Comparison Of Keyword Based Clustering Of Web Documents By Using Openstack 4j And By Traditional Method

Shiza Anand, Dr. Mukesh Rawat

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 querries in less amount of time.

Keywords: clustering, openstack4j, K-Means, centroid based

1. Introduction

K-means Based Clustering- It is a method of vector quantization, that is popular for cluster analysis in data mining. The main aim of k-means clustering is to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a blueprint of the cluster. This results in partitioning of data space into Voronoi cells. Agglomerative Based Clustering- This is a type of Hierarchical Based clustering in which clusters are analysed on the basis of hierarchy. It is a bottom up approach each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy. The complexity of agglomerative clustering is **O(n3),** which makes them too slow for large data sets. Centroid Based Clustering- This is the type of clustering in which the distance measure between the objects of the data set considered. The basic aspect of distance measure in general is derived using one of Euclidian, or Manhattan distance measuring mechanism OpenStack4j- OpenStack4j is an open source library that helps you manage an OpenStack deployment. It is a fluent based API giving you full control over the various OpenStack services. OpenStack is a large system to manage. We have made it easy by providing a simplistic fluent API and intelligent error handling.

- SHIZA ANAND, Dr. MUKESH RAWAT
- Department of Computer Engineering, Meerut Institute of Engineering and Technology, Meerut, Uttar Pradesh, India
- shizaanand0507@gmail.com, mukesh.rawat@miet.ac.in

Some of the areas we have focused on to make your experience with OpenStack easy with less boiler-plate code are:

1. Fluent Interface
2. Expected results
3. Concrete API
    - Deployment Tested
    - Exception Handling

There are many limitations of the the above discussed 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.

2. Creating Containers with OpenStack4j-

A container is essentially a bucket of objects. Containers can have access security assigned to them. Algorithm given below showing the creation of container.

```
// Simple
os.objectStorage().containers().create("myContainerName")
;
// Full control
os.objectStorage().containers().create("myContainer",
CreateUpdateContainerOptions.create().accessAnybodyRe
ad() .accessWrite(acl).metadata(myMeta) //...);
```

Algo.2.1: Creation of a Container-

```
os.objectStorage().containers().update("myContainer",
CreateUpdateContainerOptions);
```

Algo.2.2: The command above shows the updation within a container
os.objectStorage().containers().delete("myContainer")
Algo.2.3: The command above shows the deletion of a container

3. Creating Objects using OpenStack4j-

Objects are uploaded files. Objects are associated to a container. Function below shows the Object creation, deletion and copying an object.
String s= os.objectStorage().objects().put(containerName,objectName, Payloads.create(someFile));

Algo.3.1: Object creation
os.objectStorage().objects().delete(containerName, objectName);
Algo.3.2: Object Deletion
String etag = os.objectStorage().objects().copy(ObjectLocation.create(srcContainer,srcObject) ObjectLocation.create(destContainer, destObject));
Algo.3.3: Copying an Object

4. Preparing a System for Implementing OpenStack4j-

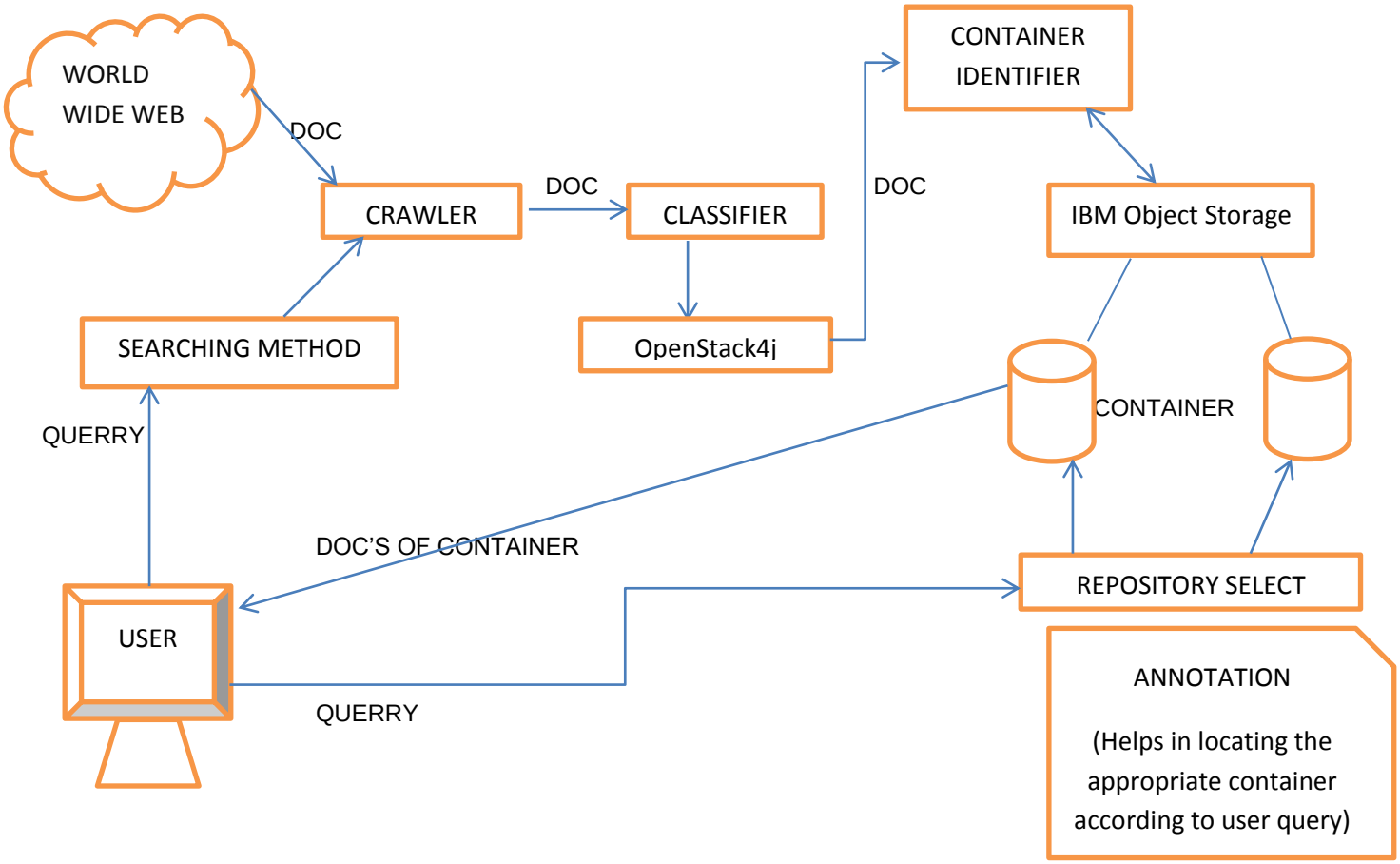

Fig: 4.1: *System for implementing OpenStack4j*

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.

5. Algorithm for Sending the documents to the Object Storage using OpenStack4j

```
Senddoctoobjstore(File f)
{
// credentials of using object storage service
String userId = "093f35db00f744f0bae29188ca4ad0e8";
String password = "vvg4u?_.XqNubs(6";
String auth_url = "https://identity.open.softlayer.com" + "/v3";
String domain = "880783";
String project = "object_storage_38b9f39b_a3a7_4f5b_84c3_53bec0836ba9";
Identifier domainIdent = Identifier.byName(domain);
Identifier projectIdent = Identifier.byName(project);
OSClient os = OSFactory.builderV3().endpoint(auth_url)
.credentials(userId, password).scopeToProject(projectIdent, domainIdent)
.authenticate();
SwiftAccount account = os.objectStorage().account().get();
List<? extends SwiftContainer> containers = os.objectStorage().containers().list();
//gettting the container name
String containerName=containers.get(0).getName();
// sending the file to respective container
String objectName=file.getName();
String etag = os.objectStorage().objects().put(containerName, objectName, Payloads.create(someFile));
}
```

6. Result Analysis

No. of Doc's	Clusters Created	Average Cluster Size	Cluster creation time taken (milisec)			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-6.1: *Table showing the comparison of time taken to transfer files and storage in cluster using OpenStack4j and other traditional techniques.*

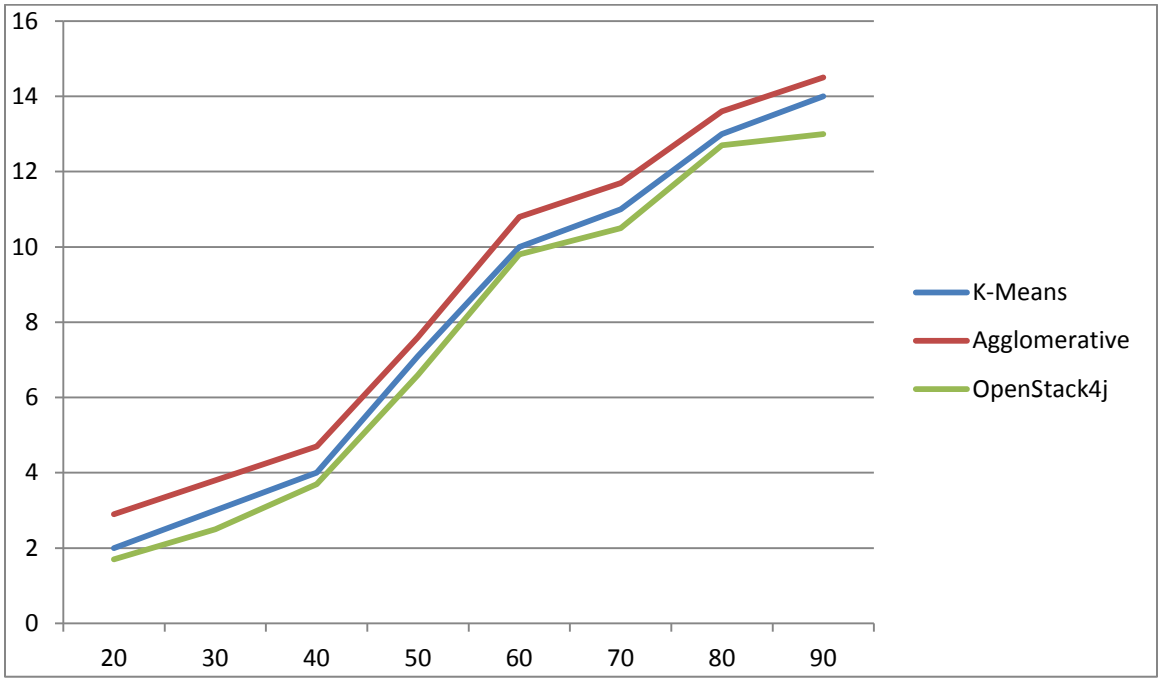

Fig-6.2: *Graph showing the comparison of time taken to transfer files and storage in cluster using OpenStack4j and other traditional techniques.*

7. 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, which the feature that 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.