A cost-efficient information providing scheme with secured accessibility in heterogeneous cloud

D. Vanajakumari¹, B. Rama Ganesh²

¹M.Tech Scholar, department of CSE, JNTUA, VEMU Institute of technology P. Kothakota, Chittoor, India.
²Associate Professor, department of CSE, JNTUA, VEMU Institute of technology P. Kothakota, Chittoor, India.

Abstract — More and more enterprises and organizations are hosting their data into the cloud, in order to reduce the IT maintenance cost and enhance the data reliability. However, facing the numerous cloud vendors as well as their heterogeneous pricing policies, customers may well be perplexed with which cloud(s) are suitable for storing their data and what hosting strategy is cheaper. The general status quo is that customers usually put their data into a single cloud (which is subject to the vendor lock-in risk) and then simply trust to luck. Based on comprehensive analysis of various state-of-the-art cloud vendors, this paper proposes a novel data hosting scheme (named 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. We evaluate the performance of CHARM using both trace-driven simulations and prototype experiments. The results show that compared with the major existing schemes, CHARM not only saves around 20% of monetary cost but also exhibits sound adaptability to data and price adjustments.

Keywords—Multi-cloud, data hosting, cloud storage, cryptography, CHARM, OSS, RASS

I. INTRODUCTION

Recent years have witnessed a “gold rush” of online data hosting services (or says cloud storage services) such as Amazon S3, Windows Azure, Google Cloud Storage, Aliyun OSS, and so forth. These services provide customers with reliable, scalable, and low-cost data hosting functionality. 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 (including the hardware, software, and operational cost) and enhance the data reliability. 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. Now they only have to pay for exactly how much they have used. Heterogeneous clouds.

Existing clouds exhibit great heterogeneities in terms of both working performances and pricing policies. Different cloud vendors build their respective infrastructures and keep upgrading them with newly emerging gears. They also design different system architectures and apply various techniques to make their services competitive. Such system diversity leads to observable performance variations across cloud vendors.

Moreover, pricing policies of existing storage services provided by different cloud vendors are distinct in both pricing levels and charging items. For instance, Rack space does not charge for Web operations (typically via a series of Restful APIs), Google Cloud Storage charges more for bandwidth consumption, while Amazon S3 charges more for storage space.

Facing numerous cloud vendors as well as their heterogeneous performances /policies, customers may be perplexed with which cloud(s) are suitable for storing their data and what hosting strategy is cheaper. The general status quo is that customers usually put their data into a single cloud and then simply trust to luck. This is subject to the so-called “vendor lock-in risk”, because customers would be confronted with a dilemma if they want to switch to other cloud vendors. The vendor lock-in risk first lies in that data migration inevitably generates considerable expense. For example, moving 100 TB of data from Amazon S3 (California datacenter) to Aliyun OSS (Beijing datacenter) would consume as much as 12,300 USD. Besides, the vendor lock-in risk makes customers suffer from price adjustments of cloud vendors which are not uncommon. For example, the fluctuation of electricity bills in a region will affect the prices of cloud services in this region. We notice that giant cloud vendors like Windows Azure and Google Cloud Storage have been adjusting their pricing terms. Unexpected bankruptcy of cloud vendors further aggravates the situation. Nirvanix, which has thousands of customers including top 500 companies, suddenly shut down its cloud storage service in Sep. 2013. Ubuntu One, also a famous player in the market of cloud storage service,
escaped in Apr. 2014. So clearly, it is unwise for an enterprise or an organization to host all data in a single cloud — “your best bet is probably not to put all your eggs in one basket.” Finally, uncontrolled data availability is (in a sense) another type of vendor lock-in risk. Though the service quality is formally guaranteed by service level agreements (SLA), failures and outages do occur. Almost all the major cloud vendors experienced service outages. Some outages even lasted for several hours. Multi-cloud data hosting. Recently, multi-cloud data hosting has received wide attention from researchers, customers, and startups. 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 show in Fig. 1.

![Diagram of multi-cloud data hosting](image)

Fig. 1. Basic principle of multi-cloud data hosting.

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. DepSky guarantees data availability and security based on multiple clouds, thus allowing critical data (e.g., edical and financial data) to be trustingly stored. RACS deploys erasure coding among different clouds in order to prevent vendor lock-in risk and reduce monetary cost. Second, new types of cloud vendors have emerged and rapidly grown up to provide real services based on multiple clouds. Third, new development tools like Apache lib cloud provide a unified interface above different clouds, which facilitates migrating services among clouds. Nevertheless, as for multi-cloud people still encounter the two critical problems: (1) How to choose appropriate clouds to minimize monetary cost in the presence of heterogeneous pricing policies? (2) How to meet the different availability requirements of different services? As to monetary cost, it mainly depends on the data-level usage, particularly storage capacity consumption and network bandwidth consumption.

As to availability requirement, the major concern lies in which redundancy mechanism (i.e., replication or erasure coding) is more economical based on specific data access patterns. In other words, here the fundamental challenge is: How to combine the two mechanisms elegantly so as to greatly reduce monetary cost and meanwhile guarantee required availability? The proposed CHARM scheme. In this paper, we propose a novel cost-efficient data hosting scheme with high availability in heterogeneous multi-cloud, named “CHARM”. It intelligently uts 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 (for efficiently re-distributing data) by monitoring the variations of data access patterns and pricing policies. We evaluate the performance of CHARM using both trace-driven simulations and prototype experiments. The traces are collected from two online storage systems: Amazing Store and orsa, both of which possess hundreds of thousands of users. In the prototype experiments, we replay samples from the two traces for a whole month on top of four mainstream commercial clouds: Amazon S3, Windows Azure, Google Cloud Storage, and Aliyun OSS. Evaluation results show that compared with the major existing schemes of monetary cost but also exhibits sound adaptability to data and ice adjustments.

At last, our contributions in this paper can be briefly summarized as follows: 1) We propose and implement CHARM, a novel, efficient, and heuristic-based data hosting scheme for heterogeneous multi-cloud environments. CHARM accommodates different pricing strategies, availability requirements, and data access patterns. It selects suitable clouds and an appropriate redundancy strategy to store data with minimized monetary cost and guaranteed availability. 2) We design and implement a flexible transition scheme for CHARM. It keeps monitoring the variations of pricing policies and data access patterns, and adaptively triggers the transition process between different data storage modes. It also starts a data migration process among different clouds if necessary. 3) We evaluate the performance of CHARM using two typical real-world traces and prototype experiments. Both trace-driven simulation and experiment results confirm the efficacy of CHARM. Roadmap.
II EXISTING SYSTEM

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. For erasure coding, data is encoded into n blocks including m data blocks and nm coding blocks, and these blocks are put into n different clouds. In this case, though data availability can be guaranteed with lower storage space (compared with replication), a read access has to be served by multiple clouds that store the corresponding data blocks. Consequently, erasure coding cannot make full use of the cheapest cloud as what replication does. Still worse, this shortcoming will be amplified in the multi-cloud scenario where bandwidth is generally (much) more expensive than storage space.

DISADVANTAGES OF EXISTING SYSTEM

- Consequently, erasure coding cannot make full use of the cheapest cloud as what replication does.
- Still worse, this shortcoming will be amplified in the multi-cloud scenario where bandwidth is generally (much) more expensive than storage space.

III PROPOSED SYSTEM

In this paper, the system is proposed a novel cost-efficient data hosting scheme with high availability in heterogeneous 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 (for efficiently redistributing data) by monitoring the variations of data access patterns and pricing policies. We evaluate the performance of CHARM using both trace driven simulations and prototype experiments. The traces are collected from two online storage systems, both of which possess hundreds of thousands of users.

ADVANTAGES OF PROPOSED SYSTEM

- Replication mechanism when the file’s size is small. That is why gray level 4 puts its feet into the region of lower read count and smaller file size.
- This storage mode table only depends on prices of the available clouds and required availability. If the prices change, the table will change accordingly, becoming a different one.

IV IMPLEMENTATION

The framework of our proposed system has the accompanying modules alongside the following prerequisites.

- Multi-cloud
- Data hosting
- Cloud Storage
- Owner Module
- User Module

Multi-cloud:

Lots of data centers are distributed around the world, and one region such as America, Asia, usually has several data centers belonging to the same or different cloud providers. So technically all the data centers can be access by a user in a certain region, but the user would experience different performance. The latency of some data centers is very low while that of some ones may be intolerable high. CHARM chooses clouds for storing data from all the available clouds which meet the performance requirement, that is, they can offer acceptable throughput and latency when they are not in outage. The storage mode transition does not impact the performance of the service. Since it is not a latency-sensitive process, we can decrease the priority of transition operations, and implement the transition in batch when the proxy has low workload.

Data hosting:

In this section, we elaborate a cost-efficient data hosting model with high availability in heterogeneous multi-cloud, named “CHARM”. The architecture of CHARM is shown in Figure 3. The whole model is located in the proxy in this system. There are four main components in CHARM: Data Hosting, Storage Mode Switching (SMS), Workload Statistic, and Predictor. 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. 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. 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 very simple predictor, which uses the weighted moving average approach, works well in our data hosting model. 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. Then we illustrate how SMS works in detail in x V, that is, when and how many times should the transition be implemented.

Cloud Storage:
Cloud storage services have become increasingly popular. Because of the importance of privacy, many cloud storage encryption schemes have been proposed to protect data from those who do not have access. All such schemes assumed that cloud storage providers are safe and cannot be hacked; however, in practice, some authorities (i.e., coercers) may force cloud storage providers to reveal user secrets or confidential data on the cloud, thus altogether circumventing storage encryption schemes. In this paper, we present our design for a new cloud storage encryption scheme that enables cloud storage providers to create convincing fake user secrets to protect user privacy. Since coercers cannot tell if obtained secrets are true or not, the cloud storage providers ensure that user privacy is still securely protected. Most of the proposed schemes assume cloud storage service providers or trusted third parties handling key management are trusted and cannot be hacked; however, in practice, some entities may intercept communications between users and cloud storage providers and then compel storage providers to release user secrets by using government power or other means. In this case, encrypted data are assumed to be known and storage providers are requested to release user secrets. We aimed to build an encryption scheme that could help cloud storage providers avoid this predicament. In our approach, we offer cloud storage providers means to create fake user secrets. Given such fake user secrets, outside coercers can only obtained forged data from a user’s stored cipher text. Once coercers think the received secrets are real, they will be satisfied and more importantly cloud storage providers will not have revealed any real secrets. Therefore, user privacy is still protected. This concept comes from a special kind of encryption scheme called deniable encryption.

Owner Module:
Owner module is to upload their files using some access policy. First they get the public key for particular upload file after getting this public key owner request the secret key for particular upload file. Using that secret key owner upload their file.

User Module:
This module is used to help the client to search the file using the file id and file name. If the file id and name is incorrect means the user does not get the file, otherwise server ask the secret key and get the encryption file. If the user wants the decryption file means user have the secret key.

V 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 makes fine-grained decisions about which storage mode to use and which clouds to place data in. The evaluation proves the efficiency of CHARM.

VI REFERENCES


AUTHOR PROFILE

D Vanajakumari is currently a M.Tech Scholar in Department of Computer Science and Engineering, VEMU Institute of Technology, Chittoor. She completed her MCA from S V University Tirupati in 2005. Her current area of interest is java.

B Ramaganesh Received his M.Tech degree in Computer Science and Engineering, and B.tech IN Computer Science and Information technology both from JNTUH University, P.hd degree pursuing in vignan university, Guntur, where he is now an Assoc.prof &HOD in Department of Computer Science and Engineering, VEMU Institute of Technology, Chittoor(dt). He serves as Dean of Accademics of VEMU Institute of Technology during from 2012 onwards. His Research interests are in the areas of Software engineering, computer organization, DWDM Software project Management, and A.I. and he has 11 Years Teaching Experience in this field.