International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 7
July 2017

# A Cost Efficient Multi-cloud Data Hosting Using Heuristic Data Placement Algorithm

**Ms. Pooja Rade** Department of Computer Engineering RSCOE, Pune

**Ms V. M. Barkade** Department of Computer Engineering RSCOE, Pune

*Abstract—In recent years, most of the enterprises an organizations are using cloud to host their data into the cloud. It will results to reduce the IT maintenance cost and enhance the data reliability. In this cloud storage, digital data is stored in logical pools, the physical storage spans multiple servers (and often locations), and the physical environment is typically owned and managed by a hosting company. These cloud storage providers are responsible for keeping the data available and accessible, and the physical environment protected and running. People and organizations buy or lease storage capacity from the providers to store user, organization, or application data. There are numerous cloud vendors as well as their heterogeneous pricing policies are available, in which customers may getting confused with which cloud(s) are suitable for storing their data and what hosting strategy is cheaper. To solve this problem, comprehensive analysis should be done for customers understanding, that will help to decide which storage is suitable for them. The started is use to select several suitable clouds and an appropriate redundancy strategy to store data with minimized monetary cost and guaranteed availability. The second is triggering a transition process to re-distribute data according to the variations of data access pattern and pricing of clouds. Also in this system we introduce the concept of data de-duplication to reduce the storage space requirement by the organizations. With the help of data de-duplication System save only one copy of the data and replace all other copies with a pointer which points to the original data file.*

***Index Terms—Cloud Service, cloud computing, data hosting, vendor lock, multi-cloud, data de-duplication.***

## I. Introduction

As of late, research saw that the online data hosting services have an extraordinary prevalence. Online storage may allude to computer information storage on a medium or a device that is under the control of a processing unit, that is capacity that is not offline storage, online file storage gave by a file hosting service, cloud storage, a model of networked enterprise storage. A file hosting service, cloud storage service, online file storage provider, or cyber locker is an Internet hosting service particularly intended to host client records. It permits clients to transfer records that could then be retrieved over the web from an alternate computer, tablet, mobile phone or other networked device, by a same client or perhaps by different clients, after a password or other validation is given. Ordinarily, the services allow HTTP access, as well as some of the time FTP access to. Related services are content-displaying hosting services, virtual storage, and remote backup.

Some online document storage services offer space on a for each gigabyte basis, and in some cases incorporate a data transmission cost component also. Typically these will be charged month to month or yearly. A few organizations offer the service for nothing, depending on advertising revenue. Some hosting services don’t put any farthest point on how much space the client’s account can expend. A few services require a product download which makes documents just accessible on computers which have that software installed; others permit clients to recover records through any web browser. Security related to data stored as well as information while transferring must be considered when storing important information at cloud storage provider.

Clients with particular records-keeping prerequisites, for example, public agencies that must hold electronic records as per statute, may experience inconveniences with utilizing cloud computing and storage. For example, the U.S. Department of Defense assigned the Defense Information Systems (DISA) to keep up a list of records management items that meet the majority of the records retention, by and by identifiable information (PII), and security necessities. It is the concomitant utilization of at least two cloud services to reduce the risk of far reaching information loss or downtime because of a localized component failure in a cloud computing environment. Such a breakdown can happen in hardware, software, or infrastructure. A multi-cloud technique can likewise enhance general enterprise execution by maintaining a strategic distance from “vendor lock-in” as well as utilizing distinctive frameworks to address the issues of differing accomplices and clients.

Existing cloud indicate unfathomable heterogeneities in term of both working exhibitions and evaluating policies. Particular cloud providers construct their individual framework and keep updating them with as of late developing gears. They in like manner plan various structure models and apply diverse methodology to make their organizations aggressive. Such system differences qualities prompt recognizable execution assortments crosswise over cloud providers. As of late, the greater part of the organizations and associations are utilizing cloud to have their information into the cloud. It will result to minimize the IT maintenance cost and upgrade the information reliability. Inside cloud storage data in digital manner is stored in logical pools, the physical storage traverses different servers, and the physical environment is regularly owned and handled by a hosting organization. These cloud storage suppliers are in charge of keeping the information accessible and available, and the physical environment secured and running. Individuals and associations purchase or rent storage capacity from the suppliers to store client, association, or application information. There are various cloud providers and in addition their heterogeneous pricing plans are accessible, in which clients may get confused for selecting cloud(s) are appropriate for storing away their information and what hosting methodology is less expensive. To tackle this issue, thorough investigation must accomplish for clients understanding, that will choose which storage is appropriate for them. They began is use to choose a few reasonable clouds and a proper redundancy system to store information with minimized money related cost and ensured accessibility. The second is triggering a transition procedure to re-disperse information as per the varieties of information get to pattern and evaluating of clouds.

In this paper we study about the related work done, in section II, the proposed approach modules description, mathematical modeling, algorithm and experimental setup in section III .and at final we provide a conclusion in section IV.

## II. Review of Literature

In this paper [1] authors developed a new data hosting system known as CHARM that consolidates two key capacities desired, in view of comprehensive investigation of various best in class cloud providers. The first is selecting a couple of reasonable clouds and a proper redundancy methodology to store data with minimized money related cost and guaranteed accessibility. The second is triggering off a move strategy to re-circulate data as showed by the varieties of data access pattern and evaluating of clouds.

In this paper [2] authors solved a fundamental yet basic inquiry: Is the present data sync traffic of cloud storage services productively used. They first describe a novel metric known as TUE to assess the Traffic Use Efficiency of data synchronization. In perspective of both real-world traces and thorough examinations, they look at and describe the TUE of six for the most part used cloud storage services.

In this paper [3] authors developed the update-batched delayed synchronization (UDS) segment to address the activity abuse issue. Going about as a middleware between the cus- tomer's record stockpiling structure and a distributed storage application, UDS clusters redesigns from customers to fundamentally diminish the overhead brought on by session upkeep movement, while protecting the quick archive synchronization that customers anticipate from cloud storage administrations. Moreover, they expand UDS with a regressive good Linux kernel modification that further advances the execution of cloud storage applications by diminishing the CPU usage.

In this paper [4] authors developed a DEPSKY, a framework that enhances the accessibility, integrity and privacy of data stored in the cloud through the encryption, encoding and replication of the information on different clouds that shape a cloud-of-clouds. They conveyed their framework utilizing four business clouds and utilized Planet Lab to run customers getting to the service from various countries. They observed that their protocols enhanced the perceived availability and, by and large, the get to latency when contrasted and cloud suppliers exclusively.

In this paper [5] authors given an study of file system snapshot and five month access trace of a campus cloud storage framework which has been stored on Tsinghua campus for three years. The framework gives online storage as well as sharing of information services for greater than 19,000 students as well as 500 student groups. Authors report various data characteristics including file size as well as file type, and some access patterns, having read/write ratio, read-write dependency as well as daily traffic. They search that there are various verifications in cloud storage framework as well as conventional file systems: their cloud framework has huge file sizes, lower read/write ratio, as well as low size of set of active files than those of a typical traditional file system.

In paper [6], authors does the first systematic analysis on advance content multi homing, by implementing algorithms for optimizing performance as well as cost for content multi homing. In specific, they implemented new, efficient algorithm to calculate assignments of content objects for content distri- bution networks to content publishers, taking cost as well as performance in mind. They also implemented low size client adaptation algorithm executing at individual content viewers to achieve scalable, fine-grained, fast online adaptation for optimizing the quality of experience (QoE) for individual viewers.

In paper [7], authors developed Scalia, a cloud storage brokerage solution which constantly adapts the placement of information depending on its access pattern as well as related to optimization objectives, like storage costs. Scalia efficiently takes repositioning of only chosen objects which may significantly reduce the storage cost. They showed the cost-effectiveness of Scalia in oppose of static placements as well as its proximity to the ideal information placement in numerous scenarios of information access patterns, of possible cloud storage solutions as well as of failures

## III. System Architecture / System Overview

### A. Proposed System Overview

This system proposes a novel cost-effective information facilitating plan with high accessibility in heterogeneous multi-cloud, named "CHARM". It insightfully places information into different clouds with minimized financial expense and ensured accessibility. In particular, system join the two broadly utilized redundancy systems, i.e., replication and deletion cod- ing, into a uniform model to meet the required accessibility in the presence of diverse information access designs. Next, sys- tem designs an effective heuristic-based calculation to choose appropriate information storage modes. Additionally, system actualizes the essential method for storage mode transition by checking the varieties of information access designs and pricing policies. System evaluates the performance of CHARM using both trace driven simulations and prototype experiments. The traces are collected from two online storage systems: AmazingStore [7] and Corsair [8], both of which possess

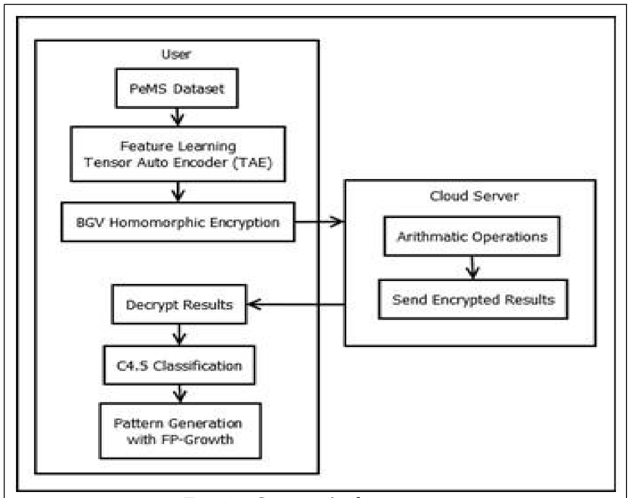

Fig. 1. System Architecture

hundreds of thousands of users.

There are four main components in CHARM:

- Data Hosting

Data Hosting stores data using replication or erasure coding, according to the size and access frequency of the data. Data Hosting is important modules in CHARM. Data Hosting decides storage mode and the clouds that the data should be stored in.

- Storage Mode Switching (SMS)

SMS decides whether the storage mode of certain data should be changed from replication to erasure coding or in reverse, according to the output of Predictor. The implementation of changing storage mode runs in the background, in order not to impact online service.

- Workload Statistic

Workload Statistic keeps collecting and tackling access logs to guide the placement of data. It also sends statistic information to Predictor which guides the action of SMS.

- Predictor

Predictor is used to predict the future access frequency of files. The time interval for prediction is one month, that is, we use the former months to predict access frequency of files in the next month. However, we do not put emphasis on the design of predictor, because there have been lots of good algorithms for prediction. Moreover, a

$P_i$ si
r bi

very simple predictor, which uses the weighted moving average approach, works well in our data hosting model.

B. Mathematical Model

System S is represented as S= { U, E, H, C, D, S }

1) Users

U= { u1, u2, u3,.....,un }

Where, U is the set of different users and u1, u2, u3,....,un are the number of users.

$P_i$ is cloud

si = Size of file

$P_{si}$ = Storage Price

2) Input Files

E= { e1, e2, e3,....en}

E is represent as a set of Input files and e1, e2, e3,....,en is a number of input files.

3) Heuristic Algorithm(H) Input = { FS, CR, NU } FS : File Size

CR: Current Frequency NU: n's upper limit Output= { $C_{sm}$, $P_{sy}$ } $C_{sm}$ = Minimal Cost $P_{sy}$ = (c1, c2, .......) Selected Cloud

4) Storage Mode Transition Process

Input:

T = Generated Table by heuristic algorithm. MI = File current storage node.

where MI = { CM, HM}

where, CM = Cold Storage Mode. HM = Hot Storage Mode.

CR = Current Read Frequency. FS= File Size.

Output= Change of storage node.

5) Data de-duplication

D = Check the deduplication file exist in the cloud server.

6) Cloud Storage

C = { c1, c2, c3}

Where, C is set of cloud storage and c1, c2, c3 represents different clouds like c1: amazon c2: Google c3: CloudOrg

**Mathematical Equation with Example**

- Sort Cloud By Normalised

β —

$$\delta_i = \alpha a_i + \quad P_i \qquad (1)$$

Where

α = Index of ith cloud

β = Minimum bandwidth of all clouds

$a_i$ = Available of cloud.

Size of File

r oi

Where $_1 = SP \quad + C \ SP \quad + C \ P \quad - \text{cloud1}$ (2)

$C_r$ = no. of read operator

$P_{bi}$ = bandwidth

$P_{oi}$ = Get operation price

$$P_{i_2} = SP_{si} + C_r SP_{bi} + C_r P_{oi} - \text{cloud2} \qquad (3)$$

$P_{i_3} = SP_{si} + C_r SP_{bi} + C_r P_{oi}$ – cloud3 (4)

$P_{i_4} = SP_{si} + C_r SP_{bi} + C_r P_{oi}$ – cloud4 (5)

$Max = (P_{i_1}, P_{i_2}) \& (P_{i_3}, P_{i_4})$

$Min = (P_{i_1}, P_{i_2}) \& (P_{i_3}, P_{i_4})$

$MinBandwith = \min(P_{i_1}, P_{i_2}), \min(P_{i_3}, P_{i_4})$

$\beta$ = MinBandwith $-\!\!\rightarrow$ put in eq 6.1.

diff = (max - min)

Normalized = (cloud - min) / diff

- Check availability of cloud with combined with dataset

$a_i$ = availability

$availability = totalAval$ (6)

Given: availability Gs = 0.0 n=2

$a_i$ = 0.99

total Aval = 0

database {“a”, “b”, “c”, “d”}

mul 1 = 1 mul 2 = 1

mul1 = mul1 $* a_i$

mul2 = mul2 $* (1 - a_i)$

total Aval = total Aval + (mul 1 $*$ mul 2)

put in eq 6.6.

Combination check from dataset.

### C. Algorithm Used

Algorithm 1: De-duplication Checking : 1. dbHash=getting files hash from the client side database.

2. FileHash=users chunked file’s hash

H(New chunk) = h H(Old n chunks) = $h_n$ Compare h and $h_n$

If H(New chunk) == H(Old n chunks)

Chunk is duplicate and refuse it to store on public server

Else

Chunk is not duplicate and allowed to store on public server

Algorithm 2: Heuristic algorithm of data placement

Input: file size S, read frequency $C_r$, n’s upper limit ε

Output: minimal cost Csm, the set of the selected clouds

1) Initialize Minimum cost = ∞ and set of cloud = **{}** and sort cloud from high to low

2) traverse n=2 to

4) traverse m=1 to n to select appropriate cloud with minimum cost for storage.

5) calculate availability of selected cloud Gs. if Acur> = require availability then calculate minimal cost.

$$P_r(N^0, k) = \sum_{j=1}^{(|N^0|)} \prod_{i \in C_j^{|N^0|,k}} a_i . \prod_{i \in N^0 \; C_j^{|N^0|,k}} (1 - a_i)]$$

$[\Pi$

$$\sum_{k=m}^{|N^0|} P_r(N^0, K) = \sum_{k=m}^{|N^0|} \sum_{j=1}^{|N^0|} [\prod_{i \in C_j^{|N^0|,k}} a_i . \prod_{i \in N^0 \; C_j^{|N^0|,k}} (1 - a_i)]$$

3) assign first cloud to Gs from the list of cloud Ls.

- remove that cloud from list Ls.
- store remaining cloud in Gc.

$j$

$i$

6) if Ccur is < than the Csm then assign Ccur to Csm and put the selected cloud in the list ψ

7) if the availability does not meet the req. value exchange the cloud in the current set Gs using greedy method.

8) firstly sort Gs by ai and Gs by Pi from low to high where Pi= $SP_{si} + c_r\, SP_{bi} + c_r P_{oi}$ .

9) then exchange the cloud in Gs from the lowest ai one by one with the cloud which has lowest Pi in Gc but higher availability than that cloud in Gs.until the availability meets the required value.

10) if the cost of obtain Gs is lower then update Csm and ψ.

## IV. Results and Discussion

### A. Experimental Setup

The system is built using Java framework on Windows platform. The Net beans IDE is used as a development tool. The system doesn’t require any specific hardware to run; any standard machine is capable of running the application.

### B. Expected Result

Table I depicts the prediction time of proposed system and existing system. Comparison table shows the time required for system with de-duplication and without de-duplication.

TABLE I
Time Comparison

| System | Existing System |
|---|---|
| System with de-duplication | 1100 minutes |
| System without de-duplication | 1300 minutes |

Figure 2 represent the graphical comparison of existing and proposed system on the basis of time required for the algorithm implementation.

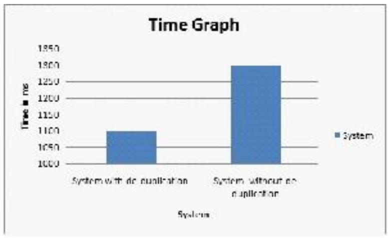

Fig. 2. Time Comparison

Table II depicts the prediction memory required for pro- posed system and existing system. Comparison table shows the memory required for system with de-duplication and without de-duplication.

TABLE II
MEMORY COMPARISON

| System | Existing System |
|---|---|
| System with de-duplication | 2100 minutes |
| System without de-duplication | 2400 minutes |

Figure 2 represent the graphical comparison of existing and proposed system on the basis of memory required for the algorithm implementation.

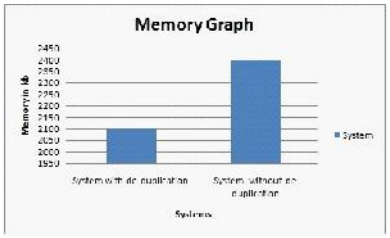

Fig. 3. Memory Comparison

## V. CONCLUSION

Cloud service provider are encountering quick advancement and the administrations based on multi-cloud additionally get to be prevailing. One of the most concerns, when moving services into cloud, is capital consumption. Along these lines, in this system, we outline a novel storage plan , which guides clients to store the data among the cloud in appropriate way with minimal cost selection of cloud. This system used the concept of de-duplication which reduce the storage space on cloud.

In future the system will became more powerful if it com- bine for block level as well as file level deduplication check. This will overcome the drawback that if whole file is same then also here it check the duplication chunk by chunk. It is avoided if we first perform the file level deduplication and if file is unique then only go for the block level deduplication check. Also checked for another encryption algorithm to provide the more security.

## ACKNOWLEDGMENT

The authors would like to thank the researchers as well as publishers for making their resources available and teachers for their guidance. We are thankful to the authorities of Savitribai Phule University, Pune and concern members of cPGCON2017 conference, organized by, for their constant guidelines and support. We are also thankful to the reviewer for their valuable suggestions.

## REFERENCES

[1] Quanlu Zhang, Shenglong Li, Zhenhua Li, Yuanjian Xing, Zhi Yang, and Yafei Dai, “CHARM: A Cost-efficient Multi-cloud Data Hosting Scheme with High Availability”, IEEE Transactions on Cloud Computing, 2015.

[2] Z. Li, C. Jin, T. Xu, C. Wilson, Y. Liu, L. Cheng, Y. Liu, Y. Dai, and Z.-L.Zhang, “Towards Network-level Efficiency for Cloud Storage Services,” in IMC. ACM, 2014.

[3] Z. Li, C. Wilson, Z. Jiang, Y. Liu, B. Y. Zhao, C. Jin, Z.-L. Zhang, and Y.Dai, “Efficient Batched Synchronization in Dropbox-like Cloud Storage Services,” in Middleware. ACM/IFIP/USENIX, 2013.

[4] A. Bessani, M. Correia, B. Quaresma, F. Andre, and P. Sousa, “DepSky: Dependable and Secure Storage in a Cloud-of-Clouds,” in EuroSys. ACM, 2013.

[5] S. Liu, X. Huang, H. Fu, and G. Yang, “Understanding Data Character- istics and Access Patterns in a Cloud Storage System,” in CCGrid. IEEE, 2013.

[6] H. H. Liu, Y. Wang, Y. R. Yang, H. Wang, and C. Tian, “Optimizing Cost and Performance for Content Multihoming,” 2012.

[7] T. G. Papaioannou, N. Bonvin, and K. Aberer, “Scalia: An Adaptive Scheme for Efficient Multi-cloud Storage,” in SC. IEEE, 2012.