INTERNATIONAL JOURNAL OF TECHO-ENGINEERING

ISSN : 0975-4520

Minimizing cost in Hybrid cloud using Content Distribution Services

Y. SANTHOSH KUMAR[1], Dr.V.JANAKI [2]

[1]MTech student, Dept. of CNIS, Vaagdevi College of Engineering, Warangal, T.S

[2]Professor, Dept. of CSE, Vaagdevi College of Engineering, Warangal, T.S

Abstract: *The latest emergence of cloud computing technologies has provided agile and scalable access to useful resources for application branching. As a result, content delivery offerings are a major category of popular Internet software. As a result, an increasing number of content carriers are considering moving to fully cloud-based services to increase scalability and lower price. There are two main obligations involved in any streaming: migrate your content to cloud storage, and distribute your load from Internet services to fully cloud-based web services. The primary mission is to leverage the cloud on a global scale and the existing on-premises server infrastructure to serve requests for variable content material with the guaranteed response time from the provider at all times while at the same time incurring minimal operating fees. By utilizing Lyapunov's optimization strategies, we provide an optimization framework for dynamic migration and reduce the price of content delivery services to a hybrid cloud infrastructure that covers geographically dispersed stats centers. Active control rule set, optimal content placement, and order placement in reliable data facilities are designed to reduce average operating cost over time, challenging vendor reaction time constraints. The careful evaluation shows that the algorithm correctly identifies interaction instances within a predefined QoS target in cases of arbitrary request arrival patterns and ensures that the total value falls within a small gap consistent with the most desirable T-slot prediction mechanism with future recognized records*.

Keywords: *Cloud computing, Quality of service, Lyapunov's optimization, Data centre, Cloud server.*

I. INTRODUCTION

Now the cloud computing era of the day is rapidly used to access assets for different applications. There are unique resource patterns: asset calculating and network asset (calculating resources that include memory, CPU, storage, networking assets, plus bandwidth). The activities of the cloud provider to use and customize the font occur at the time of the cloud environment. The type and support that each order was looking for were required to complete the consumer job. Asset allocation order and timing are also inputs for first-order asset allocation. In the proposed device, we use a hybrid cloud to use the public and private cloud. CDN (Content Delivery Network) while accessing the data you request from the cloud server and searching for files on the CDN [1]. The DCN (Data Content Network) provides an access key for the consumer to access the document. In the dynamic migration method, bloodless spots and warm spots are used. The hot spot method is used to distribute the load. This technology helps achieve load balancing and increases overall performance and productivity. The Lyapunov optimization method to reduce the price in the proposed system that we use in the strategies. Your need to dynamically update the rate when the number of consumers requests from the cloud server. The main objective of the proposed device is to reduce the operational cost over the years for the cloud transmitter and achieve load balancing, and providing protection. To provide security, we use a ciphertext coverage algorithm. Implement authentication technology to verify consumer authentication if the person has permission to access the services at that time, and if more efficient, send the configuration key to the application. User can access or used only key access pages. User cannot access or use other page [2].

II. RELATED WORK

At this stage, modern methods of migrating the provider between dedicated cloud environments are considered. This document falls within the scope of the related work that assumes systems that:

1. It can be integrated into a cloud infrastructure.
2. Providing an environment for hosting and relaying services.
3. Perform an optimization algorithm entirely based on display and cloud parameters.

System 1 [3] uses a so-called General Computing Platform (GCP) that hosts different types of offerings, from public service to infrastructure services. System 1 contains:

- A workflow version for configuring the services.
- A value model for deciding whether or not to migrate the services.
- Choosing the best location for them.

The services used in the workflow model are simpler abstract representations of real benefits. Therefore, during the selection procedure, the summaries are changed with specific times. The decision model uses a genetic algorithm to determine the optimal solution, that is, the large configuration of the workflow. System 2 [4] proposes migrating stateful web services based on Service Level Agreement (SLA) violations. The P2P overlay network is integrated via Web Services Resource Framework (WSRF) containers host the offerings.

With the WS Agreement for SLA, SLAs are described for each service. WSRF containers monitor SLA transactions and detect violations. Since individual funds are operated in unique environments, asset consolidation is performed using the function that normalizes all parameters to the best available rate. Then the container calculates the so-called health status using the health function and health scale. It beautifully defines other containers and represents the popularity of the domain. If the SLA clause is violated, the single box randomly selects a service and queries alternate containers through the P2P community to find a valid relay destination. The third system [5] builds Meta-Cloud with existing and abstract standardizations to ensure interoperability and avoid vendor lockout. The components of Meta-Cloud are Meta Cloud API (as a unified programming API for Internet programs), useful resource templates (for describing services and requirements), migration and deployment recipes (to allow automatic migration), Meta Cloud Proxy (as the intermediary between the application and the cloud sender), and monitoring Resources (to check QoS houses), Provisioning method (to adapt packages to service providers at runtime), Experience base (to collect all stats and parameters). The fourth system [6] proposes a version-based approach to designing and running applications in a cloud pair. Applications should be implemented more efficiently once the code is semi-routinely compiled for extracted clouds. The DSS then finds the first-class cloud of service related to price, threat, and impractical criteria. Finally, the runtime management API enables the dynamic migration of offers between the clouds. System 5 [7] offers a suite of tools that includes Service Builder (SB), Management Console (AC), Deployment Engine (DE), Service Optimizer (SO), and Cloud Optimizer (CO). These add-ons help service providers develop, configure, and run packages in private cloud environments. Also, monitor SLA parameters and allocate new assets or migrate the provider to any other cloud if necessary. A regular content distribution application has two main components, specifically a backstop

ISSN : 0975-4520

garage to hold content and incoming network offerings for service requests. Both can be migrated to the cloud - content can be stored on cloud servers, and requests can be funneled to fully cloud-based internet offerings. Hajat et al. developed an optimization model to migrate corporate IT applications to a hybrid cloud. Their model takes into account unique business constraints, such as transaction delays and protection guidelines. Zhang et al. advised an intelligent algorithm to determine the workload and dynamically locate the trigger in the public and non-public cloud. Chen et al. We recommend creating CDNs within the cloud to reduce the rate below the limits of QoS requirements.

III. PROPOSED METHODOLOGY

This paper introduces a regularly occurring optimization framework for dynamic migration that reduces the value of content delivery services directly in a hybrid cloud (such as public and private hybrid clouds). We design an algorithm set for content placement and load distribution that generally reduces operational value over the years, a problem for seller response time limitations. Our design is based on Lyapunov's optimization idea where minimum fees and response time are achieved simultaneously with the help of green scheduling for content relay and order submission. Lyapunov's optimization provides a framework for designing algorithms with an overall performance that is arbitrarily close to the final version over a longer device period, without the need for target statistics. It has been used especially in routing and channel assignment in wireless networks, and it has been more effective these days in addressing assisted allocation problems in some different types of networks. We adapt Lyapunov's optimization technologies to create a hybrid cloud to solve the main issues of dynamic and joint content reproduction and load distribution. We demonstrate the optimization of our algorithm through rigorous theoretical analysis. The rule set well defines service interaction states within a predefined QoS target in cases of arbitrary request arrivals and ensures that the total value is within a fixed small gap from the most appropriate end through a T-slot prediction mechanism with future data.

PROPOSED ARCHITECTURE

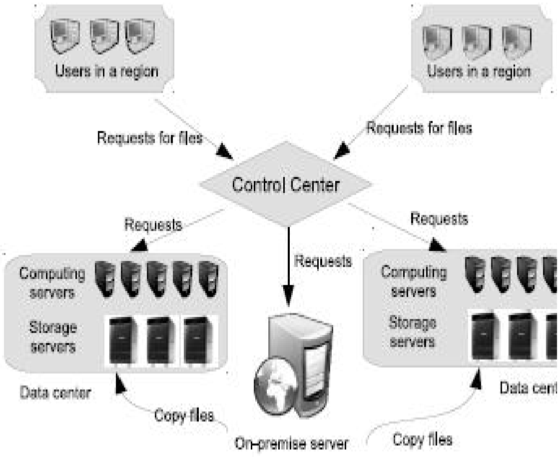

Fig.1 The system architecture

We consider a standard content delivery program, which provides a set of content (files), called an M set, to customers distributed in more than one geographic area. A local (or non-public cloud) server pool is owned by the content delivery software provider, which stores the original copies of all content. Without a lack of generality, we use a server to represent the server group on-premises. The local server has a normal extra bandwidth of b units to serve content to users. A public cloud and statistics center are located in more than one geographic region, referred to as an N cluster. There is a data center in each area. There are interconnected servers in every information center: storage servers for storing records and computer servers that direct the bios and provide virtual machines (VMs). The content delivery application provider (the utility company) must provide its provider by exploiting a hybrid cloud fabric, which includes a geographically distributed public cloud and its internal server. Key additions to the Content Delivery Tool include: (i) storing content back at the edge and (ii) a front-end Internet service that serves users' requests for content. In addition, the application company can migrate both extensions from the provider to the public cloud: contents can be copied to garage servers within the cloud, while requests can be sent to the Internet offers installed on virtual machines on computer servers. An example of the system architecture is shown in Figure 1.

The cost of uploading a byte from the on-premise server is h. The charge for storage at data center i is p_i per byte per unit time. $g_i \, and \, o_i$ per byte are charged for uploading from and downloading into data center i, respectively. The cost for renting a VM instance in data centre i is fi per unit time. These charges follow the charging model of leading commercial cloud providers, such as Amazon EC2 and S3 .We assume that the storage capacity in each data center is sufficient for storing contents from this content distribution application. We also assume that each request is served at one unit bandwidth, and the number of requests that a VM in data center i can serve per unit time is ri.

Cost Optimality

Comparison with Existing Algorithms: We first evaluate our dynamic algorithms against an easy myopic scheduling algorithm and any other set of existing rules for content placement and order routing for traditional CDN, called Maximum Weight Iterative Periodic Scheduling with Minimum Weight Evacuations (abbreviated as IPMW with MWE). Algorithm tactics schedule all requests within the time interval as they arrive without buffering them in queues and decide on material content redundancy and request distribution while reducing global operating cost (3) below constraints by changing all time-averaged expressions to those of generation Next from the particular time. Like our work, IPMW models the MWE algorithm, the front-end source nodes system, and the back-end cache (background cache is a synonym for the garage server area in this document) of the CDN as the input and output nodes. Create queues for unique document requests on provisioning nodes, make request routing selections in every appropriate slot, and periodically update back-end cache content. But unlike our algorithm, the cache scale is constant, which results in a trade-off between garage fees and queue delays. For the variance to be true, we perform a binary search for the most reliable size of your back-end cache that ends up with the lowest price, with the restriction that queue delays for more than 90% of requests fall within the given target. Another parameter configured for IPMW with MWE is periodicity while allowing content material to be refreshed from the background cache. To fair check, we set the periodicity to 1, the same as what is allowed to go for a walk with our dynamic set of rules.

Algorithm 1: Control Algorithm on the Control Center

Initialization:
Set up request queue $Q_j^{(m)}$, virtual queues G and $Z_j^{(m)}, \forall j \in \mathcal{N}, m \in \mathcal{M}$, and initialize their backlogs to 0;
In every time slot t:
1. Enqueue received requests to request queues ($Q_j^{(m)}$'s);
2. Solve optimization (14) to obtain optimal content placement and load distribution strategies $c_{ji}^{(m)}(t)$, $s_j^{(m)}(t)$, $y_i^{(m)}(t)$, $\forall j, i \in \mathcal{N}, m \in \mathcal{M}$;
3. Update content placement table with $y_i^{(m)}(t)$'s, and migrate files as follows:
for $i \in \mathcal{N}, m \in \mathcal{M}$ **do**
  if $y_j^{(m)}(t-1) = 0$ *and* $y_j^{(m)}(t) = 1$ **then**
    instruct on-premise server to upload file m to data center i;
  if $y_j^{(m)}(t-1) = 1$ *and* $y_j^{(m)}(t) = 0$ **then**
    signal data center i to remove file m;
4. Dispatch $s_j^{(m)}(t)$ requests from queue $Q_j^{(m)}$ to on-premise server, $c_{ji}^{(m)}(t)$ requests to data center i, $\forall j, i \in \mathcal{N}, m \in \mathcal{M}$;
5. Update virtual queue $Z_{ji}^{(m)}$ and G according to Eqn. (11) and (10);

INTERNATIONAL JOURNAL OF TECHO-ENGINEERING
ISSN : 0975-4520

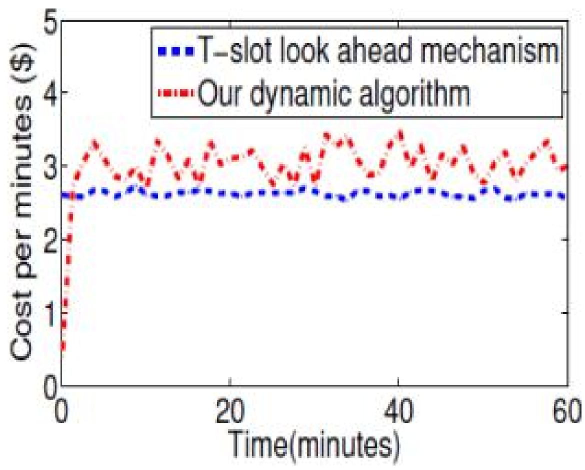

Fig.2 Cost comparison among our dynamic algorithm and the T-slot look ahead mechanism (|M| = 100) (average costs are 2.6 and 3.0 dollars respectively.

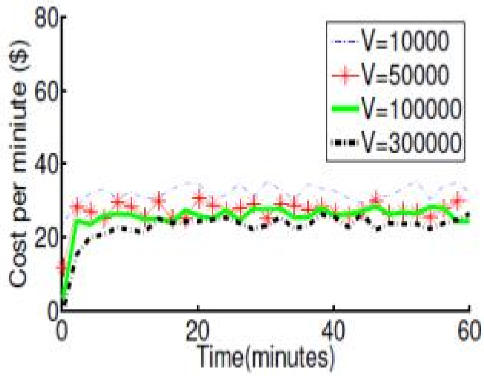

Fig.3 Cost with different V (from V = 10000 to 300000, average costs are 31.6,27.4,25.3, 23.2 dollars respectively).

Our dynamic set of rules outperforms IPMW with MWE, which is not only the most practical in terms of cost reduction but also because our dynamic algorithm can ensure that queuing delays for 100 percent of requests fall within the special QoS while subject to the same IPMW configuration With MWE you can guarantee 90% better. This is because (1) our algorithm is aware of queue closing dates in the worst case, even when the IPMW is not with the MWE; (2) Our dynamic set of bases flexibly occupy appropriately sized caches in flight due to their application in an elastic cloud, even when IPMW with MWE operates fixed size caches, incurring additional costs; (3) Our dynamic algorithm aims to achieve an excellent alternative between delays and tariffs, even when IPMW with MWE better shortens queue lengths in a satisfactory effort fashion.

IV. CONCLUSION

This paper examines the transition of a high-quality content delivery service to a hybrid cloud consisting of local personal servers and geographically distributed public cloud offerings. We propose a standard optimization framework based on the Lyapunov optimization principle. We have designed a dynamic and aggregated placement of content materials and an algorithm distribution order that reduces the operational value of the program with guarantees of the quality of service. Theoretically, we show that our algorithm computes the optimization performed by a mechanism with known facts within the target intervals T by a small regular

ISSN : 0975-4520

interval, regardless of the order arrival sample. In our ongoing work, we intend to increase the framework for content distribution services specific to special needs, along with video-on-demand services or social media software.

VI. REFERENCES

1. M. Hajjat, X. Sun, Y. E. Sung, D. Maltz, and S. Rao, "Cloudward Bound: Planning for Beneficial Migration of Enterprise Applications to the Cloud," in Proc. of IEEE SIGCOMM, August 2010.
2. H. Zhang, G. Jiang, K. Yoshihira, H. Chen, and A. Saxena, "Intelligent Workload Factoring for a Hybrid Cloud Computing Model," in Proc. Of the International Workshop on Cloud Services (IWCS 2009), June 2009.
3. W. Hao, I. Yen, and B. Thuraisingham, Dynamic service and data migration in the clouds, 33rd Annual IEEE Int. Computer Software and Applications Conf., vol.2, pp.134–139, 2009.
4. C. Reich, K. Bubendorfer, M. Banholzer, and R. Buyya, A SLA-oriented management of containers for hosting stateful web services, IEEE Int. Conf. on e-Science and Grid Computing, pp. 85–92, 2007.
5. B. Satzger, W. Hummer, C. Inzinger, P. Leitner, and S. Dustdar, Winds of change: From vendor lock-in to the meta cloud, Internet Computing, IEEE, vol.17(1), pp.69–73, 2013.
6. D. Ardagna, et al., MODAClouds: A model-driven approach for the design and execution of applications on multiple Clouds, 4th Int. Workshop on Modeling in Software Engineering (MISE), IEEE, pp.50- 56, 2012.
7. A.J. Ferrer, F. Hernandez., J. Tordsson, et al., OPTIMIS: A holistic approach to cloud service provisioning, Future Generation of Computer Systems, vol. 28, pp. 66-77, 2012.
8. M. Hajjat, X. Sun, Y. E. Sung, D. Maltz, and S. Rao, "Cloud ward Bound: Planning for Beneficial Migration of Enterprise Applications to the Cloud," in Proc. of IEEE SIGCOMM, August 2010.
9. H. Zhang, G. Jiang, K. Yoshihira, H. Chen, and A. Saxena, "Intelligent Workload Factoring for a Hybrid Cloud Computing Model," in Proc. of the International Workshop on Cloud Services (IWCS 2009), June 2009.
10. X. Cheng and J. Liu, "Load-Balanced Migration of Social Media to Content Clouds," in Proc. of NOSSDAV, June 2011