Research on Minimizing Payment Cost of Multiple Cloud Service Provider

Daimi Aafiya

Department of Computer Engineering, M.S. Bidve Engineering College, BAT University, Latur, India

Corresponding Author: aafiya.daimi@gmail.com

Available online at: www.isroset.org

Received: 15/Aug/2019, Accepted: 24/Aug/2019, Online: 31/Aug/2019

Abstract—Today's many businesses are shifting their workloads to cloud storage to save capital costs for building and maintaining hardware infrastructures and avoid the complexity of managing data centers. Cloud computing has become a popular commercial service. The CSP (Cloud Service Provider) provides data storage services in which includes Get and Put function, using this geographically distributed data centers around the world. Selection of different CSP's datacenters and cloud customers facing two challenges, first one is how to allocating data to the datacenters in worldwide to satisfy application Service level objectives (SLO) requirement which includes both data availability and retrieval latency and second is how to allocate reserve resources & data in the datacenters which belongs to different CSP to minimizing payment cost. Find out the solution of these challenges firstly used integer-programming techniques for handles cost minimization problems. Here used three techniques for reducing service latency and payment cost first multicast-based data transferring second coefficient based data reallocation and third is request redirection based congestion and using PPM-C (Prediction by Partial Matching-Cloud) data compression technique, this helps to reduce storage cost and data transfer computing time.

Keywords—Service Level Objectives, Cloud Computing, Resources Reservation, CSP, Payment Cost Minimization, PPM-C compression, and Data Availability.

I. INTRODUCTION

Cloud storage provides an infinite amount of storage space for clients to archive their data backups on a pay-as-you-go basis, so remote back-up is an Internet of dynamically scalable resources as a filling service. Because of this cloud services, you can now remotely archive data to a third-party cloud storage provider instead of managing the data center on its own. Deployment in cloud classified from the physical location model. More case studies show that the use of cloud storage to back up remote data. Apart from companies and government agencies, individuals can also use tools like Dropbox to store their personal data in the cloud. In particular, with the advent of smartphones, the storage resources of smartphones are limited, so you can use the Dropbox-like tool to transfer audio/video files to your smartphone.

However, a single CSP may not have data centers at all locations necessary for a global web application. In addition, the use of a single CSP can introduce a blocking problem for the data storage provider, in which a customer may not be free to change the optimal supplier due to prohibitive exchange costs. Storage providers charge clients by bandwidth (Transfer), data requests (Get / Put) and storage. The CSP provides various storage services such as Get and Put services help to geographically distributed data centers around the world. As with cloud computing, there are two main contributors, there are two aspects of cost optimization: cost optimization and suppliers cost optimization is carried out by users.

Now continuously increase the data volume and detail captured by the organizations, such as the Internet of Things (IoT), multimedia, and the rise of social media, released overwhelming data flow in the unstructured/structured format. Data creation is the record speed, named here as big data, and emerges as widely recognized trend Big-Data set mainly use three aspects such as a) Data stored on the cloud cannot be categorized into the regular relational databases. b) Data are numerous formats and c) Data is captured, processed them rapidly and generated. In addition, Big-Data is now used much application such as Healthcare centers, Engineering, Science, Finance, and many businesses. Advance data storage techniques and mining technologies enable saving increasing amounts of data, changes in the nature of an organization's data. Fig. I shows that the cloud storage service provider's overview. Most enterprises shift the load to cloud storage to save capital costs for maintenance and hardware infrastructure and avoid the complexities of managing data centers. Figure 1 shows the cloud storage system.
In this paper study about the Literature Review done, in section II, the Proposed Approach Modules Description, Mathematical Modeling. Algorithm and Experimental setup in section III and at final provide a Conclusion in section IV.

II. LITERATURE REVIEW

In this section discuss the literature review in detail about the multiple cloud storage providers:

In this article [1], they represent SPANStore, a key and value store that exports a unified view of storage services to geographically distributed data centers. To minimize the cost of application providers, they combine three key principles. First, SPANStore spans multiple cloud providers to increase the geographic density of data centers and minimize costs through the use of differences in prices between providers. Second, by evaluating the application workload with the right level of detail, SPANStore intelligently exchanges the large geo-distributed replication needed to achieve the latency goals, with the higher storage and distribution costs that this entails, to meet the requirements for fault tolerance and consistency. Finally, SPANStore minimizes the use of computer resources for tasks such as two-phase locking and data distribution that are required to provide a global view of the storage services on which it is based.

To solve the configuration nightmare, they used MINERVA tools used to store data automatically. For data storage, MINERVA used declarative specifications of device capabilities and application requirements constraint-based formulations of the various sub-problems; and optimization techniques to explore the search space of possible solutions. This paper evaluated the design decisions that went into MINERVA, using specialized micro and macro-benchmarks [2].

Cloud services use Volley, sending the query logs of the data center. Volley analyzes logs using an iterative optimization algorithm based on customer data access patterns and locations and provides recommendations for migrating back to the cloud service. To scale to the volume of cloud service log data, Volley provides a scalable MapReduce scope-style platform that enables Volley to perform calculations of more than 400 machine hours in less than a day [3].

Cloud storage service provider provides the platform for storing large scale data on it with the minimum prize. Massive Parallel Processing (MPP) is using for bridges the gap between the modern cloud storage and traditional data warehouse. K. Liu, they implement Open Source Prototype GPCloud for Load/Upload data on the cloud storage this technique is based on cloud storage Amazon S3 and MPP data warehouse Greenplum [4]. An experimental result shows that the performance of this technique better than the existing system. This system is supported for INSERT and SELECT operation.

Resource provisioning to the computational task is the major challenge for cloud computing. Christoph Hoch Reiner and Stefan Schulte they propose prediction of cloud resources utilization on the pre-resource level and pre-task [5]. In this, they used machine learning techniques for prediction of resource utilization on the cloud storage. For this experiment, they used a dataset on the GitHub Travis CI and Travis CI. The performance of this system is compared with the simple learning regression approach and results show that the increase the accuracy of this system.

S. H. Gary Chan and Zhangyu Chan firstly study the problems of jointly optimizing resource allocation and video management for the large scale Video-on-Demand (VoD) cloud [6]. They propose RAVO (Resource Allocation and Video Management Optimization) model for jointly manage videos to achieve low cost and allocate system resources. For managing large video pool they used clustering algorithm. The performance of this system is compared with the other state-of-art techniques such as iGreedy, IPTV-RAM (internet protocol television-resources allocation and management) and super-optimum.

Cloud computing is a powerful technology for performing the operation such as complex and massive-scale computing. At this time the size of data increases and also verity of data generated and expanding every day. The use of a cloud service provider is to process, store and analyze data. Ibrar Yaqoob, Samee present classification techniques for big data, cloud service model and the conceptual view of data [7]. A review is conducting on the scalability, volume, data protection, availability, data transportation, data heterogeneity, regularity/legal issues, governance and data access.
Zhiming Shen, Qin Jia, they present techniques such as “Supercloud”, it is deployed the use of resources from the several cloud service providers includes Rackspace, Amazon EC2, and HP Clouds [8]. Superclouds enable organizations, businesses, and individuals in cloud computing environments. In particular, cloud users manage the live location and migration of their storage, computing, and networking without owning the entire underlying infrastructure.

Boyang Wang, Jiqiang Liu propose privacy in frequent itemset mining on the encrypted cloud [9]. In this, they use three protocols such as Protocol 1 achieving higher mining performance and Protocol 2 providing the strong privacy guarantee and Protocol 3 for improved efficiency. Mining performance is achieved separate form protocol 1 and protocol 3. Performance of system compared with the association rule mining and they also used a chess database for this experiment, it has totally 3196 transactions and 74 possible attributes. Performance of comparison carried out into two different security levels: Database privacy and Item Privacy.

Miguel Correia, Alysson Bessani, they present DEEPSKY’s methods for improving the privacy, integrity, and availability of information storing on the cloud and for this used encryption techniques for the data replica in a cloud. For achieving these objectives they used building Clouds of Clouds on the set of storage clouds and it also combines Cryptographic Secret Sharing with the Byzantine Quorum System Protocols [10].

Den Bossche, Jan Broeckhove develops techniques they also combine automated times series with the load prediction techniques which is based on the Double-Seasonal Holt-Winter [11]. Load forecasting with automated time series forecasting based on the Holt-Winters is the two-Season model it makes cost-effective procurement decision across a wide range of contract types, taking into account the organization’s current contract portfolio. They analyze their cost-effectiveness by modeling real traffic. This analysis explores the impact of different forecasting methods on cost versus the clairvoyant predictor and compares the performance of the algorithm on the stationary contract renewal approach. System results show that the algorithm can significantly reduce the cost of IaaS resources through automated procurement of reserved contracts.

Michael Borkowski, Christoph Hochreiner they are presenting techniques to the predicting use of the cloud resources at the task and resource level [12]. To do this, they use a machine learning prediction technique which is based on an extensive evaluation, they reduce the prediction error in 20% and improvements in this they achieved 89% accuracy.

S.H. Gary Chan, Zhangyu Chang they represent how to reduce the cost of deployment of optimizing video management like hosting and searching video on the server and resource allocation in the clouds such as processing, linking, and storage it is subject to a specific user to defer the requirement to access the video [13]. Firstly, they formulate the joint optimization problem and it is shown as NP-hard. For solving this problem, they offer Resource Allocation and Video Management Optimization (RAVO) which is based on linear programming with the proven optimality gap.

They proposed SCC(Storage Configuration Conclusion) to compile these inputs into cost-effective cluster configurations. This SCC technique is applied on different application workloads and storage options show that SCC captures enough detail to assign the right mix of storage and server hardware at the right scale; architecture change or decrease in magnitude leads to a significant decrease in performance. To meet application needs, SCC often predicts heterogeneous cluster architectures, resulting in significant cost savings compared to simply scaling homogeneous architectures [14].

NetPilot is a system that can automatically fix DCN failures.

This is a departure from the status quo, which largely depends on human intervention. This technique work in critical condition managing today's DCNs given the growing number of devices in these DCNs and the trend towards marketable equipment. NetPilot works by identifying the potential set of affected components that could cause the problem and iteratively taking mitigation measures aimed at each until the problem is resolved [15].

The present implementation of cops, a key-value store that delivers this model of consistency across the wide-area. A key contribution of COP is its scalability, which can provide cause-and-effect dependencies between keys stored across the cluster rather than on a single server, as in previous systems. The Central approach in COPS is to track and explicitly check whether the causal dependencies between keys in the local cluster are satisfied before providing the record. Also, in COPS-GT, introduce get transactions to get a consistent view of multiple keys without locking or locking. The experimental result shows that COPS completes operations in less than a millisecond, provides throughput similar to previous systems when using one server per cluster, and scales in the same way as for increasing the number of servers in each cluster. It also shows that the COPS-GT provides similar latency, throughput, and scaling for common workloads [16].

This paper proposes storage system architecture it distributes the data across Autonomous SSPs using informed hierarchical Erasure coding that, for a given replication cost, provides several additional 9's of durability over what can be achieved with existing black-box SSP interfaces, and it
performs an efficient end-to-end audit of SSPs to detect data loss that, for a 20% increase in the cost, improves the integrity of the data on two 9-by reducing the downtime of the network, and (3) offers durable storage with a price, performance and low availability with traditional storage systems. They create and evaluate these ideas by creating a SafeStore-Based file system with the NFS-like interface [17].

In this, they introduce a conductor, a system that frees cloud clients from the burden of deciding which services to use when deploying MapReduce computing to the cloud. Using Windows Explorer, customers specify only goals, such as minimizing cost or completion time, and the system automatically selects the best cloud services to use, deploys calculations according to that choice, and adapts to changing conditions during deployment. The guide design incorporates several new features, such as a system to manage the deployment of cloud calculations through various services, and a resource abstraction layer that provides a unified interface to these services, therefore hiding their low-level differences and simplifying calculation planning and deployment [18].

In this, they propose CALMS (Cloud-Assisted Live Media Streaming), a universal platform that facilitates migration to the cloud. Soothes adaptive leases and adjusts cloud server resources depending on the level of satisfaction of temporal and spatial dynamics of demands from live streaming users [19]. They also present the best solutions for working with cloud servers of different capacity and cost of rent, as well as potential delays in the initiation and termination of the lease in real cloud platforms.

III. PROPOSED APPROACH

A. Problem Statement
Develop the techniques to minimize the payment cost under the aforementioned constraints using data compression technique with PPM (Prediction by Partial Matching) this helps to reduce storage cost and data transfer computing time.

B. Proposed System Overview
Here, present a geo-distributed cloud storage system for Data storage; request Allocation and resource Reservation across the multiple CSPs in DAR. It transparently helps customers to minimize their payment cost while guaranteeing their SLOs. Building distributed cloud storage across multiple CSPs can avoid the vendor lock-in problem since a customer will not be constrained to an obsolete provider and can always choose the optimal CSPs for the cloud storage service.

A data allocation algorithm based on the dominant cost, which finds the dominant cost (storage, Get or Put) of each data element and assigns it to the data center with the minimum unit price of this dominant cost to reduce the cost of how to pay-as-you-go. An optimal resource reservation algorithm, which maximizes the cost of payment established by the payment reservation-like-you-goes while avoiding over the reservation.

![Proposed System Architecture](image)

Reallocation of data based on coefficients, it aims to balance the workloads among all of the billing periods in order to minimize the cost of payment by maximizing the advantage of the reservation. Transfer of data in multicast, which builds a tree of minimum expansion to create new replicas of data in order to minimize the cost of Transfer for the creation of replicas on a new deployment of data mapping.

Request redirection based congestion control, which redirects Get requests from overloaded datacenters to under loaded data centers that have received Gets more than (or less than) their expected number of Gets after data allocation to minimize the payment cost, respectively.

C. Algorithm

Algorithm: PPM-C (Prediction by Partial Matching - Cloud) Algorithm

Begin;
While (not last character) do;
Begin;
Readsymbol();
Shorten Context;
While (Context not found and context length -1) do;
   Begin;
   Output (escape sequence);
   Shorten context;
   End;
   Output (Character);
   While (Context length not -1) do;
   Begin;
   Increase count of character (Create a node of nonexistent)
   Shorten Context;
   End;
End;

- **PPM-C algorithm Compression working:**
In ppm only need to store the context that has occurred in the sequence which is being encoded. At the beginning of the encoding will need to encode the letters that have not occurred previously in the context. In the ppm use “escape symbol” <ESC> is used to signal that letter to be encoded has not been seen in the context.

In the basic algorithm have to keep in mind the following three points:
1. If the symbol has not occurred into the context then an escape symbol is encoded.
2. Attempt to use of next smaller context i.e reduce the size of symbol one by one.
3. Each time the symbol is count corresponding to that entire symbol is updated in all tables.

Consider the phrase to be encoding examples (“this is the wonderfull day”) of the given phrase have to make the encoding table until getting the higher order. For examples, higher order is 2 then making of the table from -1 order to 2nd order. To make the table need the fields like context, letter count and cum-count.

Where:

The letter is the letter in the word phrase.
Context is the symbol (letter) which is present before that letter for -1 and 0 order table there is no such field context. For 1st order: context is the symbol which comes before the corresponding letter.i.e. in the above example of the phrase is the context ‘t’ for letter ‘h’.
For 2nd order context= ‘th’ for the letter = i; Count = it means the number of time that letter present in the word before <ESC>
Cum-count = represents the addition of cum-count of previous letter and count of the current letter.

- Step1: in the initial state have to set the order of PPM model to ‘n’.
- Step2: Here consider the order of ppm is n that means highest context order is ‘n’. To achieve the highest order have to calculate the table of context order from -1 to n.
- Step3: Once done with the calculation of context table calculate the Lower order value (l) and Higher-order value (u) by using the following equations:
  \[ L^n = L^{n-1} + \left[ (u^{n-1} - l^{n-1} + 1) \times \text{cumcount} \right] / \text{total count} \]
  \[ u^n = L^{n-1} + \left[ (u^{n-1} - l^{n-1} + 1) \times \text{cumcount} \right] / \text{total count} \]
- Step4: Then assume the word length of the arithmetic coder to the length of a word. In arithmetic, coding tag lies between [0,1]. For example, if consider phrase is “this is” then word length is 6. That means the value of higher and lower order is 6 bit i.e (for example like 000010).
- Step5:
  I. For calculations of its letter consider 2nd order context table first and then determines whether the corresponding 1 context is encountered for that letter.
  II. If it encountered then take count and cum-count of that letter and find the value of lower
  III. Order and higher by using above equations and then check MSB of both values are same then transmit that MSB bit to the right-hand side and update the values of that letter in the 2nd context table.
  IV. Then move to 1st order context table. Note: if the context of the corresponding letter present then also move directly 1st order context table.
  V. Similarly, calculate lower order value and higher order value update order c table if the context is encountered in the 0th order context table. It will repeat step 3, 4, and 5 until reach the end of the phrase.

**D. Mathematical Model**

- **Input:** Customer datacenters denoted by \( D_c \)
  \[ D_c = \{ D_{c1}, D_{c2}, \ldots, D_{cn} \} \] (1)
- **Process I:**
  \[ C_T = \{ C_S + C_g + C_P \} \] (2)

The storage cost in a data center is the product of data size and unit storage price of each data center. Then, the total storage cost is calculated by:

\[ C_s = \sum_{t_k \in T} \sum_{d_t \in D_c} \sum_{j \in D_s} Ax_{p, j} \times p_{j} \times S_{d_t} \] (3)

Where \( S_{d_t} \) denotes the size of data,
\[ d_t \text{ and } P_{P_j}^d \text{ denote the unit storage price of datacenters } P_j \text{ and } X_{P_j}^{d_{t_k}} \text{ denotes the binary variable: it is } 1 \text{ if } d_t \text{ is store in } P_j \text{ during } t_k \text{; and } 0 \text{ otherwise.} \]

\[
C_c = \sum_{t_k \in T} \sum_{d_t \in D} \sum_{P_j \in D_s} q_{d_t}^c \times P_{P_j}^d \times S_{d_t}^c \quad (4)
\]

Where, \( p_{P_j}^0 \) is the cheapest unit transfer price of replicating \( d_t \) to \( P_j \) among all datacenters storing \( d_t \).

\[
C_g = \sum_{t_k} \sum_{P_j} \left( \max \left\{ \sum_{c_i} r_{c_i P_j}^k \times t_k - R_{P_j}^g, 0 \right\} + \alpha R_{P_j}^g \right) \times P_{P_j}^g \quad (5)
\]

\[
C_p = \sum \sum \left( \max \left\{ \sum_{\omega_{c_i P_j}^k} \times t_k - R_{P_j}^p, 0 \right\} + \alpha R_{P_j}^p \right) \times P_{P_j}^p \quad (6)
\]

The Get/Put billing is based on pay-as-you-go and reservation manners. The reserved number of Gets/Puts \( R_{P_j}^g \) and \( R_{P_j}^p \) is decided at the beginning of each reservation time period \( T \). The reservation prices for Gets and Puts are a specific percentage of their unit prices in the pay-as-you-go manner [7].

- **Process II:** To formulate the SLO objective, first need to calculate the actual percentage of Gets/Puts satisfying the latency requirement within the billing period \( t_k \)

\[
Q^g \times Q^p = 1 \quad (7)
\]

\[
\forall_c \forall_{t_k} \sum_{d_t} \sum_{P_j \in D_s} X_{P_j}^{d_{t_k}} \geq \beta \quad (8)
\]

Each customer datacenter maintains a table that maps each data item to its replica datacenters with assigned request serving ratios.

\[
\forall_{P_j} \min \left\{ \phi_{P_j}^g, \phi_{P_j}^p \right\} \geq 0 \quad (9)
\]

\[
\forall_c \forall_{t_k} \sum_{d_t} H_{C_{P_j}}^{d_{t_k}} \leq X_{P_j}^{d_{t_k}} \geq 1 \quad (10)
\]

\[
\forall_c \forall_{t_k} \forall_{d_t} \sum_{P_j} H_{C_{P_j}}^{d_{t_k}} = 1 \quad (11)
\]

Constraints (7), (8) and (9) satisfy the deadline requirement and data availability requirement in the SLO and the data center capacity constraint, as explained previously. Constraints (10) and (11) together indicate that any request should be served by a replica of the targeted data.

- **Output:**
  Data allocations and data storage \( D_s \)

\[ D_s = \{ D_{s_1}, D_{s_2}, \ldots, D_{s_n} \} \]

**IV. RESULTS AND DISCUSSION**

**A. Experimental Setup**

For this work, the required technologies:

- **Software Technology:**
  1. Operating System: Windows 7
  2. Coding Language: Java/J2EE
  3. Tools: NetBeans 7.2.1

- **Hardware Technology:**
  1. System: Pentium dual core
  2. Hard Disc: 120 GB.
  3. RAM: 1 GB.
  4. Input Device: Keyboard, Mouse

**B. Result Analysis:**

Table I show that data effectiveness and efficiency of SPAN-Store, DAR-C and PPM-C system. The data store cost of PPM-C is less than the SPAN-Store, DAR-C.

<table>
<thead>
<tr>
<th>Data compression in % Vs Data Size</th>
<th>DAR</th>
<th>100</th>
<th>100</th>
<th>100</th>
<th>100</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPM-PPMC</td>
<td>55</td>
<td>60</td>
<td>60</td>
<td>55</td>
<td>59</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4 shows the graph of effectiveness and efficiency DAR and DAR-PPMC system. The result of the graph shows the data compression in % of original data as found in stimulation observed that original data size is reduced up to approximately 44.3% of original data.

Table II shows that computing time required for data allocation of SPAN-Store, DAR-C, and PPM-C system. The PPM-C required computational times are less than the SPAN-Store, DAR-C.
Table 2: Computing Time Vs Data Size

<table>
<thead>
<tr>
<th></th>
<th>Computing Time Vs Data Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAR</td>
<td>65 75 85 95 105</td>
</tr>
<tr>
<td>DAR-PPMC</td>
<td>60 70 80 90 100</td>
</tr>
</tbody>
</table>

Below figure 4 shows that computing time comparison graph between DAR and DARPPM-C. As observed data size of original data is reducing approximately 40-44.3% of original data size by using PPM compression method with DAR. Hence while performing Get/Put operation if the size of data is less than why it requires less computing time for Get/Put operation.

![Computing Time on Data](image)

Fig. 4. Graph of Computing Time Vs Data in Bytes

Below figure 5 shows that the cost ratio of DAR and DAR-PPMC. DAR compared with and without PPM-c Compression method v/s of data size. This is because the PPM-C compression method reduces the data size to approximately 40-44.3% of the original data size. So that computing cost of data will be reduced.

![Computing Cost vs Data Size](image)

Fig. 5. Graph of Computing Cost Vs Data Size

V. CONCLUSION AND FUTURE SCOPE

The aim of this paper is to minimize the cost of the Cloud Storage Service (CSPs) providers with the guarantee of services used worldwide distributed data centers, which belong to the different CSPs with different resource unit prices. Here, firstly used an integer programming for cost minimizing problems. It is NP-hard problem; that why here introduce Data Storage, Request Allocation and Resource Reservation across the multiple CSPs (DAR) techniques for the Dominant Cost Based Data Allocation Algorithm to reduce the cost of each datacenter. Reducing service latency and payment cost datacenters used three methods 1. Multicast-Based Data Transfer, 2. Coefficient Based Data Allocation and 3. Request Redirection Based Congestion Control. The PPM data compression technique it helps to reduce storage cost and data transfer computing time.

ACKNOWLEDGMENT

The authors would like to thank the researchers as well as publishers for making their resources available and teachers for their guidance. Thankful to the authorities of Savitribai Phule University, Pune and concern members of the cPGCON-2019 conference, organized by, for their constant guidelines and support. Thankful to the reviewer for their valuable suggestions.

REFERENCES


