International Journal of Intelligent Computing and Technology
ISSN: 2457-0249 (Online)
IJICT

Website: https://www.ijict.com **June (2021) - Vol. 5, Iss.1 Pages:13-20**

MIGRATION SCHEME FOR DATA CLONING IN DISTRIBUTED STORAGE SYSTEMS

[1]Dr. P. Revathi, [2]T. Nithya Rubini

[1]Assistant Professor, [2]M.C.A. Student

Holy Cross College, Trichy, Tamilnadu

Article History- Received: April 2021; Published: June 2021

Abstract

Data is the big think in the world, every few seconds people are shared large amount of data. Among the wide adoption of most important is internet services and big data, the cloud has become the perfect environment to satisfy the ever-growing storage demand data replication has been touted as the ultimate solution to improve data availability and reduce access time. Typically, cloud providers build several largescale data centers in geographically distributed locations. They then rely on data replication as an effective technique to improve fault tolerance, reduce end-user latency and minimize the amount of data exchanged through the network. Replica placement schemes may result in a large number of data replicas created or migrated over time between and within data centers, incurring significant amounts of traffic exchange. However, replica management systems usually need to migrate and create a large number of data replicas over time between and within data centers, incurring a large overhead in terms of network load and availability. This causes with a proposal of CRANE, an Efficient Replica Migration Scheme for Distributed Storage Systems.

Keywords: Migration, CRANE, Latency, Bucketization, Data Replica, Cloning

INTRODUCTION

With the wide implementation of large-scale Internet services and the increasing amounts of exchanged data, the cloud has become the final resort to provide to the ever-growing demand for storage, providing on the face of it limitless capacity, high availability and faster access time. Typically, cloud providers build several large-scale data centers in geographically distributed locations. Reduce end-user latency and minimize the amount of data exchanged through the network. Consequently, effective replica management has become one of the major challenges for cloud providers [1].

In recent years, a large body of work has been committed to address this challenge and, more specifically, to address the problem of replica placement considering several goals, such as minimizing storage costs, improving fault-tolerance and access delays. However, replica placement schemes may result in a large number of data replicas created or migrated over time between and within data centers, incurring significant amounts of traffic exchange. This ever-growing exchange of large amounts of data between data centers may overload the network, especially when using the same paths or links. This can hurt the overall network performance in terms of latency and packet loss. Moreover, replica migration processes are usually distributed is the case for Swift, managing to store data storage [3] . That is, when a replica is to be relocated or created in a new destination machine, every machine in the infrastructure already storing the same replica will try to copy the data to the new destination.

PROPOSED SYSTEM

CRANE complements any replica placement algorithm by efficiently managing replica creation in geo-distributed infrastructures in order to minimize the time needed to copy the data to the new replica location, to avoid network congestion, and also to ensure the minimum desired availability for the data. Through simulation and experimental results, data set is subject to the CRANE mechanism which provides a sub-optimal solution for the replica migration problem with lower computational complexity than its integer linear program formulation.

CLASSIFICATION PROCESS FOR BUCKETIZATION

Upload data

The user only can upload the data and the data details should uploaded database like file size, file name, type of file etc as shown in Fig.3.

Transferring data from one remote system to another under the control of a local system is remote uploading. Remote uploading is used by some online file hosting services [4]. It is also used when the local computer has a slow connection to the remote systems, but they have a fast connection between them. Without remote uploading functionality, the data would have to first be download to local host and then uploaded to the remote file hosting server, both times over slow connections.

Partitioning the data

Using Clustering data method, then the given data should divide into some partition and then the partition data should store on buckets as shown in Fig.3&4. Finally, the bucket stored on big data environment.

Request for the stored data

Then the user can send the request for uploaded data for further accessing like adding content to the data. Transferring data from one remote system to another under the control of a local system is remote uploading.

Unite Data for Accessing

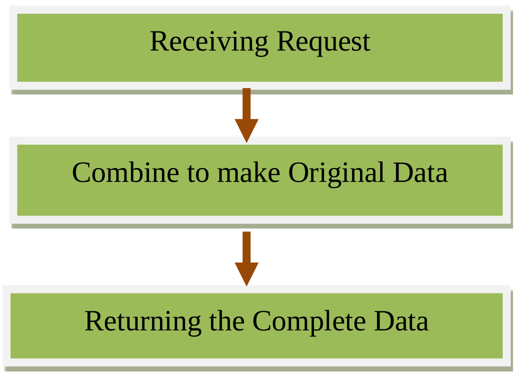

Fig.1 Data Accessing

Here, the partitioned data are combined to make original data and the data should be downloaded by the user. The File Transfer Protocol (FTP) is a standard network protocol used to transfer computer files between a client and server on a computer network. FTP is built on client-server model architecture and uses separate control and data connections between the client and the server refer Fig.1.

Support Vector Machines (SVM)

SVM is a supervised classification is one of the most important Machines Learning algorithms in Python, plots a line that divides different categories of your data. In this ML algorithm, we calculate the vector to optimize the line. This is to ensure that the closest point in each group lies farthest from each other.

While finding this to be a linear vector, it can be other than that from sklearn. datasets.samples_generator as shown in Fig.2.

import make_blobs

x,y=make_blobs(n_samples=500,centers=2, random_state=0,cluster_std=0.40)

import matplotlib.pyplot as plt

plt.scatter(x[:,0],x[:,1],c=y,s=50,cmap='plasma')

plt.show()

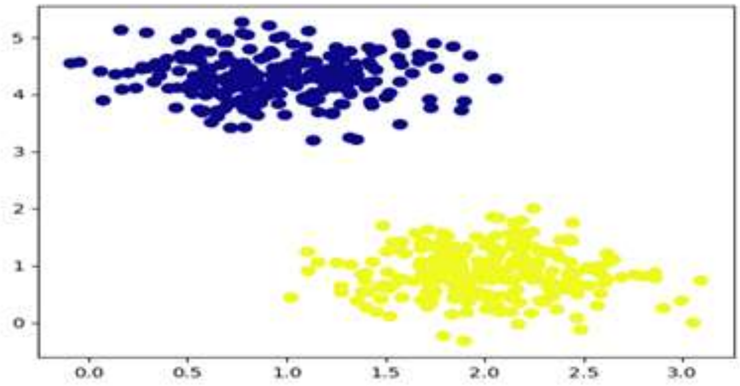

Fig.2 Scatter Plot

Table 1 Sample Data

ID	AGE	SIZE	GENDER	SOURCE	TARGET	DOWNLOAD
George Keith	63	10	Male	1	0	0
Robert Barclay	37	5	Male	2	0	0
Benjamin Furly	41	3	Male	3	0	2
Anne Conway	56	8	Female	3	1	2
Franciscus	57	4	Male	4	0	2
William Penn	57	8	Male	4	2	1
George Fox	56	2	Male	5	0	1
George Whitehead	44	8	Male	5	1	2
William Bradford	52	6	Male	5	2	2
James Parnel	57	1	Male	5	3	2
Stephen Crisp	54	8	Male	5	4	2
Peter Collinson	48	8	Male	6	0	2
John Bartram	49	6	Male	6	1	2
James Logan	64	3	Male	6	4	1
Joseph Wyeth	58	9	Male	6	5	2
Thomas Ellwood	50	7	Male	7	2	1
Dorcas Erbery	58	100	Female	7	5	2
James Nayler	66	100	Male	7	6	0
William Mucklow	43	99	Male	8	3	2
William Dewsbury	69	88	Male	9	3	2
Edward Burrough	59	66	Male	9	8	1
John Crook	44	33	Male	10	3	2

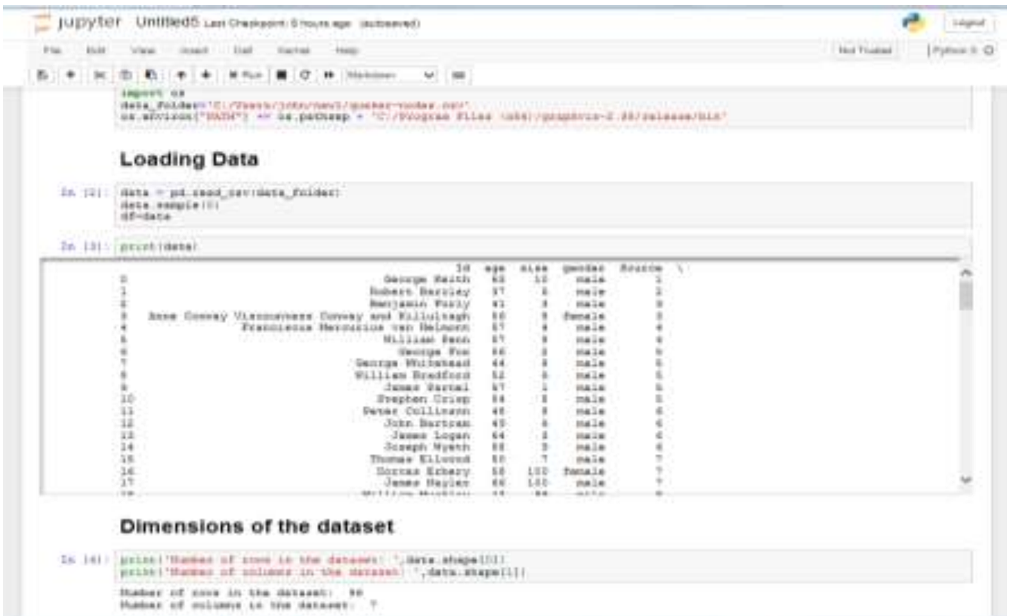

Fig.3 Loading Data

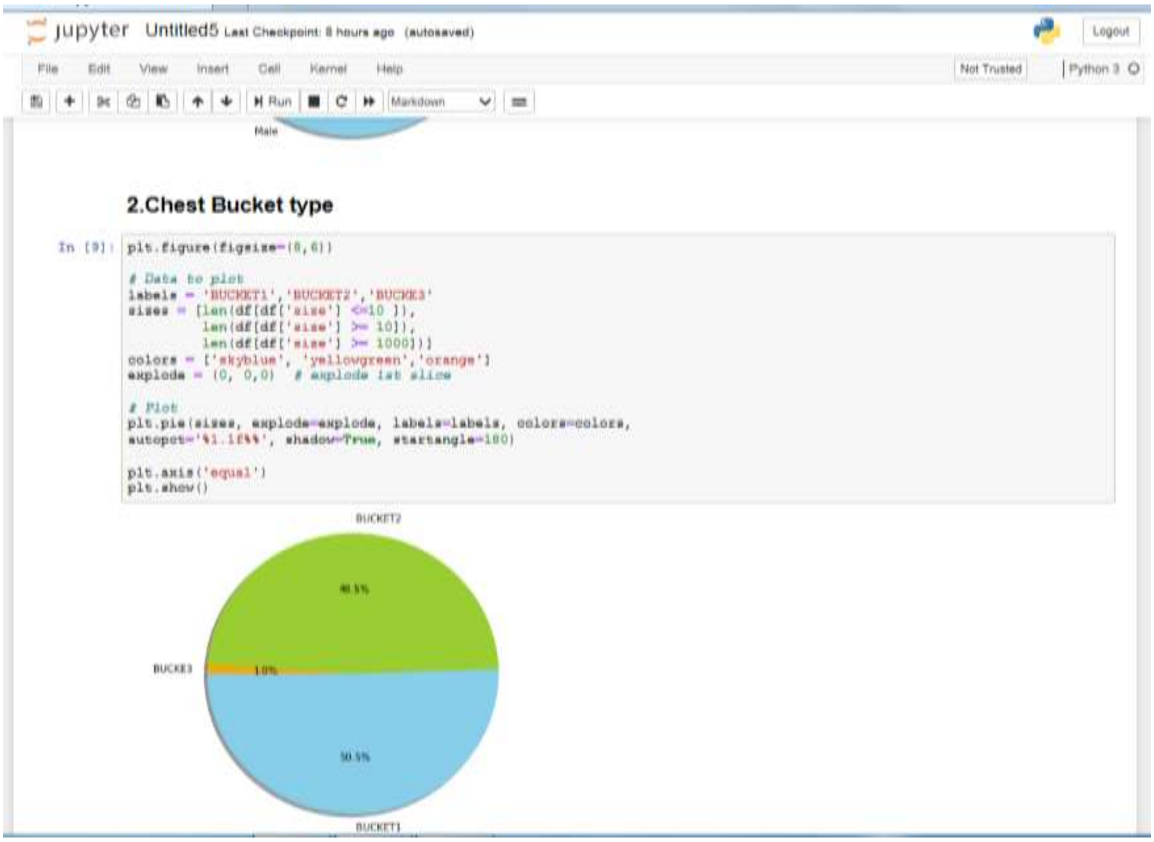

Fig.4 Checking Bucket Type

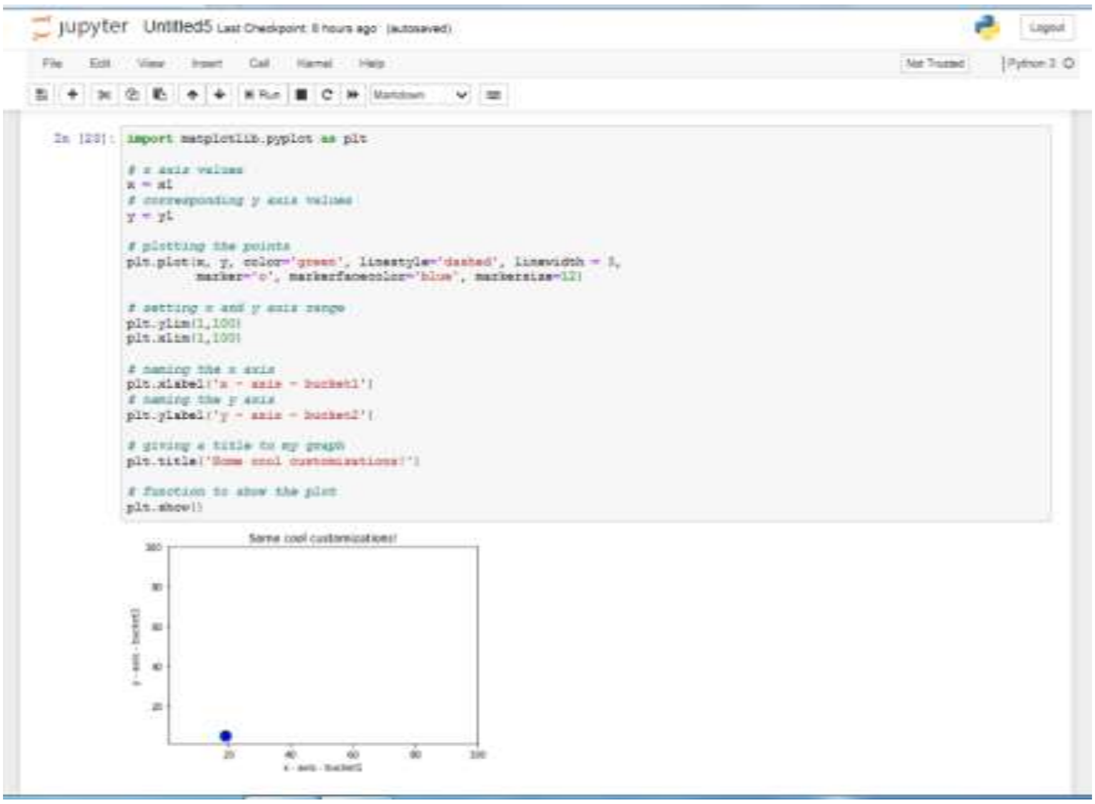

Fig.5 Comparison between buckets

Results and Findings

Thus this work shows the migration scheme leads to Easy migration process with no Congestion in network and with which the data cloning sequences gives Easy access of files and prompt availability of Data. In multi-VDI (Virtual Desktop Infrastructure) environment, it happens because of the low cross-similarity of virtual environment or due to the fact that the chunks with same content are exactly hashed to the same host. The comparison between Dataset using different bucket sizes will focus the I/O requests in this situation. The network traffic has effects on the remote chunk accesses and deals with this negative effect by using mapping strategies. This is investigated to optimize the I/O performance and improve the resource utilization further in this kind of heterogeneous network environment. Some production environments are also used for dedicated storage and management networks along with the bucketing scheme. Due to the uneven traffic pattern, the situation becomes more complex. And this investigation also shows the controlled performance of networking to enhance and improve the Virtual resource consumption in implicit environment, as the host on a Central Server.

Conclusion

Data replication has been generally adopted to improve data availability and to reduce access time. However, replica placement systems usually need to migrate and create a huge number of replicas between and within data centers, incurring a large overhead in terms of network load and availability. In this paper, the proposed method CRANE, an effiCient Replica migration scheme for distributed cloud Storage systems. CRANE complements replica placement algorithms by efficiently managing replica creation by minimizing the time needed to print data to the new replica location while avoiding network congestion and ensuring the required availability of the data. In order to evaluate the performance of CRANE, we compare it to the optimal result for the replica migration problem considering availability and to the standard swift, the Open Stack project for managing data storage. Results show that CRANE has sub-optimal performances in terms of migration time and an optimal amount of transferred data. Moreover, experiments show that CRANE is able to reduce up to 60% of the replica creation time and 50% of inter-data center network traffic and provide better data availability during the process of replica migration [5]. In the future work, this proposal can be enhanced with larger scale simulations to achieve high metric levels to scrutinize the performance of our heuristic. Other improvements will also be considered to address reliability and consistency requirements.

References

[1] B. A. Milani and N. J. Navimipour, "A comprehensive review of the data replication techniques in the cloud environments: Major trends and future directions," Journal of Network and Computer Applications, vol. 64, pp. 229–238, 2016.
[2] S. Ghemawat, H. Gobioff, and S.-T. Leung, The Google file system. ACM, 2003, vol. 37, no. 5.
[3] R.-S. Chang and H.-P. Chang, "A dynamic data replication strategy using access-weights in data grids," The Journal of Supercomputing, vol. 45, no. 3, 2008.
[4] Q. Wei, B. Veeravalli, B. Gong, L. Zeng, and D. Feng, "CDRM: A cost-effective dynamic replication management scheme for cloud storage cluster," in IEEE International Conference on Cluster Computing (CLUSTER),2010, pp. 188–196.
[5] D.-W. Sun, G.-R. Chang, S. Gao, L.-Z. Jin, and X.-W. Wang, "Modeling a dynamic data replication strategy to increase system availability in cloud computing environments," Journal of Computer Science and Technology, vol. 27, no. 2, pp. 256–272, 2012.

How to cite this article:

Dr. P. Revathi, T. Nithya Rubini, "Migration Scheme for Data Cloning in Distributed Storage Systems", International Journal of Intelligent Computing and Technology (IJICT), Vol.5, Iss.1, pp.13-20, 2021