AN OPTIMAL MULTI-CLOUD DATA HOSTING WITH HIGH POSSIBILITY

Chaithra N Shetty[1], Vijay R[2], Bikash Thapa[3], Rikesh Shresta[4], Mrs Nethra M.V.O[5]

ABSTRACT: This paper is about a novel data hosting scheme which integrates two key functions desired, based on comprehensive analysis of various state-of-the-art cloud vendors. A company starts paying for consulting before even buying the software, and continues to pay for services (or hire IT staff) throughout the life of the software. The total cost of services for selection, implementation, maintenance and support ranges from 40% to 100% of the cost of software.The paper proposes and implement, a novel, efficient, and heuristic-based data hosting scheme for heterogeneous multi-cloud environment. It 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 with high possibility

INDEX TERMS: Data hosting; prediction; switching; multi-cloud;

INTRODUCTION:

Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centres. It relies on sharing of resources to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network. At the foundation of cloud computing is the broader concept of converged infrastructure and shared services.

With cloud computing, multiple users can access a single server to retrieve and update their data without purchasing licenses for different applications. In the shift from on-premises software to cloud computing, IT teams are still paying consulting and support fees that may be unnecessary. That's about to change. Businesses seem to have accepted that cloud computing leads to lower IT costs, and it's true that many companies experience some savings in switching to the cloud. Yet, the total cost of ownership for most is still up to 50% more than it should be. Here's why. Companies still assume that the software buying and deployment process is the same for cloud and on-premises products. They persist with the old model and end up paying for consulting and support services that are redundant or available for free.

The great variety of commercial cloud products ensures a better fit out-of-the-box, which means implementation costs go down. In fact, several cloud providers don't charge any implementation fees. Cloud apps are ready in days, sometimes hours. True cloud providers don't charge for maintenance and support. They operate on a subscription model and, because they want you as a long-term customer, they will probably spend time and effort to keep your company happy. Choosing the right provider helps avoid maintenance and support costs altogether.

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, The "proxy" component plays a key role by redirecting requests from client applications and coordinating data distribution among multiple clouds.

The system proposes and implements a novel, efficient, and heuristic-based data hosting scheme for heterogeneous multi-cloud environments. It accommodates different pricing strategies, availability requirement, and data access patterns. It selects suitable clouds and an appropriate redundancy strategy to store data with minimized monetary cost and guaranteed availability.

PROPOSED SCHEME:

In proposed system, the system proposes an economic data hosting scheme with high possibility in heterogeneous multi-cloud. It intelligently puts data into multiple clouds with minimized monetary cost and guaranteed availability.

The system combine the two widely used redundancy mechanisms specifically, i.e., replication and erasure coding, into a uniform model to meet the required availability in the presence of different data access patterns.

Then the system designs an efficient heuristic-based algorithm to choose proper data storage modes (involving both clouds and redundancy mechanisms). It 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. 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. 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.

Data Hosting and SMS are two important modules in this system. Data Hosting decides storage mode and the clouds that the data should be stored. We explain detail working in modules, that is, when and how many times should the transition be implemented.

MODULES: The major modules involved in implementing this system are:

- Data Hosting
- Storage Mode Switching (SMS)
- Workload Statistic
- Workload Predictor
- Performance Evaluation

Data hosting: Data Hosting decides storage mode and the clouds that the data should be stored in. We first formally define the mathematical model applied in Data Hosting. When talking about erasure coding, we usually mean $m > 1$ (not replication). However, replication is a special case of erasure coding (i.e., $m = 1$). So we combine the two storage mechanisms and define a unified model.

Storage Mode Switching: when the read frequency of the file drops below or increases above a certain value, changing storage mode can save more money. The value is determined by the prices of clouds. Given the available clouds including their prices and availability, we can figure out the storage mode and the selected clouds with the input of file's size and read count. However, it does not mean we should change the storage mode once a file's storage mode crosses the boundary, because the transition of storage mode also generates cost, which is definitely not negligible.

Workload statistics: 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.

Performance Evaluation: Different types of data may require different availabilities For example; backup data usually requires relatively low availability, while documents in work folders demand high availability. The experiments prove the effectiveness in this scenario, since it performs best for various availabilities. However, a more complicated use case is that the availability is varying with the access frequency of data.

For example," hot" data may demand high availability while "cold" data does not have that strict requirement. In order to show that it can also naturally adapt to this scenario, we run the two traces and assign different availabilities to the files according to their access frequency. More specifically, in our experiments 3 read requests a month is the boundary between high (99.99999%) and low (99.99%) availabilities. In order to avoid switching back and forth frequently, when the frequency drops below 1 request a month we change the availability from high to low, and when the frequency rises above 5 requests a month, the availability is changed from low to high. Setting these threshold values is reasonable since there is no strict boundary between different availabilities. The other schemes also use the same way to switch the availability.

System Architecture

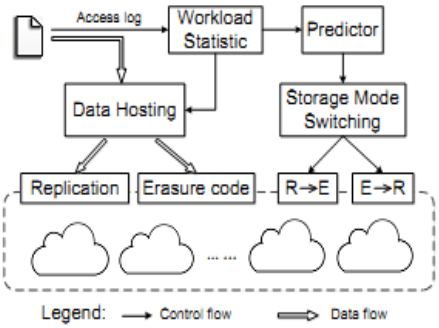
Heuristic algorithm of data placement

```
Input: file size S, read frequency c_r, n's upper limit ξ
Output: minimal cost C_sm, the set ψ of the selected clouds
C_sm ← inf; ψ ← {}
L_s ← sort clouds by normalized ωa_i + β/P_i from high to slow
for n = 2 to ξ do
    G_s ← the first n clouds of L_s
    G_c ← L_s − G_s
    for m = 1 to n do
        A_cur ← calculate the availability of G_s
        if A_cur ≥ A then
            C_cur ← calculate the minimal cost
            if C_cur < C_sm then
                C_sm ← C_cur
                ψ ← G_s
            end
        else
            /*heuristically search better solution*/
            G_s ← sort G_s by a_i from low to high
            G_c ← sort G_c by P_i from low to high
            for i = 1 to n do
                flag ← 0
                for j = 1 to N − n do
                    if a_{G_c[j]} > a_{G_s[i]} then
                        swap G_s[i] and G_c[j]
                        flag ← 1
                        break
                    end
                end
                if flag = 0 then
                    break
                end
                A_cur ← calculate the availability of G_s
                if A_cur ≥ A then
                    C_cur ← calculate the minimal cost
                    if C_cur < C_sm then
                        C_sm ← C_cur
                        ψ ← G_s
                    end
                    break
                end
            end
        end
    end
end
return C_sm, ψ
```

Working procedure for data hosting:

A user would select a file that has to be uploaded to the cloud server; user would also select the mode of storage by selecting the rack number and the data centre node. If the selected rack and data node has enough space to store the data, then the cloud server would allow you to store, else the cloud server response with a message that tells the space is not available. Same thing works with retrieval of file from cloud.

Data flow and control flow diagram of the scheme



REDUNDANCY MECHANISM

Replication: replicas are put into several clouds, and a read access is only served by the "cheapest" cloud that charges minimal for out-going bandwidth and GET operation.

Erasure coding: It is widely applied in storage system. It is used for reducing storage consumption. Data is encoded into n blocks including m data blocks and n-m coding blocks, and these blocks are put into n different clouds. Storage mode of 3 replicas. The data is lost when all 3 node crash.

CONCLUSION

This paper proposed a cost effective Multi-cloud data scheme for reduce IT maintenance cost. One of the most concerns, when moving services into clouds, is capital expenditure. This paper proposed this scheme for address that problem. It guides customers to distribute data among clouds cost-effectively. To reduce monetary cost, it applies this technique at the cloud storage level, which avoids vendor lock-in efficiently and reduces the cost of switching cloud vendors. And moves a step further that taking data access pattern into consideration, but it does not consider different redundancy mechanisms. Cloud Foundry is an open platform as a service, providing a choice of clouds, developer frameworks, and application services. It becomes much faster and easier to build, test, deploy, and scale applications. The other one is that operation cost in the prototype experiments is higher than that in the simulations. 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. The evaluation proves the efficiency of this scheme.

REFERENCES

"An Ensemble of Replication and Erasure Codes for Cloud File Systems", Yadi Ma, Thyaga Nandagopal, 2013.
"Rollerchain: a DHT for Efficient replication", Joao Paiva, 2013.
"NCCloud: Applying Network Coding for the Storage Repairing Cloud-of-Clouds", Yuchong Hu, Henry C. H. Chen, Patrick P. C. Lee, Yang Tang, 2012.
"SLA-based Admission Control for a Software-as-a-Service Provider in Cloud Computing Environments", Linlin Wu, Saurabh Kumar Garg, and Rajkumar Buyya, 2011.
"Decision Support for Application Migration to the Cloud: Challenges and Vision", Vasilios Andrikopoulos, Steve Strauch, Frank Leymann, 2013.
"Evolution of Cloud Storage as Cloud Computing Infrastructure Service" , R. Arokia Paul Rajan1, S. Shanmugapriyaa, 2012.
"On the Feasibility of Data Loss Insurance for Personal Cloud Storage", Xiaosong Ma, 2014.
"XORing Elephants: Novel Erasure Codes for Big Data", Maheswaran Sathiamoorthy, 2013.

"Erasure Coding in Windows Azure Storage", Cheng Huang, Huseyin Simitci, Yikang Xu, Aaron Ogus, Brad Calder, Parikshit Gopalan, Jin Li, 2012.

"Heading off Correlated Failures through Independence-as-a-Service", Ennan Zhai†, Ruichuan Chen, David Isaac Wolinsky, Bryan Ford, 2014.