

An Cost Effective Euclidean Steiner Tree based Mechanism for Reducing Latency in Cloud

Rahul Kumar Sharma¹ Amrendra Singh Yadav² Mitra Bhushan³ Mayank Deep Khare⁴

^{1,2,3,4}M. Tech. Student

^{1,2,3,4}Department of Computer Engineering

^{1,2,3,4}Madan Mohan Malaviya University of Technology, Gorakhpur, India

Abstract— Latency is the most effective performance factor in cloud. Online gaming and online shopping are totally depend on the performance of data service, If there is a large waiting time or queue delay in processing of request then the user will abort that and change the site. Here we are trying to reduce the latency in cloud. In this paper we are using to approaches one work between the user and data center that is iCloudAccess and another which work between the VM in the data center that is FARCREST. iCloudAccess is used to perform intelligently server provisioning and request dispatching, where on the other hand FARCREST is light weight real time service latency prediction system which is based on Euclidean Steiner Tree (EST) model that is used in delay-sensitive cloud services for optimum VM resource allocation. We perform iCloudAccess algorithm in that environment which is made the help of FARCREST mechanism to get minimum latency in our system.

Key words: Cloud Computing, Latency, Virtualization, Jitter, Round Trip Time

I. INTRODUCTION

Cloud computing can be implemented at various level of abstraction depending on the specific service that is being offered by the cloud provider (i.e. storage, computation, application framework, etc.). video game was launched in the market around 45 years ago, we have witnessed a series of significant revolutions in the video game industry. The potential of cloud gaming has already attracted a great amount of attention from many industrial practitioners, ventures, and researchers. It is predicted that the size of global video game market revenue will grow up to U.S.\$78 billion in 2017, among which cloud gaming market is expected to expand the most. It is very challenging to build a cloud gaming platform that can provide users with high quality of experience (QoE)[1]. Online game is totally depends on the speed of transferring data and how much time would be spend to gather data or information from the database or cloud. Latency and the prediction system play an importance role to improve quality of experience (QoE).

Cloud computing offers a mean to decouple the application activities from the physical resources required. This has enabled consolidation of multiple applications onto a lesser number of physical servers resulting in an increase in server utilization. Today latency play a very essential role to internet over the world.

In this paper, we focus on understanding and mitigating the latency problem of cloud gaming services from the perspective of cloud gaming service providers (CGSPs). We are using an algorithm “iCloudAccess: Cost-Effective Streaming of Video Games from the Cloud With Low Latency”[2] for transferring the request and response between the datacenter and user. It also works for maintaining the request between the physical server or data

center, with this algorithm we are using a tree based placement approach (FARCREST: Euclidean Steiner Tree-based Cloud Service Latency Prediction System [3]) for Virtual machine in the data center, due to which latency will be reduce and performance will be increase. It will be very beneficial to have an expansive algorithm which gives the minimum latency result due to which performance automatically increases.

II. CLOUD LATENCY & MEASUREMENT

Latency can occur due to delay in the network because of congestion. Latency can be measured by applying some formula or algorithm. Latency could occur when two virtual Machines are co-located at the same server communicating with each other. It also cause application to spend amount of time waiting for response from distant data center, then the bandwidth may not be fully utilized and then performance will be suffer and sometime medium itself introduce some latency could occur when two VMs co-located at the same server communicate. The different between time, a user request delay, which vary from one medium to another medium. These are some point on which latency will be measure in any network.

A. Queuing Delay

The difference between the time when a user request enters in the waiting queue of a data center and the time when data center starts to serve the request. If queuing delay is too long, players will have to choose another data center or abort playing the game.

B. Response delay:

The difference between the time when client sends a player’s command to the server and the time when generated video frame is decoded and presented on the screen. Processing delay is a combination of network delay which work at network side and processing delay at the server side. Response delay has impacts on the interactivity of cloud.

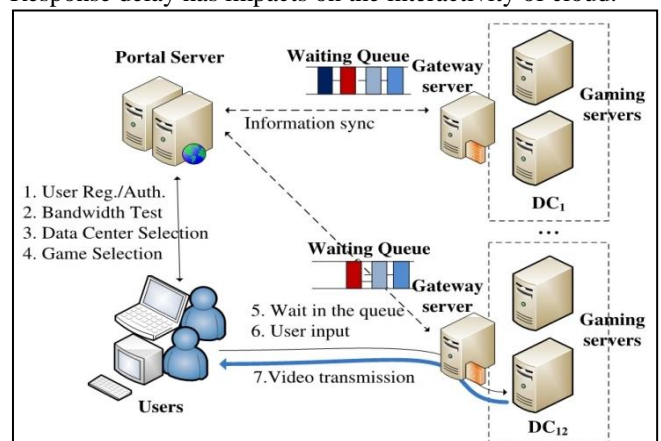


Fig. 1: Response Delay

For measurement[4][5] of latency, let us consider the distance between two LAN to be 400 miles and assume each router adds 2ms. Current network utilization without storage application is 15%. So amount of bandwidth available for new storage network application is to be 85%.The distance between two end points of network link is 400 miles. Therefore round Trip Time propagation delay is $400*2=800$ miles or equal to 8ms. If there are two routers in the path taken by data. So estimated RTT node delay is $2 \text{ nodes} * 2\text{ms}$ equal to 4ms. Now, the congestion processing delay is increased to $4\text{ms}/0.85$ equal to 5ms. Hence total network latency [Propagation Delay+ Network Delay + processing Delay] is 17ms {i.e. 8ms+4ms+5ms}.

III. RELATED WORK

When we are playing any online game or watching any video and at that time the delay in the fetching of data will make slow down the speed of processing any program due to which the program would suffer to run properly. Latency have great role to kill any program and we are presenting a efficient low latency algorithm approach which minimize the transferring and searching time and give better result to improve the quality of experience(QoE).

A. System Architecture

Basically, here we have an algorithm iCloudAccess which provides a cost-effective approach to stream video games with low latency by smart request dispatching among data centers and dynamic cloud resource provisioning and a tree based placement setting of the virtual machine at physical server due to which the distance will be minimize between the virtual machine, After the combination of approaches we will get more cost effective and time saving algorithm. Here we define both algorithm architecture and working process.

B. iCloudAccess

We are describing (fig 2) the role of iCloudAccess in the cloud gaming platform. iCloudAccess contains two major components which are follows.

- Request dispatching unit (RDU) which is responsible for dispatching requests intelligently.
- Server provisioning unit (SPU) which is responsible for adjusting number of game servers provisioned at each data center according to user demand.

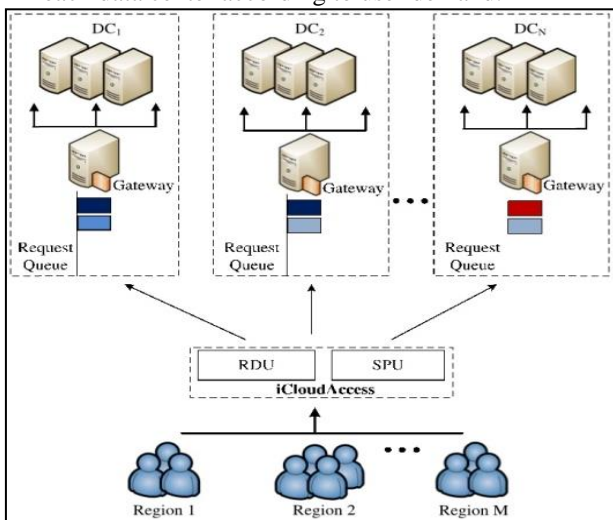


Fig. 2: iCloud Access

The operations of RDU and SPU are performed at different timescales. For every incoming user request, RDU needs to dispatch the request timely and the dispatching operation needs to be completed within a few seconds. An SPU adjusts the provisioning of cloud servers for each data center periodically, in the order of hours.

With the help of intelligently dispatching of a request (RDU) to data center with shortest waiting time, a user request queuing delay would be significantly reduced. When the demand would be over the capacity of server, the SPU will start to provision more cloud servers in the corresponding data center. After that, request processing will be transferred to the nearest and free data center to perform requesting task. It is a overview of the iCloudAccess algorithm.

C. Euclidean Steiner Tree (EST)

In the Euclidean Steiner Tree, there is explore the model of embedding latencies between VMs onto Euclidean metric space and estimating service latency taken on unmeasured VMs by constructing metric tree. Based on our empirical studies, cloud service latency can be approximated with metric tree and updated frequently based on cloud resource demand distribution.

iCloudAccess algorithm is work between only user and data center, there is not any role of virtual machine placement design in the data center so now we will re-evaluate it in the Euclidean Steiner Tree based virtual machine environment to improve the performance.

IV. WORKING OF MECHANISM

Here we are implementing the iCloudAccess algorithm in the Euclidean Steiner Tree based environment due to which we will get better performance. In this working section firstly we will define the iCloudAccess working procedure and after that we will describe the environment which based on the Euclidean Steiner Tree due to which the performance would definitely increase.

| |
|---|
| <p>Algorithm 1 iCloudAccess: Online Control Algorithm for RDU and SPU</p> <p>Input: The values of $N, M, m, \mu, \epsilon, V$; Prices of on-demand cloud servers; Number of incoming user requests $\lambda_{ij}(t)$; Network delay between regions and data centers, $d_{ij}(t)$;</p> <p>Output: RDU and SPU decision $\vec{\lambda}(t), \vec{n}(t)$.</p> <ol style="list-style-type: none"> 1: Initialization step: Let $t = 0$, and set $Q_j(0) = 0, H_j(0) = 0$, for $j = 1, 2, \dots, N$. 2: while the cloud gaming service is running do 3: if $(t \bmod m) == 0$ then 4: Monitor the queue backlog $Q(t), H(t)$ and the real-time information of $c_j(t)$ for each data center j. 5: Determine the SPU decision $\vec{n}(t)$ by solving $\min_{\vec{n}(t)} \Theta_2$; 6: end if 7: Update information of network delay between a region i and a data center j, and the amount of user requests from a region i (i.e., $\lambda_i(t)$) for $i = 1, 2, \dots, M, j = 1, 2, \dots, N$. 8: Determine the RDU decision $\vec{\lambda}(t)$ by solving $\min_{\vec{\lambda}(t)} \Theta_2$; 9: Update Q and H according to (9) and (11), respectively. 10: end while |
|---|

Fig. 3: Algorithm

In this algorithm N geographically distributed data centers are providing gaming services to users by spreading over M regions. When a region ith will send a request for

service then RDU will to send its request to short queue j^{th} data center and after that SPU will work if that data center has already a large queue of requests to proceed then SPU will search the nearest and free server or database to transfer that request to process in smallest time.

– Farcrest: Euclidean Steiner Tree-based Cloud Service Latency Prediction System

In this approach the service latencies measured between VMs satisfy the properties of a tree metric and the shortest paths between VMs can be represented by the EST [6] model. With this tree model, latencies between VMs can be derived from the summation of tree edge(s) values that are partially collected from actual measurement, and partially predicted. We call the derivative tree as a Service Latency Prediction Tree (SLPT).

The EST formulates a model to construct a shortest length spanning tree in a metric space. Let (V, d) be a metric space and let $x, y, z \in V$. Consider a set D measure of service latency between x, y and z , represented by $d(x,y), d(x,z)$ and $d(y,z)$ respectively. This set of measurements is a tree metric if the following properties hold:

- 1) $d(x,y) \geq 0$ – non-negativity
- 2) $d(x,z) \leq d(x,y) + d(y,z)$ – triangle equality
- 3) $d(x,y) = d(y,x)$ – symmetry

The Steiner Tree is formed by intersecting all edges $e_{xy} e_{xz} e_{yz} \in E$, through an extra intermediate point called a Steiner point. Note that in EST, a Steiner point is connected up to 3 degrees. The measurements D can be embedded in Euclidean space d by calculating the gromov product of each host respectively. For example, the gromov product of y and z at x , denoted by $(y,z)_x$, is defined by

$$(x,y)_z = 1/2(d(x,y) + d(x,z) - d(y,z)).$$

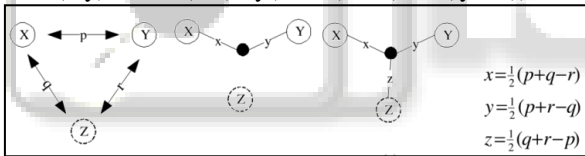


Fig. 4: Service Latency Prediction Tree (SLPT)

The EST provides a few key intrinsic values to form a SLPT. First, it reduces the number of measurement required for service latency towards all VMs by providing a model to add a new VM to the metric tree by only requiring the new node to measure to a subset of the existing nodes. This is achieved by predicting the service latency using the measurement of a selected set of actual service latencies.

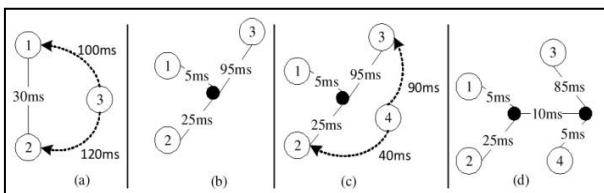


Fig. 5: SLPT

In Figure (a), the service latency between VM1 and VM2 can be obtained from actual measurements or deduced from the existing EST in case prior measurements have already been done. The service latency between VM3 towards VM1 and VM2 should be measured. An EST is formed as shown in Figure (b). When VM4 is selected as the candidate VM for a distributed task, the measurement can be done from VM4 to any two of the existing VMs, in this case VM2 and VM3 are selected, as shown in Figure (c). The

service latency from VM1 to VM4 can be derived by sum of values from the set of edges which constitute the path from the VM1 to VM4. In this example, service latency between VM1 to VM4 is (5+10+5)ms.

V. PREDICTABLE PERFORMANCE

In this section we are try to shoe predictable performance of the given system on the behalf of both approach

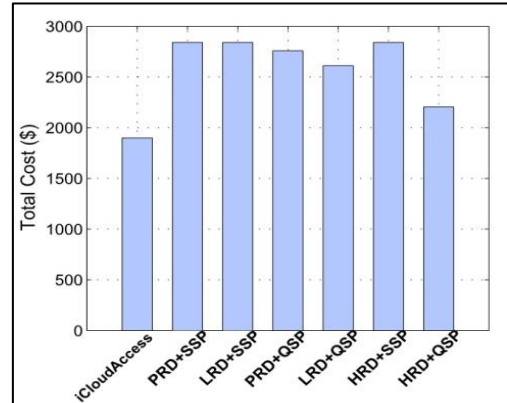


Fig. 6: Performance of iCloud Access

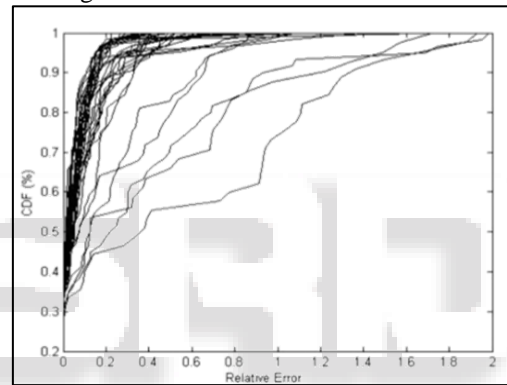


Fig. 7: Performance result of Farcrest

Figure 6 shows the total server provisioning cost incurred by different methods during the simulation period. Our simulation lasts for 3000 min. PRD + SSP and LRD + SSP incur the highest provisioning cost, and our proposed iCloudAccess has the lowest provisioning cost. In Figure 7 the prediction tree shall be more accurately generated with measurement towards 100% of existing VMs, but then this requires mesh measurement, which violates the purpose of using prediction-based method for selective measurement

VI. CONCLUSION AND FUTURE WORK

By analysing of these two results, we can say that this approach will definitely work because both perform different conditions one is work between the users and data center and another is work between virtual machine in the data center. In this mechanism, our work in process condition. We also trying to reduce the time complexity of iCloudAccess algorithm in future and make it more flexible and easy for online shopping and gaming also.

REFERENCES

[1] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, "An evaluation of QoE in cloud gaming based on subjective tests," in Proc. IEEE 5thInt. Conf. IMIS, Jul. 2011, pp. 330–335.

- [2] Di Wu, Zheng Xue, and Jian He “ iCloudAccess: Cost-Effective Streaming of Video Games from the Cloud With Low Latency” in IEEE Transactions on Circuits And Systems for Video Technology, Vol. 24, No. 8, August 2014.
- [3] Boon Ping Lim, Poh Kit Chong, Ettikan Kandasamy Karupiah, Yaszrina Mohamad Yassin, Amril Nazir, Mohamed Farid Noor Batcha “FARCREST: Euclidean Steiner Tree-based Cloud Service Latency Prediction System”(2013) in 10th annual IEEE CCNC.
- [4] Z. Xue, D. Wu, and J. He, “A measurement study of a large scale commercial cloud gaming system,” Dept. Comput. Sci., Sun Yat-sen Univ., Guangzhou, China, Tech. Rep. CS-2013-05, 2013.
- [5] S. Choy, B. Wong, G. Simon, and C. Rosenberg, “The brewing storm in cloud gaming: A measurement study on cloud to end-user latency,” in Proc. IEEE 11th Annu. Workshop Netw. Syst. Support Games, Nov. 2012,pp. 1–6.
- [6] D. Du, X. Hu, "Steiner Tree Problems In Computer Communication Networks", World Scientific Publishing Company, 2008.
- [7] L. Ang, X.W. Yang, K. Srikanth, and M. Zhang, “CloudCmp: shopping for a cloud made easy”, in USENIX HotCloud, 2010.

