A User Centric Cost Effective Model For Dynamic Content Distribution In Cloud Environment

Aluri Shiva Kumar¹, Dr Pankaj Kawad Kar², Dr. N.Rajasekhar³

¹Research Scholar, Dept. of Computer Science & Engineering, Sri Satya Sai University of Technology & Medical Sciences, Sehore, Bhopal Indore Road, Madhya Pradesh, India

²Research Guide, Dept. of Computer Science & Engineering, Sri Satya Sai University of Technology & Medical Sciences, Sehore, Bhopal Indore Road, Madhya Pradesh, India

³Professor, Research Co-Guide, Department of Information Technology, Gokaraju Rangaraju Institute of Engineering & Technology (Autonomous), Hyderabad

ABSTRACT

Cloud providers give storage access and efficient content placement and delivery services to content providers by optimizing cloud-based content delivery. The cost-efficient model ought to consider the content delivery cost as well as the storage cost associated with the cloud network. In this article, a novel cloud-based content delivery model is proposed that uses shared storage models for cost optimization in content delivery. With the recent advent of cloud computing technologies, a developing number of content conveyance applications are contemplating a change to cloud-based services, for better versatility and lower cost. Two key tasks are involved for such a move: to migrate the contents to cloud storage, and to distribute the web service load to cloud-based web services.

1. INTRODUCTION

Cloud computing technologies have enabled quick provisioning and release of server utilities (CPU, storage, bandwidth) to users anywhere, anytime. To exploit the diversity of electricity costs and to provide service closeness to users in different geographic regions, a cloud service often ranges multiple data centers over the globe, e.g., Amazon Cloud Front, Microsoft Azure, Google App Engine. The elastic and on-demand nature of resource provisioning has made cloud computing attractive to providers of different applications. More and more new applications are being created on the cloud platform, while many existing applications are likewise considering the cloud ward move, including content dispersion applications. As a significant category of mainstream Internet services, content circulation applications, e.g., video streaming, web facilitating and file sharing, feature large volumes of contents and demands that are exceptionally unique in the temporal area.

A cloud platform with multiple, distributed data centers is ideal to host such a service, with generous advantages over a traditional private or public content dissemination based arrangement, in terms of more readiness and
critical cost reduction with respect to machines, bandwidth, and management. Along these lines, the application providers can zero in their business more on content provisioning, rather than IT infrastructure maintenance. Two significant components exist in a commonplace content appropriation application, namely backend storage for keeping the contents, and front-end web services to serve the requests. Both can be migrated to the cloud: contents can be stored in storage servers in the cloud, and requests can be distributed to cloud-based web services.

Figure 1. Cloud Computing Services Overview

Therefore, the key challenge for cloud-ward move of a content dispersion application is the manner by which to efficiently replicate contents and dispatch requests over multiple cloud data centers, just as the provider's existing private cloud, with the end goal that good service response time is guaranteed and just modest operational expenditure is incurred. It may not be too difficult to even think about designing a simple heuristic for dynamic content placement and load appropriation in the hybrid cloud; however, proposing one with guarantee of cost optimality over a since quite a while ago run of the system, is a fascinating yet scaring challenge, especially when self-assertive appearance rates of requests are considered. Some existing work have advocated ideal application movement into clouds, however none spotlight on guaranteeing over-time cost minimization with a unique algorithm.
II. COST AWARE RESOURCE PROVISIONING IN CLOUD

The resource provisioning innovation encourages a simple and direct resource portion measure in the cloud environment. With an effective resource provisioning innovation, the clients consider the resource type, renting period and cost for their applications. As the resource types depend on the application qualities, the clients give more significance to the renting time frame and cost. Further, the clients hope to run their applications with insignificant expense. In the event that the application isn't time basic, the clients may endure some time delay for more cost reserve funds. Various applications utilize diverse expense and distinctive execution enhancement strategies. These methods can be utilized for both the single cloud supplier and numerous cloud supplier.

In light of the entertainers included the cost enhancement methods can be arranged into two kinds, for example, improvement performed by the suppliers and advancement performed by the clients. The cost advancement performed by the cloud suppliers primarily focus on decreasing the expense to keep up the physical server farm. Mostly by lessening the force utilization, the cost advancement is performed. The cost improvement performed by the cloud clients pick the right harmony between the sorts of the occurrences.

On account of multiuser environment, a successful valuing model and honest instrument are intended to plan the resources in a savvy way. Further, the issues of the cloud agent in different cloud supplier environment are investigated. In this exploration, to limit the scheduling delay and the expense of the resource provisioning measure, a proficient Cost Energy Aware Nash Equilibrium (CEANE) model is proposed. The current techniques center around lessening either the expense or the vitality, yet the proposed model spotlights on the decrease of both the vitality and the cost usage for the cloud resource provisioning measure.

Cost Estimation in Cloud Environment

Since cloud computing utilizes on-request evaluating, it is imperative to ascertain the expense of keeping up IT infrastructure in house. Despite the fact that numerous creators propose more refined cost estimation model for cloud computing (Stuer et al., 2007; Hosanagar et al., 2004; Abramson et al., 2002), on-request valuing would in any case have its omnipresent nearness in totally cost computation strategies. This segment investigates different costs associated with in-house the board of IT infrastructure which is independent of a specific venture necessity. For the majority of the parts associated with this area, the idea of complete expense of possession (TCO) is utilized. TCO is the methods for tending to the genuine expense ascribing to claiming and overseeing IT infrastructure (Cappuccio et al., 1996). It extensively considers the whole lifetime spending, capital costs, cost of tasks and thus is reasonable for base cost assessment. An aggregate of nine parts sanctuary been considered in base cost assessment including amortization, cost of workers, arrange cost, power cost, software cost, cooling cost, land cost, office cost and backing and support cost. For every segment, the accompanying subtleties are given: a) clarification of the apparent multitude of factors included and b) the strategy to figure the expense of the segment. The general point is to concocted month to month costs for all the segments being thought of and subsequently all factors are changed over to month to month parameters.
III. AMORTIZATION

It is critical to comprehend the commitment of IT infrastructure expenses to the month to month rental structure in an association. Thus, amortization parameter is determined for servers and different offices so reasonable attribution of expenses for different IT resources (equipment/software) can be achieved. This parameter is needed to figure the month to month deterioration cost (amortization cost) of every infrastructure thing being thought of. These things have starting buy cost, the expense of which is determined dependent on the length over which the speculation is amortized at expected loan fee. Studies have uncovered that the expense of CPU, stockpiling and data transmission generally twofold when the expenses are amortized over the lifetime of the infrastructure.

The financing cost is commonly 5% per annum (Greenberg et al., 2008) and the devaluation time of land is ten years while that of servers/different offices is three years (Hamilton, 2009). When the amortization parameter is gotten, it tends to be at that point utilized in the estimations of expected segment to get the month to month cost. The amortizable parameter for office (Ap_F) is registered uniquely in contrast to that of server (Ap_S) inferable from the distinctive amortization periods. The loan fee is spoken to as Cost of Money (Com) and is kept in factor structure (rather than 5%) to oblige any changes. Ap_F is determined as (Com/(1-power ((1+Com),(- 1 * Time_F))), where Time_F is the office amortization period and estimated in months. Essentially, Ap_S is determined as (Com/(1-power ((1+Com),(- 1 * Time_S))), where Time_S is the server amortization period and estimated in months.

IV. COST OF SERVERS

Servers are commonly mounted on racks and it is expected that all the servers have comparable setups. This supposition that is made to facilitate the calculation for cost of the server (without amortization). Consequently, Cost_S can be registered as (N_S * Cost_PS), where N_S is the quantity of servers in a firm and Cost_PS is the expense per server in Dollars. The amortizable parameter for server determined in the past part will be utilized to decide the amortized server costCost_Am_S. It very well may be determined as Cost_Am_S=(Cost_S * Ap_S), where Ap_S is Amortizable Parameter for Server from past sub-area. The costs other than base expense related with the acquisition of the server have been determined independently.

V. NETWORK COST

The segments that add to the systems administration costs are NIC, switches, ports, links, software and upkeep. The expense of NIC is as of now credited in the server cost while that of the software will be taken up in the software cost segment. Upkeep exercises have additionally been taken up independently in type of Support and Maintenance Cost. Henceforth, this area would just arrangement with the expense of switches, ports, links and the usage costs. Since cost related with systems administration again has an underlying cost, it is amortized to think of the month to month cost.
The complete systems administration cost (Cost_Net) is an aggregate of Cost of Port (Cost_Port), Cost of Cable (Cost_Cab), Cost of Switch (Cost_Switch) and execution cost (Cost_Imp). All expenses are estimated in USD and determined utilizing the accompanying conditions:

\[ \text{Cost}_\text{Port} = \text{N}_\text{Port} \times \text{Cost}\_\text{per}\_\text{Port} \]
\[ \text{Cost}_\text{Cab} = \text{N}_\text{Cab} \times \text{Cost}\_\text{per}\_\text{Cab} \]
\[ \text{Cost}_\text{Switch} = \text{N}_\text{Sw} \times \text{Cost}\_\text{per}\_\text{Sw} \]
\[ \text{Cost}_\text{Net} = \text{Cost}_\text{Port} + \text{Cost}_\text{Cab} + \text{Cost}_\text{Switch} + \text{Cost}_\text{Imp} \]

However, networking costs should be amortized to calculated the amortized networking cost represented by Cost_Am_Net and given by

\[ \text{Cost}_\text{Am}_\text{Net} = \text{Cost}_\text{Net} \times \text{Ap}_S \]

VI. PROPOSED METHODOLOGY

This paper about a regular substance conveyance application, which gives an assortment of substance (records), meant as set M, to clients spreading over various geological locales. There is a private cloud possessed by the provider of the substance circulation application, which stores the first duplicates of the apparent multitude of substance. The private cloud has a general upload data transfer capacity of b units for serving substance to clients. There is a public cloud comprising of server farms situated in various topographical districts, indicated as set N. One server farm dwells in every district. There are two kinds of between associated servers in every server farm: stockpiling servers for information stockpiling, and computing servers that help the running and provisioning of virtual machines (VMs). Servers inside a similar server farm can get to one another through a certain CN (Data Center Network).

The provider of the substance dissemination (application provider) wishes to arrangement its service by abusing a mixture cloud architecture, which incorporates the geo-conveyed public cloud and its private cloud. The significant parts of the substance dissemination application include: (I) back-end stockpiling of the substance and (ii) front end web service that serves clients' solicitations for substance. The application provider may move both service segments into the public cloud: substance can be imitated away servers in the cloud, while missions can be dispatched to web services introduced on VMs on the computing servers. A representation of the framework architecture is given in Fig. 1. Our target in this paper is to plan a dynamic, ideal algorithm for the application provider to deliberately settle on the accompanying choices for service movement into the crossover cloud architecture: (I) content replication: which substance ought to be duplicated in which server farm at each time? (ii) demand appropriation: what number solicitations for a substance ought to be coordinated to the private cloud and to every one of the server farms that store this substance at that point?
VII. COST-MINIMIZING SERVICE MIGRATION PROBLEM

In the event that framework runs in a period opened style. Each schedule opening is a unit time which is sufficient for uploading any document \( m \in M \) with size \( v(m) \) (bytes) at the unit data transfer capacity. In time allotment \( t \), \( a(m)_j(t) \) demands are produced for downloading record \( m \in M \), from clients in locale \( j \). We expect that the solicitation appearance is a self-assertive cycle after some time, and the quantity of solicitations emerging from one district for a document in each time allotment is upper-limited by \( A_{\text{max}} \).

The expense of uploading a byte from the private cloud is \( h \). The charge for capacity at server farm \( I \) is \( p_i \) per byte per unit time. \( g_i \) and \( o_i \) per byte are charged for uploading from and downloading into server farm \( I \), individually. The expense for leasing a VM case in server farm \( I \) is \( f_i \) per unit time. These charges follow the charging model of driving business cloud providers, for example, AmazonEC2 and S3. We accept that the capacity limit in every server farm is adequate for putting away substance from this substance circulation application. We likewise expect that each solicitation is served at one unit transmission capacity, and the quantity of solicitations that a VM in server farm \( I \) can server unit time is \( r_i \). The document can be completed in equal: subsequent to accepting a little bit of the record, a server farm would already be able to begin to serve the got lumps of the record to clients. We accept that upload transfer speed is held for reproducing records to server farms from the private cloud, and this data transmission isn't included in \( b \), the most extreme units of data transfer capacity that the private cloud can use to upload substance to clients. Not all solicitations emerging in one schedule opening are dispatched in a similar time allotment, subject to limit limitations.

Service quality. The service quality experienced by clients is assessed according to popular demand reaction delay, comprising of two significant segments: queueing delay in the solicitation line, and full circle delay from when the solicitation is dispatched from the line to the time the primary byte of there quested record is gotten. We disregard the preparing delay inside a server farm, because of the high between association transfer speed and CPU limits inside a server farm. Let \( d_j \) and \( e_j \) signify the full circle delay between locale \( j \) and the private cloud, and between district \( j \) and datacenter \( I \), separately. Let \( _\mu \) be the upper-bound of the normal full circle delay per demand, which the application provider wishes to implement in this substance dissemination application. We sensibly expect \( _\mu > e_{ii} \) \( \forall i \in N \), i.e., this bound is bigger than the full circle delay between a client and the server farm in a similar area. We will show that our dynamic ideal service relocation algorithm can bound both the normal full circle delay and queueing defer experienced by clients.

Our algorithm centers around limiting repeating operational expense of the substance circulation framework, not one-time costs, for example, the acquisition of machines in the private cloud and substance.
VIII. CONCLUSION

In this paper, ideal relocation of a substance dissemination service into a crossover cloud comprising of a private cloud and public cloud services. We are utilizing the Lyapunov improvement hypothesis which is limits the operational expense of the application with Quality of service ensures. Lyapunov enhancement gives a structure to planning algorithms with execution self-assertively near the ideal presentation over a since quite a while ago run of the framework, without the requirement for any future data. We propose a conventional streamlining system for dynamic, ideal relocation of a substance dispersion service to a cross breed cloud comprising of a private cloud and public geo-conveyed cloud services.

REFERENCES