# CHARM: A Survey on Multi-Cloud Hosting for Performance and Cost-efficient in Cloud Computing

**Mallikarjunaiah K M,** (M.Tech) Dept. of CSE, AIET, Moodbidri

**Harish Kunder,** Assistant Professor, Dept. of CSE, AIET, Moodbidri

***Abstract: Cloud computing is used to store data from various resources by the user. It is difficult for the user to store entire data within the system; therefore clouds are formed to store the user data. More enterprises and organizations are hosting their data into the cloud, in order to reduce the IT maintenance cost and enhance the data reliability. IT resources are rapidly and elastically provisioned and provided as standardized subscription to users over the internet in a flexible pricing model and effort by interacting with the service provider. Cost is also a major issue in cloud computing when we are switching to multi cloud. Based on comprehensive analysis of various state of the art cloud vendors, this paper proposes a novel data hosting scheme (CHARM) which integrates two key functions desired. The first is selecting 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. While sending data to third party administrative control in cloud, it also becomes an issue in cloud related to security. The efficient dynamic collaboration of multiple clouds provide several potential benefits, such as high availability, scalability, fault tolerance and reduced infrastructural cost.***

***Keywords: Multi-cloud, data hosting, cloud storage.***

## I. INTRODUCTION

Cloud computing gets its name as a metaphor for today's internet world. Cloud typically contains an outstanding pool of resources, which can be reallocated to different purposes within short span of frames. The process is typically automated and takes minute. Recent years have witnessed online data hosting services such as Amazon S3, Windows Azure, Google Cloud Storage, Aliyun OSS [1], and so forth. These services provide customers with reliable, scalable, and low-cost data hosting functionality. To accessed these cloud services security and reliability we are using different models like: i) Using single service provider. ii) Using multiple service providers. The weakness of single service provider is that it can be easily be hacked by intruders and if the service provider fails or down for some technical reasons than client will not at all access his/her data. The problem in multiple service provider models is to compromise the security because there is lack of security techniques. More and more enterprises and organizations are hosting all or part of their data into the cloud, in order to reduce the IT maintenance cost and enhance the data reliability [2], [3], [4]. For example, the United States Library of Congress had moved its digitized content to the cloud, followed by the New York Public Library and Biodiversity Heritage Library [5]. Now they only have to pay for exactly how much they have used.

In Cloud computing storing and sharing of data is been done via trusted third party. For a cloud to be secure, all of the participating entities must be secure. The highest level of the system's security is equal to the security level of the weakest entity. Therefore, in a cloud, the security of data does not solely depend on an individual's security measures. The neighbouring entities are also responsible to provide an opportunity to an attacker to tackle the user's defences. The data outsourced to a public cloud must be secured. Unauthorized data access by other users and processes whether it may be accidental or deliberate must be protected. For a cloud provider, such answers can point it in the right direction for improvements. For instance, a provider should pour more resources into optimizing table storage if the performance of its store lags behind competitors.

Multi-cloud data hosting has received wide attention from researchers, customers, and start-ups. The basic principle of multi-cloud (data hosting) is to distribute data across multiple clouds to gain enhanced redundancy and prevent the vendor lock-in risk, as shown in Figure 1. The "proxy" component plays a key role by redirecting requests from client applications and coordinating data distribution among multiple clouds. The potential prevalence of multi-cloud is illustrated in three folds. First, there have been a few researches conducted on multi-cloud.

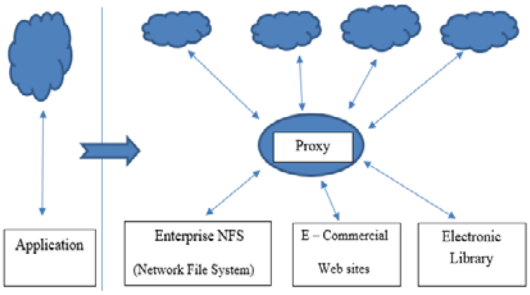

**Figure 1: Basic principle of multi cloud data hosting**

The proposed CHARM scheme. In this paper, we propose a novel cost-efficient data hosting scheme with high availability in heterogeneous property in multi-cloud, named “CHARM”. It intelligently puts data into multiple clouds with minimized monetary cost and guaranteed availability. Specifically, we combine the two widely used redundancy mechanisms, i.e., replication and erasure coding, into a uniform model to meet the required availability in the presence of different data access patterns. Next, we design an efficient heuristic-based algorithm to choose proper data storage modes involving both clouds and redundancy mechanisms. Moreover, we implement the necessary procedure for storage mode transition by monitoring the variations of data access patterns and pricing policies.

## II. Background

### A. Pricing Models of Mainstream Clouds

In order to understand the pricing models of mainstream cloud vendors, we select to study five most popular cloud storage services across the world: Amazon S3, Windows Azure, Google Cloud Storage, Rackspace, and Aliyun OSS (deployed in China). Their latest pricing models (in 2014) are presented in Table I (Storage and bandwidth pricing) and Table II (Operation pricing). Basically for these clouds, customers are charged in terms of storage, out-going (i.e., from cloud to client) bandwidth and operations. Data Hosting stores data using replication or erasure coding, according to the size and access frequency of the data. 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.

**TABLE 1**

**PRICES OF STORAGE (IN $/GB/MONTH) AND OUT-GOING BANDWIDTH (IN $/GB).**

| CLOUD | Amazon S3 | | | Windows Azure | | | Google Cloud Storage | | Rackspace | Aliyun OSS |
|---|---|---|---|---|---|---|---|---|---|---|
| | Tokyo | Singapore | America | US East | US RA-GRS | Asia | America | Asia pacific | All Regions | China |
| **Storage** | 0.033 | 0.03 | 0.03 | 0.0243 | 0.0616 | 0.024 | 0.026 | 0.026 | 0.105 | 0.028 |
| **Out-going Bandwidth** | 0.201 | 0.19 | 0.12 | 0.12 | 0.12 | 0.9 | 0.12 | 0.21 | 0.2 | 0.116 |

### B. Erasure Coding

Erasure coding has been widely applied in storage systems in order to provide high availability and reliability while introducing low storage overhead [6]. As we all know, the storage mode of “three replicas” is putting replicas into three different storage nodes. Then the data is lost only when the three nodes all crash. However, it occupies 2x more storage space. Erasure coding is proposed to reduce storage consumption greatly while guaranteeing the same or higher level of data reliability.

### C. Combining Replication and Erasure Coding

In existing industrial data hosting systems, data availability (and reliability) are usually guaranteed by replication or erasure coding. In the multi-cloud scenario, we also use them to meet different availability requirements, but the implementation is different. For replication, replicas are put into several clouds, and a read access is only served (unless this cloud is unavailable then) by the “cheapest” cloud that charges minimal for out-going bandwidth and GET operation.

## III. Data Hosting Scheme

### A. CHARM Overview

In this section, we elaborate a cost-efficient data hosting model with high availability in heterogeneous clouds in multi-cloud named “CHARM”. The architecture of CHARM is shown in Figure 2. The whole model is located in the proxy in Fig 1. There are four main components in CHARM: Data Hosting, Storage Mode Switching (SMS), Workload Statistic, and Predictor. CloudCmp enables predicting application performance without having to first port the application onto every cloud provider.

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. Data Hosting stores data using replication or erasure coding, according to the size and access frequency of the data. 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. Data Hosting and SMS are two important modules in CHARM. Data Hosting decides storage mode and the clouds that the data should be stored in. This is a complex integer programming problem demonstrated in the following subsections. We first demonstrate the implementation of storage mode transition: the proxy gets the data from the clouds where the data is originally stored, and puts it into the newly selected clouds using new storage mode. The implementation consumes out-going bandwidth, in-going bandwidth, and several operations (i.e., GET, DELETE, and PUT).

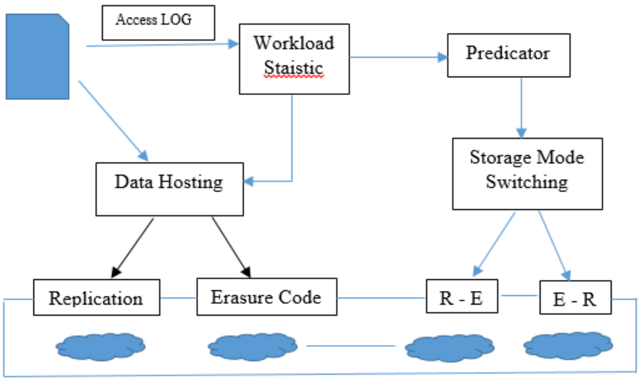

**Figure 2: The architecture of CHARM. "R" represents replication and "E" represents erasure coding**

### B. Heuristic Solution

We first assign each cloud a value δi which is calculated based on four factors (i.e., availability, storage, bandwidth, and operation prices) to indicate the preference of a cloud. We choose the most preferred n clouds, and then heuristically exchange the cloud in the preferred set with the cloud in the complementary set to search better solution. This is similar to the idea of Kernighan-Lin heuristic algorithm 1, which is applied to effectively partition graphs to minimize the sum of the costs on all edges cut. The preference of a cloud is impacted by the four factors and they have different weights. The availability is the higher the better, and the price is the lower the better. So we use δi = αai + β/ Pi as the preference of the ith cloud, where Pi is the synthetically price of storage, bandwidth, and operation. Intuitively, if a file has much read access, the cloud with lower bandwidth price is more preferred.

Out-going bandwidth is more expensive than storage, so we have to make sure that the cost of transition can be earned back by the new storage mode. That is, the following inequality has to be met: Mf >Mp + T, where, Mf and Mp are the monetary cost of the previous storage mode and new storage mode respectively. They are both calculated using the read frequency provided by Predictor. The above equation is impacted by the time period t. Since the storage cost is storing a file of size S for a time period t and Cr is the read count during t, we should set t first in order to calculate Mf and Mp. The total probability that there are k simultaneously available clouds can be expressed below:

$$Pr(N',k) = \sum_{j=1}^{\binom{|N'|}{k}} [\prod_{i\in C_j^{|N'|,k}} a_i \prod_{i\in N'\backslash C_j^{|N'|,k}} (1-a_i)] \quad (1)$$

Since the storage mode (m, n) can tolerate any 0 = (n<m) simultaneously failed clouds, its availability can be expressed as the sum of Pr(N', k). The storage cost can be expressed below:

$$\sum_{i=1}^{N} \frac{S}{m} P_{si} u_i = \sum_{i\in N'} \frac{S}{m} P_{si} \quad (2)$$

The normal read access does not need data decoding. Thus, the bandwidth and operation cost can be defined by rate of data flow in the network. CHARM, which guide customer to distribute their data on different clouds which is cost effective. The storage mode table can be calculated in advance, because it is only affected by the available clouds, their pricing policies, and availabilities. When deciding the storage mode for each file, we use the read frequency and the size of the file to look up the table for the corresponding storage mode. This table is re-calculated through Algorithm 1, only when availabilities and prices are modified, some clouds are kicked out due to performance issue, or new available clouds emerge.

**Algorithm 1:** Heuristic Algorithm of Data Replacement

**Input:** file size S, read frequency Cr, n's upper limit ξ

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

```
Csm ← inf; Ψ ← {}
   Ls ← sort clouds by normalized αai + β/Pi from
high to slow
   for n = 2 to ξ do
      Gs ← the first n clouds of Ls
      Gc ← Ls - Gs
      for m = 1 to n do
      Acur ←  calculate the availability of Gs
      ifAcur ≥ A then
      Ccur ← calculate the minimal cost
      if Ccur<Csmthen
      Csm ← Ccur
             Ψ ← Gs
           end
      else
         /*heuristically search better solution*/
      Gs ← sort Gs by ai from low to high
      Gc ← sort Gc by Pi from low to high
      fori = 1 to n do
          flag ← 0
      for j = 1 to N - n do
      if a_Gc[j]>a_Gs[i]then
            swap Gs[i] and Gc[j]
            flag ← 1
      break
      end
      end
        if flag = 0 then
       break
        end
```

```
            Acur ← calculate the availability of Gs
            **if** Acur ≥ A **then**
            Ccur ← calculate the minimal cost
            **if** Ccur<Csm**then**
            CsmCcur
                Ψ ← Gs
            **end**
            **break**
            **end**
             **end**
        **end**
     **end**
   **end**
            **return**Csm, Ψ
```

## IV. Prototype Experiments

We implemented the prototype experiments on four mainstream commercial clouds: Amazon S3, Windows Azure, Google Cloud Storage and Aliyun OSS, and pick 10 different data centers 2 from them. We created accounts in the four clouds, and replayed Amazing Store trace and Corsair trace for a whole month. In order to make sure the experiments can be finished in one month, we scale down the traces on the premise of correctness.

We also run simulations for the same traces to contrast with the real-world experiments. For the prototype experiments, CHARM performs out RepGr and EraGr by 21.7% and 27.8% for Amazing Store trace, and similarly the savings are 37.8% and 13.8% for Corsair trace, which proves the efficacy of CHARM. Equally in the cloud importantly, the prototype experiments show similar results as the simulation results, which proves the correctness of our simulations. When the price adjustment occurs, CHARM re-calculates the storage mode table, and uses the new table to store data and implement transition. The cloud we choose for price decrease is not used by the five schemes. That is to say this cloud has relatively higher price before price adjustment.

## V. Related Work

With the blossom of cloud services, there is a recent interest in addressing how to migrate data and applications into clouds seamlessly [7], the system designed in migrates Network File System (NFS) into the cloud, and meanwhile makes it feel like working locally. A similar work in [8] proposes a hybrid cloud-based deployment, where enterprise operations are partly hosted on premise and partly in the cloud.

Based on the measurements and analysis of six state-of-the-art cloud storage services, we unravel the key impact factors and design choices that may significantly affect the traffic usage efficiency. Most importantly, we provide guidance and implications for both service providers and end users to economize their sync traffic usage. There is a common concern that moving data into a single could would incur vendor lock-in risk. So many works propose storage architectures and mechanisms based on multiple clouds. Dura Cloud provides a convenient service to move content copies into the cloud, and store them with several different providers, all with just one click.

A similar work to ours is CAROM, which replication and erasure coding in multiple data centres. But it does not consider the heterogeneity of multi-cloud and the selection of clouds. They design a cache in the primary data centre. When a file is swapped out, this file is stored using erasure coding across multiple data centres. When this file is accessed again, it will be stored back to the cache. This scheme is efficient for the trace used in their paper. However, its performance relies on the characters of the targeted trace, more specifically, the cache hit rate (about 90% for their trace). For Amazing Store trace, the hit rate is only 44.7% with the cache size of 2TB. Frequent data swap inevitably induces much additional monetary cost to CAROM, which makes it even not competitive compared with the greedy data hosting schemes.

## VI. Conclusion

Cloud services are experiencing rapid development and the services based on multi-cloud also become prevailing. One of the most concerns, when moving services into clouds, is capital expenditure. So, in this paper, we design a novel storage scheme CHARM, which guides customers to distribute data among clouds cost-effectively. CHARM work presents the first tool, CloudCmp, to systematically compare the performance and cost of cloud providers along dimensions that matter to customers. CHARM makes fine-grained decisions about which storage mode to use and which clouds to place data in. The evaluation proves the efficiency of CHARM.

## References

[1] “Aliyun OSS (Open Storage Service),” http://www:aliyun:com/product/oss.

[2] “Gartner: Top 10 cloud storage providers,”www:networkworld:com/news/2013/010313-gartner-cloud-storage-65459:html?page=1.

[3] 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.

[4] 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.

[5] C. M. M. Erin Allen, “Library of Congress and DuraCloud Launch Pilot Program Using Cloud Technologies to Test Perpetual Access to Digital Content,” The Library of Congress, News Releases, http://www:loc:gov/ today/pr/2009/09-140:html.

[6] J. S. Plank, “Erasure Codes for Storage Systems: A Brief Primer,” The Usenix Magazine, vol. 38, no. 6, pp. 44–50, 2013. J. S. Plank, “Erasure Codes for Storage Systems: A Brief Primer,” The Usenix Magazine, vol. 38, no. 6, pp. 44–50, 2013.

[7] B. Trushkowsky, P. Bod´ık, A. Fox, M. J. Franklin, M. I. Jordan, and D. A. Patterson, “The SCADS Director: Scaling a Distributed Storage System Under Stringent Performance Requirements,” in FAST. ACM, 2011.

[8] M. Hajjat, X. Sun, Y.-W. E. Sung, D. Maltz, S. Rao, K. Sripanidkulchai, and M. Tawarmalani, “Cloudward Bound: Planning for Beneficial Migration of Enterprise Applications to the Cloud,” 2010