Queueing Based Edge Server Selection in Content Delivery Network Using Haversine Distance

Debabrata Sarddar
Department of Computer Science & Engineering
University of Kalyani
Kalyani, India

Sandip Roy
Department of Information Technology
Brainware Group of Institutions
Kolkata, India

Rajesh Bose
Senior Project Engineer
Simplex Infrastructures Ltd.
Kolkata, India

Abstract—**New generation of technology is inclining towards the uphill cloud computing technology. Many users and organizations could not able to embrace cloud services blindly because of their uncovering security issue for managing huge amount of data. Data is transmigrated between different servers which are disseminated globally. In cloud computing large number of cloud user should be cognizant of the physical location of their data over the earth surface to guarantee whether their data localization should also be maintained different data privacy law. At any peak hour when large number of cloud users' request send to the server accordingly network congestion and the number of data loss are drastically increased, the network becomes unmanageable. Content Delivery Network is solution to remove these network hazards. We build the queueing based edge server selection algorithm to handle huge number of requests made by giant web customers to overcome the intrinsic limitations of cloud network.**

Keywords- Cloud Computing; Content Delivery Network; Edge Server; Haversine Distance; Network Latency; Queuing Theory

I. INTRODUCTION

Content Delivery Network (CDN) replicates contents over several edge servers placed at various geographical locations for handling the request of burst crowds over the internet [1]. CDN improves the performance of the cloud network in which popular web services suffer network congestion due to the large demands made on their services by serving user's request through the nearest server and this approach helps to reduce the response time of the user request.

The replica of the end-users' content are distributed among a set of edge servers scattered over geographical regions using content replica placement algorithm. The request routing infrastructure is responsible for redirecting client request to appropriate edge servers when requests are generated from end-users for particular services for the original host server and original host server is far away from the client at that instant [2]. CDN improves the network performance by maximizing bandwidth and maintaining correctness by content replication. In this paper, we focus on searching an edge server such that the latency time between client and selected edge server must be minimized. Therefore the average access time of the network should be reinforced.

II. PRIVIOUS WORK

Al-Mukaddim Khan Pathan and Rajkumar Buyya presented detail taxonomy of CDNs with respect to four different factors as request-routing procedure, content replication mechanisms, load balancing methodology and cache management [2]. In this paper, they have further build up taxonomies for each of these paradigms to relegate the common trends in content networking. Here comparison of existing CDNs are articulated in such a manner that helps one to get an insight into the technology, services, strategies, and practices which are currently followed in this field. S. Saroiu, K. P. Gummadi, R. J. Dunn, St. D. Gribble, and H. M. Levy examined four content delivery systems such as web traffic, Content Delivery Network of the Akamai and Kazaa and Gnutella peer-to-peer file sharing traffic [4]. G. Pallis, A. Vakali, K. Stamos, A. Sidiropoulos, D. Katsaros and Y. Manolopoulos addressed the content replication problem. Their objective is to minimize the average response time when clients fetch objects from the nearest edge server [5]. Khai Hsiang Wong, Phooi Yee Lau, and Sungkwon Park focused on simulating the concepts of Soarin and their simulation technique concentrate in the relationship between the placement strategy and throughput among different edge servers [6]. In queueing theory based cloud computing, M/M/S queueing model is used for multiple servers overcome M/M/1 queueing model for reducing the mean queue length and waiting time [7].

III. OUR LATENCY BASED PROPOSED SIMULATION TECHNIQUE

Our latency based edge server simulation technique aims to calculate latency from sender to receiver by focusing on the area of calculating propagation delay, serialization delay and queueing delay. Propagation delay depends upon the distance between source and destination. We can calculate distance between two locations using haversine formula [8]. Serialization delay concentrates in the bandwidth available between sender and receiver [3]. Here we consider the M/M/S queueing model to calculate queueing delay [10]. When the server becomes busy, queue length and average waiting time will be increased. Finally we compare our latency based proposed simulation technique with the latency using M/M/1 queueing model to calculate queueing delay to improve quality of service (QoS) in cloud environment [9].

A. Calculate distance between two locations on the earth using haversine formula

The haversine formula is used for calculating distance between sender and receiver on the earth evaluated from their given longitudes and latitudes.

The following algorithm is used for calculating distance between sender and receiver.

1. R ← 6371 // Radius of Earth in k.m.
2. Φ_1 ← lat1.toRadians() // Latitude of sender
3. Φ_2 ← lat2.toRadians() // Latitude of receiver
4. λ_1 ← lon1.toRadians() // Longitude of sender
5. λ_2 ← lon2.toRadians() // Longitude of receiver
6. ΔΦ ← (lat2-lat1).toRadians()
7. Δλ ← (lon2-lon1).toRadians()
8. a ← Math.sin(ΔΦ/2) * Math.sin(ΔΦ/2) + Math.cos(Φ_1) * Math.cos(Φ_2) * Math.sin(Δλ/2) * Math.sin(Δλ/2)
9. c ← 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a))
10. d ← R * c // Haversine distance between sender and receiver over the earth surface

B. Calculate propagation delay between sender and receiver using haversine distance

Propagation delay is the amount of time required for the head of the signal to travel from sender to receiver over the cloud environment. Satellite communication uses electromagnetic waves to circulate information through the space. The transmitter converts the information from electrical signals to radio signals. The radio signals travel approximately at the speed of light for free space. Propagation delay can be calculated by the following formula:

Propagation delay = d / Speed of Light

C. Calculate serialization delay between sender and receiver

Serialization delay is the amount of time required to push all the packet's bits over the channel between sender and receiver in the cloud system. Larger bandwidth plays an important role for reducing the serialization delay. Serialization delay is calculated as follows:

Serialization delay = Packet size in bit / Transmission rate in bits per second

D. Calculate queueing delay between sender and receiver

Queuing delay is the amount of time a job waits in a queue until it can be served by a particular server. We can calculate queuing delay or waiting time for single and multiple servers using queuing theory [11]. Let's consider arrival rate λ and service rate μ and mean waiting time for M/M/1 and M/M/S queueing model are calculated by following formulas.

Mean waiting time for M/M/1 is given by

$$W_q = \lambda / \mu (\mu - \lambda)$$

Mean waiting time for M/M/S is given by

$$W_q = L_q / \lambda$$

Where $L_q = \frac{P_0\, \rho^s \rho_s}{s!(1-\rho_s)^2}$ and the probability that no customers in the queue is given by

$$P_o = \left[\sum_{n=0}^{s-1}\left(\frac{\rho^n}{n!}\right) + \frac{\rho^s}{s!\,(1-\rho_s)}\right]^{-1}$$

E. Calculate latency from sender to receiver

Latency is an expression of how much time it takes for a packet of data to get from one point to another. Latency from sender to receiver can be calculated by the following formula.

Latency = Propagation delay + Serialization delay + Queueing delay

IV. OUR PROPOSED ALOGORITHM FOR SELECTING EDGE SERVERS OF A PARTICULAR CDN

The earth is divided into several regions by the geographers. The land of the surface of the world can be logically partitioned into eight regions: Asia, Europe, Antarctica, Africa, Australia, North and South America and the water surface of the world can be subdivided into four major oceans from largest to smallest, are the Pacific Ocean, Atlantic Ocean, Indian Ocean, and Arctic Ocean [12]. In our algorithm we are placing the edge servers of a particular Content Delivery Network (CDN) over the eight regions and all the edge servers are periodically synchronized by the CDN Provider which is monitoring a particular CDN. Selecting a specific edge server based on least latency of a particular CDN Provider is the primary goal of our algorithm [13, 15].

To access a particular content from the web server cloud user sends corresponding request to the particular web server. Subsequently web server provides the basic index page to the corresponding cloud user [14]. Then the original web server redirect the request to the specified CDN Provider for delivering the content. CDN Provider can update the information periodically.

Our simulation technique helps CDN Provider to calculate latency using propagation time, serialization time and queuing time of each registered edge server of the CDN Provider. Then CDN Provider creates a list of registered edge server based on the minimum latency. It then finds out the suitable edge server from the list to establish the connection with cloud user. In this paper we have studied the fact of selecting edge server with minimum latency value and at busy status would lead to high average waiting time. We select the edge server M/M/S queuing model instead of M/M/1 queuing model. We explained our proposed edge server selection algorithm of a particular CDN in the next section.

Figure 1. The earth can be logically partition into eight regions and four oceans

A. Algorithm for selecting edge server of a particular CDN

1 A set **U = {**e_1, e_2, e_3,…, e_n**}** of multiple cloud users from different location over the earth surface request to connect with the original host server using Uniform resource locator (URL) or specific IP address.

2 Par For each i = 1 to n do
2.1 Original host server returns the basic index page to the corresponding requested cloud user $\mathbf{e_i}$.
2.2 Redirects the request to the CDN Provider.
2.3 CDN Provider checks the geographic coordinate values (i.e. latitude or longitude) of $\mathbf{e_i}$ and based upon its coordinate value CDN Provider selects the set **S** of edge server specific or neighbor to the geographic region of the requested cloud user $\mathbf{e_i}$.
2.4 Calculate haversine_distance between requested cloud user $\mathbf{e_i}$ and each edge servers from set S.
2.5 Collect the available bandwidth of each edge server belongs to the set S.
2.6 Based upon haversine distance and bandwidth calculate propagation delay and serialization delay respectively for each edge server of set S.
2.7 Calculate the queuing delay of each edge server belonging in set S of the CDN Provider based on M/M/S queueing model.
2.8 Calculate the latency of each edge server belonging to the set S.
2.9 Selects edge server S_L based upon least latency from the set of edge servers S.
2.10 If selected least latency edge server $\mathbf{S_L}$ has the ability to serve the content requested by the cloud user $\mathbf{e_i}$.
2.10.1 Selection of the least latency edge server $\mathbf{S_F}$ gets finalized with $\mathbf{S_L}$.
2.11Else
2.11.1 Eliminate $\mathbf{S_L}$ from S.
2.12Repeating the step 2.9 for choosing the next least latency edge server.
3 End for
4 Connection is established between requested cloud user $\mathbf{e_i}$ and finally selected suitable latency edge server $\mathbf{S_F}$ of user specific geographic region until requested cloud user $\mathbf{e_i}$ wants to disconnect the facility from the original host server.
5 End

B. Practical example with result analysis

Step 1. At any instant suppose there are multiple users from different region are requesting for a particular host server. For simplicity let us consider a single cloud user from Kolkata, who wants to request for a mail service for sending 1500 bytes data using the mail server located at Chicago and the request send to the original server at Chicago.
Step 2. The haversine distance d from Kolkata to Chicago is 12840 km and original server redirects the request to the CDN Provider.
Step 3. The CDN Provider stores the details information of the edge servers. All edge servers, which are geographically clustered based upon seven geographical regions as shown in Table 1. The set S of edge servers specific or neighbor to the geographic region of the requested cloud user is generated enlisting those placed at Singapore, Colombo, New Delhi, Ankara, Islamabad and London respectively.
Step 4. In Table 2 we have stored the mean waiting time of every edge server using M/M/1 and M/M/S queueing model.
Step 5. The haversine distance and bandwidth of each edge server from cloud user located at Kolkata is calculated and enlisted in Table 3.
Step 6. Propagation delay, serialization delay and queuing delay are calculated and periodically updated by the CDN Provider as like as Table 4 and Table 5.
Step 7. CDN Provider selects edge server S_L placed at Colombo from six edge servers belonging to the set S and is

capable of serving the request of user placed at Kolkata. So edge server placed at Colombo is our S_F.

Step 8. S_F is connected with the cloud user of Kolkata until the user disconnects from the service.

TABLE I. NAME OF THE PLACES OF THE EDGE SERVER OF PARTICULAR CDN

TABLE II. MEAN WAITING TIME OF EVERY EDGE SERVER USING M/M/1

	Name of the region	Name of the place of edge server
1	**Asia**	**Singapore**
		Colombo
		New Delhi
		Ankara
		Islamabad
2	**Europe**	**London**
3	North America	Mexico
		Ottawa
		Chicago (Host server)
		Kingston
4	South America	Santiago
		Buenos Aires
5	Africa	Harare
		Cape Town
6	Australia	Canberra
7	Antarctica	Not available

AND M/M/S MODEL

Host server	**Arrival rate λ Customers/sec**	**Service rate μ Customers/sec**	**Mean waiting time W_q ms M/M/1**	**Mean waiting time W_q ms M/M/S**
Chicago	20	40	25	0.000065
Edge server	**Arrival rate λ Customers/sec**	**Service rate μ Customers/sec**	**Mean waiting time W_q ms M/M/1**	**Mean waiting time W_q ms M/M/S**
London	20	40	25	0.000065
Singapore	10	11	909	0.006648
Colombo	120	125	192	0.000787
New Delhi	120	121	991	0.000969
Ankara	10	15	133	0.00088
Islamabad	120	125	192	0.000787

TABLE III. HAVERSINE DISTANCE AND BANDWIDTH OF HOST SERVER AND EACH EDGE SERVER

Host server	**Latitude**	**Longitude**	**Haversine distance (km)**	**Bandwidth**
Chicago	41.8819°N	87.6278°W	12840	10Gbps
Edge server	**Latitude**	**Longitude**	**Haversine distance (km)**	**Bandwidth**
London	51.5072°N	0.1275 °W	7982	250 Mbps
Singapore	1.3000 °N	103.8000°E	2894	1 Mbps
Colombo	6.9344 °N	79.8428°E	1963	500 Mbps
New Delhi	28.6100°N	77.2300°E	1302	1Mbps
Ankara	39.9300°N	32.8600°E	5522	1Mbps
Islamabad	33.7167°N	73.0667°E	1942	1 Mbps

TABLE IV. LATENCY OF HOST SERVER AND EACH EDGE SERVER USING M/M/1 QUEUEING MODEL

	Original host server	**Propagation delay(ms) A**	**Serialization delay(ms) B**	**Queuing delay (ms) C**	**Latency (ms) A + B + C**
1	*Chicago*	42.81	0.0012	25	67.8112
	Edge server	**Propagation delay(ms) A**	**Serialization delay (ms) B**	**Queuing delay (ms) C**	**Latency (ms) A + B + C**
1	London	26.63	0.048	25	51.678
2	Singapore	9.65	12	909	930.65
3	Colombo	6.55	0.024	192	198.574
4	New Delhi	4.34	12	991	1007.34
5	Ankara	18.41	12	133	163.41
6	Islamabad	6.8	12	192	210.48

TABLE V. LATENCY OF HOST SERVER AND EACH EDGE SERVER USING M/M/S QUEUEING MODEL

	Original host server	**Propagation delay (ms) A**	**Serialization delay (ms) B**	**Queuing delay (ms) C**	**Latency (ms) A + B + C**
1	*Chicago*	42.81	0.0012	0.000065	42.811265
	Edge server	**Propagation delay (ms) A**	**Serialization delay (ms) B**	**Queuing delay (ms) C**	**Latency (ms) A + B + C**
1	London	26.63	0.048	0.000065	26.678065
2	Singapore	9.65	12	0.006648	21.656648
3	Colombo	6.55	0.024	0.000787	6.574787
4	New Delhi	4.34	12	0.000969	16.340969
5	Ankara	18.41	12	0.00088	30.41088
6	Islamabad	6.48	12	0.000787	18.480787

In this paper, our proposed algorithm calculate the haversine distance between cloud user located at Kolkata to the original host server at Chicago and all the edge servers located different places particular or neighbor to the user as shown in Table 3 [16]. It is found that the haversine distance of New Delhi from Kolkata is the smallest distance among the all edge servers and the calculated propagation delays are represented in the Figure 2 (a). Figure 2 (b) explains the serialization delay which is also directly proportional with available bandwidth between cloud user and edge server that is calculated by our simulation process. Based upon periodically updated value of arrival rate and departure rate from the available server in a particular CDN, our proposed algorithm is calculated the queueing delay for M/M/S queueing model which is shown in Figure 3 (b) and also compare with the queueing delay for M/M/1 queueing model in Figure 3 (a).

At last latency is calculated by our simulation algorithm and CDN Provider selects the least latency available edge server. From our data CDN Provider select the edge server located at London using M/M/1 queueing mode land also choose the edge server located at Colombo using M/M/S queueing model as shown in Figure 4 and Figure 5 respectively and at last connection is established between the cloud user and the aforesaid edge server. In this paper we have studied that latency of selecting edge server at Colombo using M/M/S queueing model is 47.1 ms faster than the latency of choosing edge server located at London using M/M/1 queueing model.

C. Comparison study using simulation graph

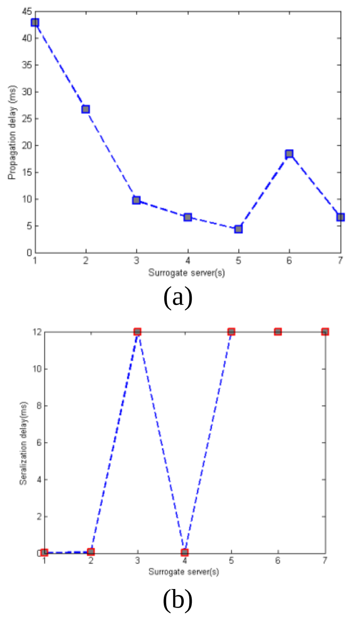

Figure 2. Propagation delay (a) and serialization delay (b) of different edge servers of a particular CDN

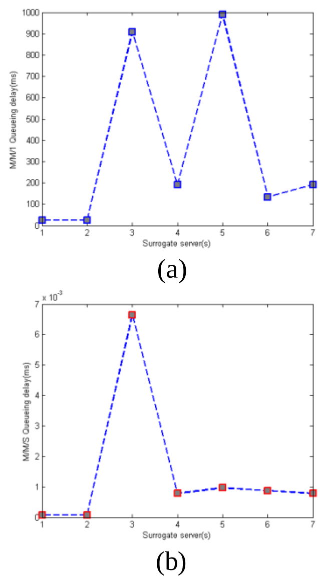

Figure 3. M/M/1 queueing delay (a) and M/M/S queueing delay of different serevrs of a particular CDN

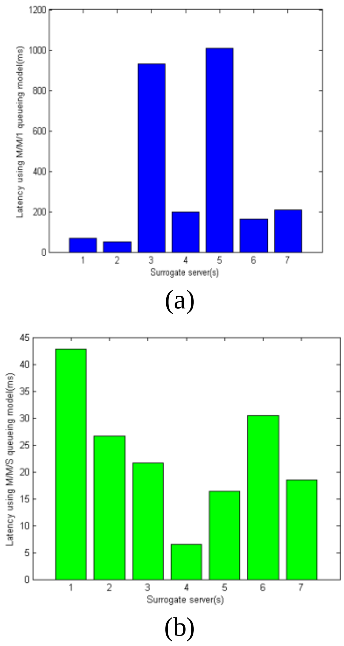

Figure 4. Latency of different servers using M/M/1 queueing delay (a) and M/M/S queueing delay (b) of a particular CDN

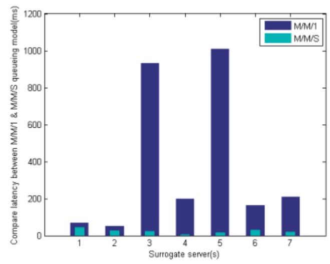

Figure 5. Compare latency between M/M/1 and M/M/S queueing model of a particular CDN

V. CONCLUSION

In this paper, region based clustering and neighbor region selection for a requested cloud user is challenging problem for selecting set of multiple edge servers of a particular Content Delivery Network. Here we have shown latency based edge server selection algorithm based on propagation delay, serialization and queueing delay. Moreover we have also considered M/M/S queueing model for multiple servers which grows the performance over using one server reducing queue length and waiting time. Our simulation analysis and numerical result shows that latency based edge server selection guarantees the quality of service requirements of the cloud user's jobs and also makes maximum benefits for the CDN Provider.

REFERENCES

[1] M. Arlitt and T. Jin, "A Workload Characterization Study of 1998 World Cup Web Site," *IEEE Network*, May/June 2000, page no(s) 30-37.

[2] Al-M. Khan Pathan and R. Buyya, "A Taxonomy and Survey of Content Delivery Networks,"www.cloudbus.org/reports/CDN-Taxonomy.pdf

[3] www.03bnetworks.com, "What is Network Latency and Why Does It Matter?"

[4] S. Saroiu, K.P. Gummadi, R J. Dunn, S D. Gribble and H. M. Levy, "An Analysis of Internet Content Delivery Systems," NSF grants CCR-0121341, ITR-0085670 and IIS 0205635.

[5] G. Pallis, A. Vakali, K. Stamos, A. Sidiropoulos, D. Katsaros, and Y. Manolopoulos, "A Latency-based Object Placement Approach in Content Distribution Networks," national programs funded by the ΕΠΕΑΕΚ.

[6] K. H. Wong, P. Y. Lau and S. Park, "Using Surrogate Servers for Content Delivery Network Infrastructure with Guaranteed QoS," JACN, (Vol. 1: No. 1), March 2013.

[7] T. Sai. Sowjanya, D. Praveen, K. Satish and A. Rahiman, "The Queueing Theory in Cloud Computing to Reduce the Waiting Time," IJCSET, (Vol.1: Issue 3), April 2011, page no(s). 110 - 112.

[8] Sinnott, R. W. "Virtues of the Haversine," Sky and Telescope, (Vol.68: No. 2), 1984, page no(s). 159.

[9] J. Abate and W. Whitt, "Transient behavior of the M/M/l queue: Starting at the origin", Queueing Systems, (Vol. 2: Issue. 1), March 1987, page no(s).41-65.

[10] D. J. Daley and L. D. Servi, "Idle and busy periods in stable M / M / k queues," Journal of Applied Probability, (Vol.35: No. 4), 1998, page no(s) 950-962.

[11] Dr. J. Sztrik, Basic Queueing Theory, University of Debrecen, Faculty of Informatics, Debrecen, 2012, page no(s). 17, 149.

[12] "Geography — Student Reading Understanding Maps of Earth", NASA ISS EarthKAM

[13] L. Cherkasova "From Internet Data Centers to DataCenters in the Cloud", HP Research Labs

[14] D. C. Verma, Content Distribution Networks*:* An Engineering Approach, John Wiley & Sons, Inc., New York, USA, 2002.

[15] M. Andrews, B. Shepherd, A. Srinivasan, P. Winkler and F. Zane, "Clustering and server selection using passive monitoring, "Proc. of IEEE INFOCOM, NY, USA, 2002.

[16] D. Sarddar, S. Roy and R. Bose, "An Efficient Edge Servers Selection in Content Delivery Network Using Voronoi Diagram", IJRITCC, (Vol. 2: No. 8), 2014 page no(s) 2326-2330.