResponse.SC_NOT_FOUND);

System.out.println("File not found.");

return;

    }

    ActionResponsedeleteResponse = objectStorage.objects().delete(containerName,fileName);

    if(!deleteResponse.isSuccess()){

    response.sendError(deleteResponse.getCode());

    System.out.println("Delete failed: " + deleteResponse.getFault());

    return;

    }

    else{

response.setStatus(HttpServletResponse.SC_OK);

    }

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

    ObjectStorageService objectStorage = authenticateAndGetObjectStorageService();

    System.out.println("Storing file in ObjectStorage...");

    String containerName = request.getParameter("container");

    String fileName = request.getParameter("file");

    if(containerName == null || fileName == null){ //No file was specified to be found, or container name is missing

    response.sendError(HttpServletResponse.SC_NOT_FOUND);

    System.out.println("File not found.");

    return;

    }final InputStream fileStream = request.getInputStream();

    Payload<InputStream> payload = new PayloadClass(fileStream);

    objectStorage.objects().put(containerName, fileName, payload); }
```

3. PROPOSED SYSTEM

As the quantity of hypertext documents are increasing continuously day by day on internet. In this way, clustering techniques will be required to tie reports into the groups (archives) as per the comparability lying between the documents. In this paper, we proposed an optimized virtual machine migration mechanism which based on the Open Stack cloud platform. openstack4j software development kit is a new technique for creating containers and shifting web documents into these containers according to the similarity in much more less amount of time as compared to the traditional methods. We use OpenStack4j for clustering of document while searching for documents to improve the accuracy of result.

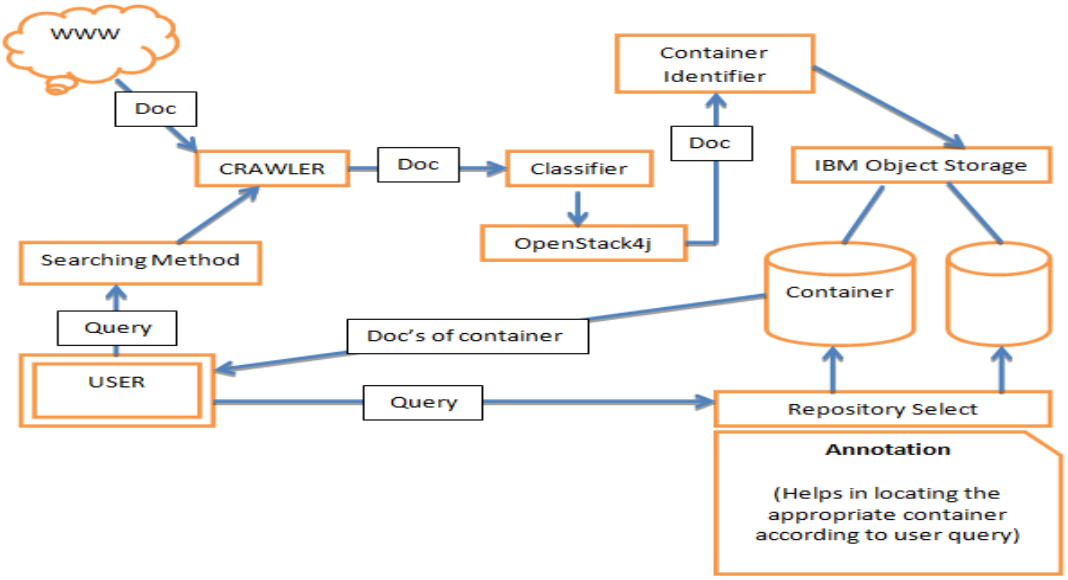


Fig 1. General Architecture

Documents are fetched from the www and submitted to the crawler. A crawler is a program that fetches pages and information from the www. Crawler in turn forwards the document to the classifier. The work of the classifier is to categorize and clusters objects or instances having common behaviors and structural attributes. Classifier searches the documents with the help of different searching methods and algorithms. This paper is least concerned about the searching method applied and its selection. Classifier will work with OpenStack4j to forward the fetched documents. IBM object storage contains containers in which different data is placed based on similarity among them. Therefore, particular Container is selected by the Container Identifier based on matching of documents. IBM Object Storage is like a Warehouse. It is a service introduced by IBM for storing objects. The IBM Object Storage is linked with the containers in which all the information and files reside. And it has a Bi-directional relation with the Container Identifier. The user can generate Query for the documents required by him. And Repository Select will provide the user the required information by choosing files from the specific container. Thus, The chain continues and the information retrieval is done in a much less time as compared to the traditional methods because of OpenStack4j. An Annotation is an explanation attached to the text, image and other different types of data and helps in locating the appropriate container according to user query.

4. RESULT ANALYSIS

Table1:Table showing the comparison of time taken to transfer files and storage in cluster using OpenStack4j and other traditional techniques.

No. of Doc's	Clusters Created	Average Cluster Size	Cluster creation time taken (millisec)			Time taken to transfer documents to their specific cluster(milisec)		
			By K-means	By Agglomerative	By OpenStack4j	By K-means	By Agglomerative	By OpenStack4j
20	2	9	2	2.7	1.8	2	2.9	1.7
30	4	15	4	4.5	3.2	3	3.8	2.5
40	5	10	7	6.7	5.5	4	4.7	3.7
50	8	18	10	10.9	8.9	7.1	7.6	6.6
60	11	20	11	11.4	10.3	10	10.8	9.8
70	15	8	14	15	13.8	11	11.7	10.5
80	14	10	17	17.4	16.6	13	13.6	12.7
90	17	12	18	18.7	17.8	14	14.5	13

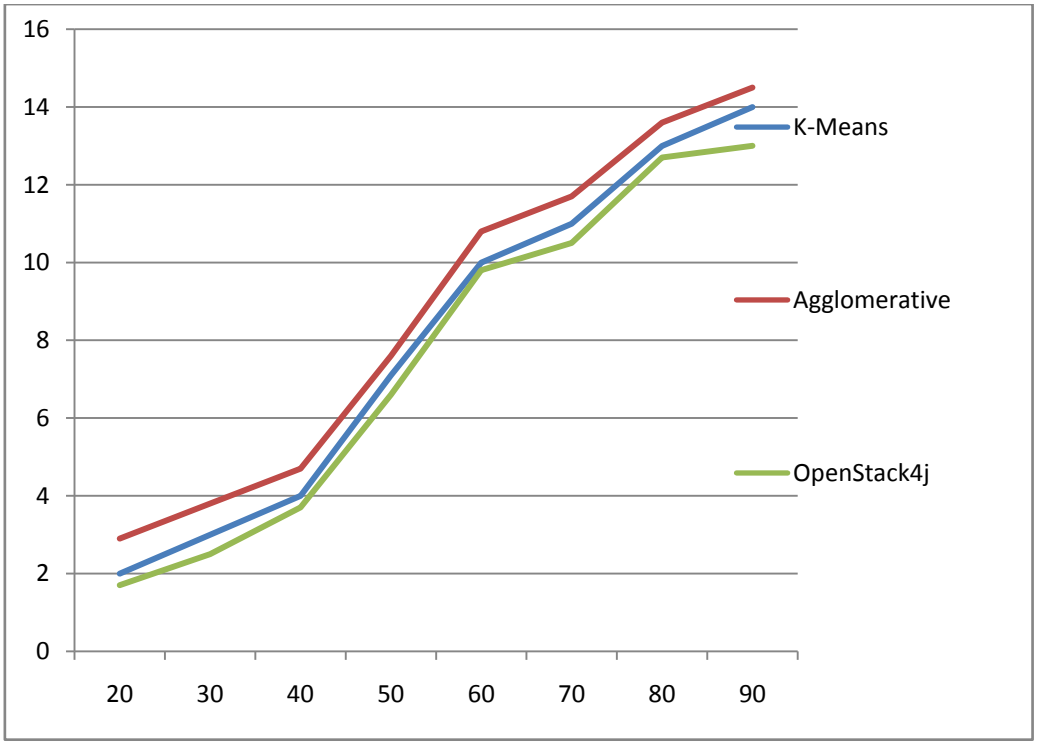
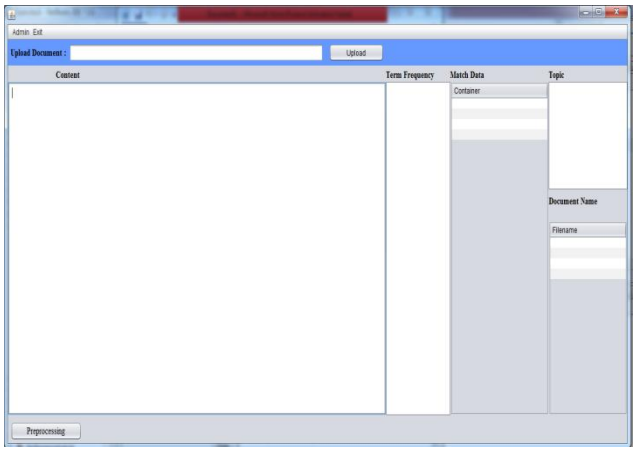


Fig 2: Graph showing the comparison of time taken to transfer files and storage in cluster using OpenStack4j and other traditional techniques

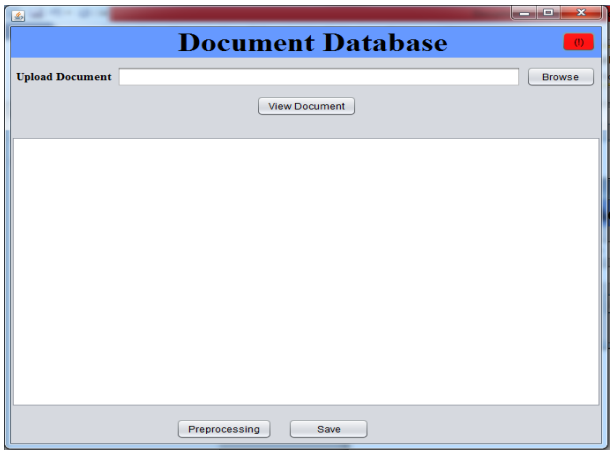
5. OUTPUT SCREENS

1. The first screen of the Project

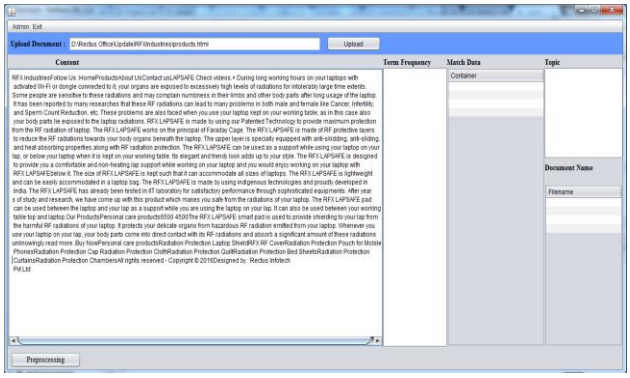


2. This shows the Admin Login in order to store the HTML documents in the Database

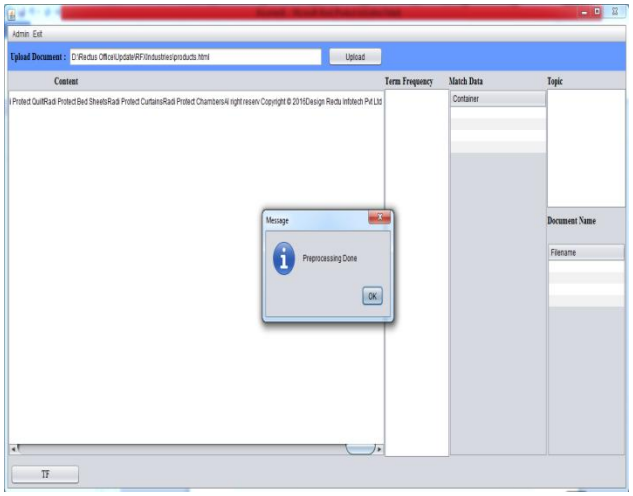
3. Upload the documents in the Document Database. We can browse and upload documents from the computer system and save them.



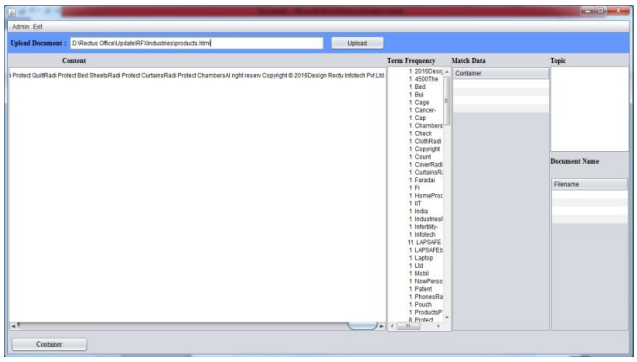
4. Now the Preprocessing of the documents takes place, thereby, removing all the images etc. Only the text content is available for further processing.



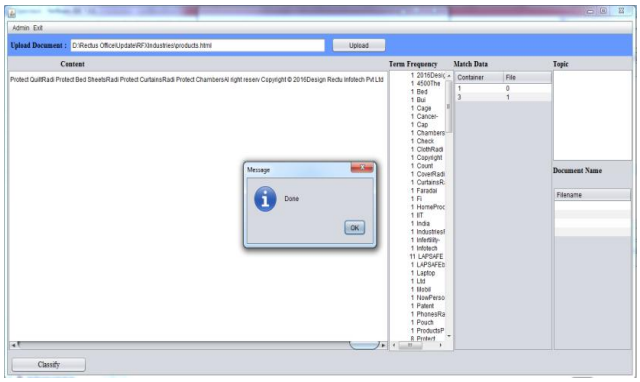
5. Preprocessing of the document has been performed.

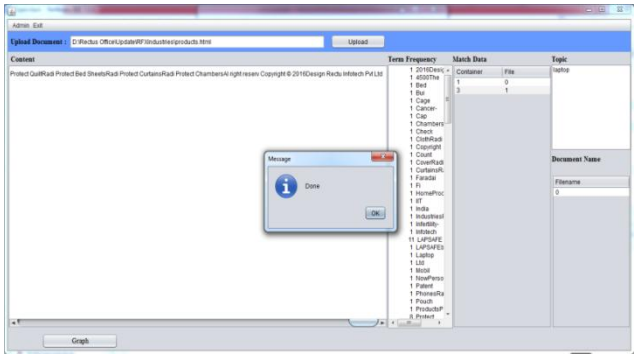


6. After preprocessing, Term Frequency of words are calculated, all the stop words and stemming words are removed.

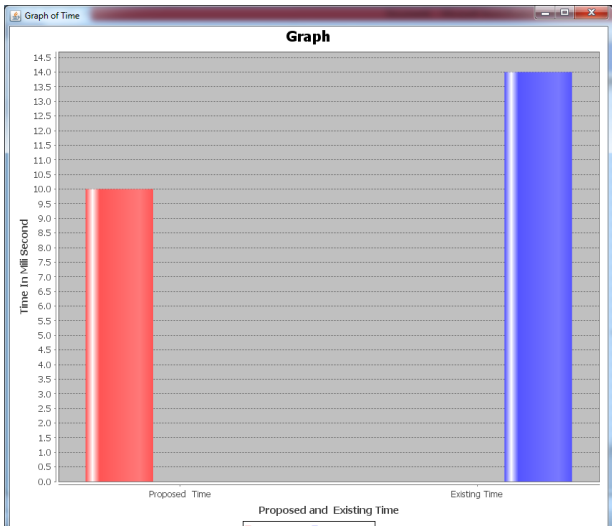


7. Now the files will be classified and will be put in the containers that were created using OpenStack4j





8. A graph is plotted to show the time taken by Traditional Clustering Methods and OpenStack4j



6. CONCLUSION

The paper concludes with the fact that clustering and transfer of web -based data is much more faster and easier when done through Object based storage using OpenStack4j as compared to the existing methods of clustering like Agglomerative based clustering, clustering by K-Means, Centroid based clustering and various other traditional methods of clustering. The above shown graph and the table of comparison on the basis of time taken by each clustering technique to transfer the documents to their specific containers, clearly depicts that OpenStack4j takes the least amount of time. The reason behind is that it does not need to convert documents of any format, it already has pre-defined and built in functions that can accept the documents in their existing formats, and this feature is not found in the traditional clustering techniques, therefore, they need to convert first and then perform clustering, which consumes a lot of extra time. The paper deals with the way that clustering and exchange of web - based information is a great deal more quicker and simpler when done through Object based capacity utilizing OpenStack4j when contrasted with the current techniques for bunching like Agglomerative based grouping, Clustering by K-Means, Centroid based grouping and different other customary strategies for grouping. The above demonstrated diagram and the table of correlation on the premise of time taken by every clustering technique to exchange the archives to their particular holders, plainly portrays that OpenStack4j takes minimal measure of time. The purpose for is that it doesn't have to change over records of any configuration, it as of now has pre-characterized and worked in capacities that can acknowledge the archives in their current arrangements, which the element that is not found in the customary bunching strategies, subsequently, they have to change over first and afterward perform grouping, which devours a great deal of additional time.

7. ACKNOWLEDGEMENTS

The work presented in this paper would not have been possible without my close association with many people. I take this opportunity to extend my sincere gratitude and appreciation to all those who made this thesis possible.First and foremost, I would like to extend my sincere gratitude to my research guide, Prof. Pradeep Pant for continuing guidance, constructive comments, motivation,

andencouragement throughout. During our course of interaction, I have learnt extensively from him, including how to approach a problem by systematic thinking, data-driven decision making and exploiting serendipity. His invaluable suggestions and precious ideas have helped me to walk through various stages of my dissertation, while his passion and extraordinary dedication to work have always inspired me and encouraged me to work harder.I owe my deepest gratitude towards Dr.Mukesh Rawat for his selfless support and encouragement. This thesis is indeed a realization of his dream. I am greatly indebted to my parents.Their infallible love and support has always been my strength.Finally, my greatest regards to the Almighty for bestowing upon me the courage to face the complexities of life and complete my dissertation successfully.

8. REFERENCES

[1] Kevin Jackson and Cody Bunch, "OpenStack Cloud Computing Cookbook", Second Edition, 2001, Page No.400

[2] Tom Fifield, Diane Fleming & Joe Topjian, "OpenStack Operations Guide" by O'Reilly Publications, @itarchitectkev, openstack.prov12n.com

[3] John Rhoton,Jan De Clercq, Franz Novak, "OpenStack Cloud Computing", Architecture Guide, 2014 Edition, Recursive Press Publications, March11, 2014

[4] Dan Radez, "OpenStack Essentials",PACKT Publishing, 2012, www,PacktPub.com

[5] Charu C.Aggarwal, "Data Clustering Algorithms & Applications", January1, 2013.

[6] Junjie Wu, "Adances In K-Means Clustering", January1, 2012.

[7] Sewell, Grandville, and P.J. Rousseau, "Finding groups in data: An introduction to cluster analysis", 1990,2005, Page no,223

[8] Ipeirotis, P., Gravano, L. & Mehran, S. (2001), 'Probe, count, and classify: categorizing web databases', ACM SIGMOD 30(2), 67 – 78.

[9] Omar Khedher, " Mastering OpenStack", Packt Publishing, www,PacktPub.com

[10] IBM-Object Storage, www.ibm.com/object-storage/ , Date: 18.05.2016, Time: 11.30am

[11 Object Storage- IBM Bluemix, https://console.ng.bluemix.net/object-storage/ Date:29.05.2016, Time: 10.00am.

[12] I.Ceema, M.Kavitha, G.Renukadevi, G.sripriya, S. RajeshKumar, "Clustering Web Documents using Hierarchical Method for Efficient Cluster Formation", International Journal of Advanced Research in Computer Science and Electronics Engineering, Volume 1, Issue 5, November 2012, ISSN: 2277 – 9043,

[13] Wei Xu, Xin Liu, Yihong Gong, "Document Clustering Based On Non-negative Matrix Factorization",

[03] Rajendra Kumar Roul, Omanwar Rohit Devanand, S. K. Sahay, "Web Document Clustering and Ranking using Tf-Idf based Apriori Approach",

[14] Oren Zamir and Oren Etzioni,"Web Document Clustering: A Feasibility Demonstration", SIGIR'98, Melbourne, Australia 1998 ACM 1-58113-015-5 8/98,

[15] Hua-Jun Zeng, Qi-Cai He, Zheng Chen, Wei-Ying Ma, Jinwen Ma, "Learning to Cluster Web Search Results", SIGIR'04, July 25–29, 2004, Sheffield, South Yorkshire, UK., ACM 1-58113-881-4/04/0007,

[16] Yongzheng Zhang Evangelos Milios Nur Zincir-Heywood, "A Comparison of Word- and Term-based Methods for Automatic Web Site Summarization", WWW2004, May 17–22, 2004, New York, NY USA, ACM,

[17] Mohammad Rezaei and Pasi Fränti, "Matching Similarity for Keyword-based Clustering", adfa, p. 1, 2011. © Springer-Verlag Berlin Heidelberg 2011.