SindhUniv. Res. Jour. (Sci. Ser.) Vol. 49(1) 243-248 (2017)

SINDH UNIVERSITY RESEARCH JOURNAL (SCIENCE SERIES)

Estimating the Best Performance Option for Content Delivery Networks

A. KUMAR++, F. KHUHAWAR, N, ZAKI, S. ALI, B. KHALID, A. SHAH

Department of Telecommunication, Mehran University of Engineering and Technology, Jamshoro,Sindh, Pakistan

Received 15thAugust 2016 and Revised 26th February 2017

Abstract: Content Delivery Network (CDNs) have been developed to overcome the fundamental limitations of Internet in terms of Quality of Service (QoS). A CDN replicates the content of origin server to surrogate servers placed across the globe to deliver the contents to the end users in an efficient way. Content delivery on the Web has received considerable research attention and the idea is to suggest an efficient networking infrastructure and replica scenario through performance analysis of CDN. Despite its inherent limitations, even a not very complex infrastructure can show many characteristics about its ability to serve website and media traffic in a low latency model. In this paper Server utilization issues, request failure as well as mean response time are analyzed by testing different CDN policies. An efficient infrastructure is then suggested to provide guaranteed QoS for Web contents using content delivery networks.

Keywords: CDN, CDN Policies, Cache, QoS

1. INTRODUCTION

Internet traffic is increasing day by day, people use resources hungry applications such as web objects (*text, graphics and scripts*), download-able content (*images and video files, software tools, and papers*), applications (*e-commerce, portals*), real time streaming media, multimedia services, and social networks. Whenever user sends a request, sometimes it takes a lot of time to process or clients request are aborted due to network congestion, missing cache location or narrow bandwidth of the links (Krishnan, *et al.,* 2000). Hence, it is difficult to manage and deliver data through single main server. In order to store and deliver large amount of web contents, an efficient network infrastructure is required to provide service to users geographically at the edges of the globe (Dilley, *et al.,* 2002).

By introducing CDN architecture, congestion and round trip time is reduced, whereas service quality is increased. In today's dynamic web aspect, it is more important that service providers understand the needs and demands of clients. For instant, consider a case of video broadcasting services such as YouTube and Netflix, while conveying video object to geologically scattered clients, the video experience can fluctuate relaying upon the delivery path to the clients (Cronin, *et al.,* 2002), (Day *et al.,* 2003). Internet advertisement companies, data centers, Internet service providers (ISP's), on line music, mobile operators and so many other companies are the typical customers of CDN and wants to deliver their content to the end users in a very reliable manner (Silva, *et al.,* 2016), (Fortino *et al.,* 2008), (Suresh, *et al.,* 2016). Studies demonstrate that the impacts of clients to content quality issues can mostly affect the subscriptions to the services offered by the video scattering service. The architecture of CDN is based on surrogates that are distributed geographically at the edges of the globe. These servers replicates the data from the origin server through CDN distribution node, allowing content providers to upload their data over these surrogates servers (Wang, *et al.,* 2002). If content is not found at any surrogate server, the request is redirected to the other surrogate server (Katsaros, *et al.,* 2009). CDN reduces the hop to hop delivery and hence reults in a more efficient network with less bandwidth utilization for delivering content. Refer to the **(Fig. 1)** for basic architecture of CDN (Hofmann and Beaumont, 2005), (Jamin, *et al.,* 2001).

The paper is organized as follows. The related work is discussed in Section 2. Section 3 presents the simulation model and design. The results and discussion are highlighted in Section 4. Finally, we conclude the work in Section 5.

2. RELATED WORK

Past few years have been observed an advancement of technologies that target to boost content delivery and repair provisioning over the Internet. The initial concept of CDNs have been developed around since the 1990s, and in their initial concept they were known as static CDNs. At that time, their main objective was enhancing website performance quality and they achieved this by serving up stored static HTML and downloadable documents (Chen, *et al.,*

++Corresponding Author Email: aneshrajpoot@gmail.com; Tel.: +92-333-7109437

2002). These initial stage CDNs accompanied hefty sticker prices, and as such they were to a great extent reserved for the corporate division. The second era of CDNs were dynamic, which were able to deliver both static and dynamic content, including rich interactive internet content (Kamiya, *et al.*, 2009). The other real progression was the capability to provide load balancing, distributing traffic across the network of servers so as to ensure accessibility (Sidiropoulos, *et al.*, 2008). This newly and enhanced version of CDNs was more moderate than the static CDNs, yet they were still for the most part utilized by the corporate area and the professional side. CDN's from Akamai technology and commercial service providers supervised and managed to deliver internet contents (i.e., websites, messages, emails, videos, URLs and documents) to any end of the world with the high accessibility and quality demanded by the end users. There was no any kind of expensive infrastructure before 9/11 incident in USA, which resulted in serious caching problem. The researchers from MIT university evolved out Akamai Technologies to solve the flash crowd problem. This evolution motivated the CDN technology providers to invest more in CDN framework development. Recently, Akamai has been delivering about 15 to 30 percent of all internet traffic, crossing 4 terabits per second (Bartolini, Casalicchio, and Tucci, 2004). Many researchers have proposed various techniques for supporting delivery of various communication content over the Internet. The techniques which were used are caching proxy, CDN process using web caching, load balancing, request routing, and server cluster in single CDN (Oberheide, *et al.*, 2007), (Mulerikkal and Khalil, 2007), (Ubuntu Content Delivery Networks), (Sarma and Setua, 2016). For narrow bandwidth users, there is deployment of caching proxy by ISP. In order to improve performance and less bandwidth consumption, caching proxies are installed near to the end users, send request through these caches rather than sending to the origin server. The user web browsing session is completed through particular caching proxy when the entire configuration is properly done. Different level of arrangement may be deployed by ISP such as Local, Regional, International referred as hierarchical caching and this has improved performance and saved bandwidth (Ubuntu request-routing-in-cdn). CDN is a distributed architecture and based on the idea of content replication, where bunch of surrogate servers are placed geographically at the edges of the globe and the end users are connected to it. Any request sent by the user will be proceed by the three techniques web caching, load balancing and request routing. Many methodologies have been proposed in the placement and usage of surrogate servers in past. A particular paper has proposed a new CDN architecture by creating server cluster form in a single CDN.

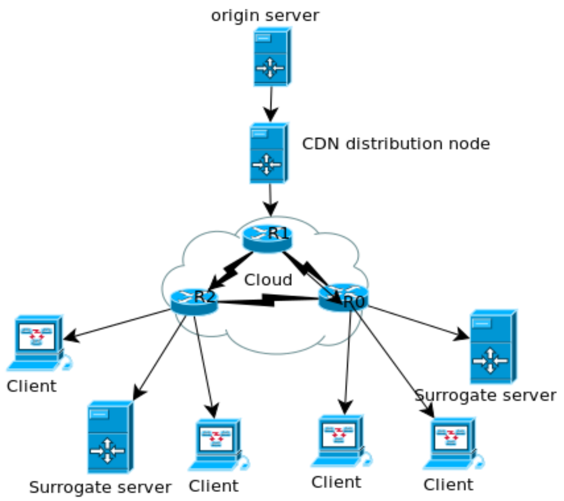

Fig. 1: Basic Architecture of CDN

3. SIMULATION MODEL

A simulation framework has been developed to provide real-time simulation for content delivery networks (CDNs), where surrogate servers, the TCP/IP protocol, and the primary CDN functionalities (Pallis, *et al.*, 2005) are simulated. The fundamental advantages of the simulation tool are its high throughput and its extensibility to configure its parameters. A number of experiments are conducted to simulate different CDN scenarios such as surrogates placed closest to the clients, clients to randomly access the surrogates, load balancing servers, and clients closest to origin server.

The aim of the work is to analyze the performance of CDN and to know what is the configuration that would results in designing an efficient CDN. We have varied a number of parameters to observe its impact such as to know, the impact of increasing number of surrogates and number of customers and so on.

To simplify, the link bandwidth is kept to 10Mbps for each network link. **(Fig. 2)** shows the sample topology having 15 routers, one origin server, 10 surrogate servers, and 20 clients. A number of experiments are performed using the similar topology to analyze the performance of 4 different policies for CDN. For each experimental scenario, separate parameter code file has been made and these files are for router's topology design. A content distribution file, which represents how the content are delievered to the clients. A traffic distribution file, which reflects the traffic. For instance, which client/clients, when and what they will request from the CDN. A cache placement file, and some other files are uploaded to create bottles. Simulation is run for adequate interval of time and the experimental execution takes around half an hour to generate result files.

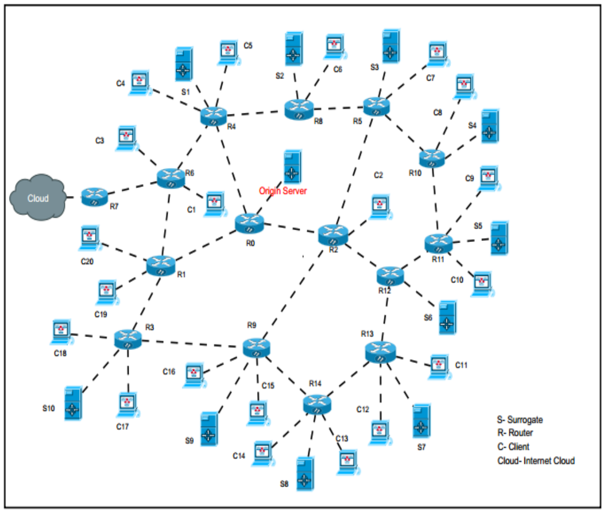

Fig. 2: CDN Network Topology

4. EXPERIMENTAL RESULTS AND DISCUSSION

The results of extensive experiments using 4 different policies (closest surrogates, random surrogates, Load balancing surrogates and Closest Origin) are described in this section in the form of graphs to show the relationship between various parameters such as mean response time vs number of clients. Experiments have been performed by setting up two conditions, first by varing number of surrogates while keeping constant number of clients. In second, surrogtes are kept constant but clients number has been varied. It is noticed that CDN policies are able to handle the increasing number of clients but require time to balance out the increased capacity to serve content.

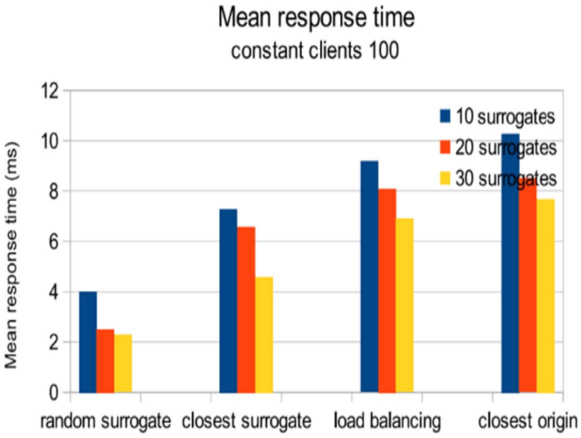

Fig. 3: Mean response time keeping 100 constant clients

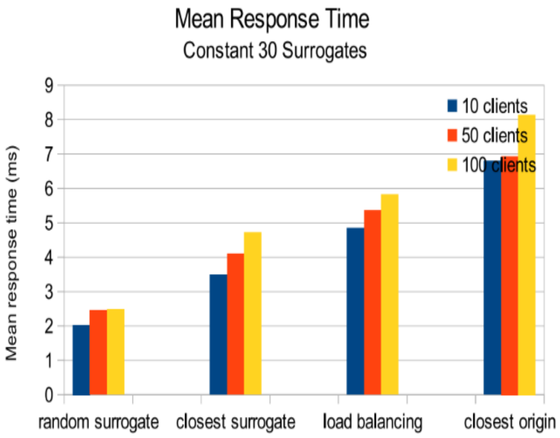

Fig. 4: Mean response time keeping 30 constant surrogates

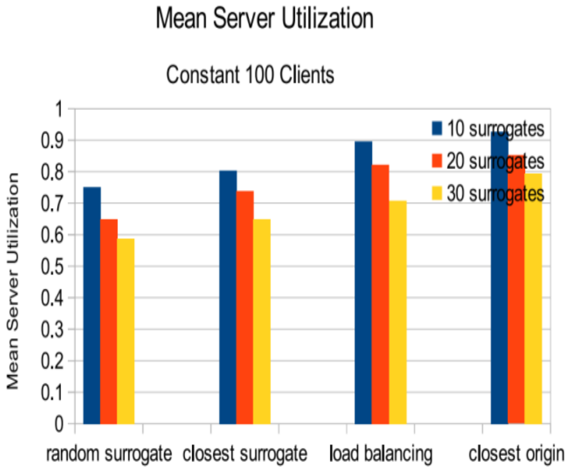

Fig. 7: Mean server utilization keeping 100 constant clients

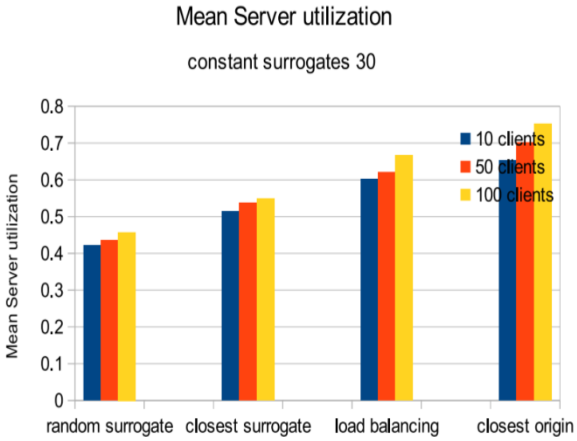

Fig. 6: Mean server utilization keeping 30 constant surrogates

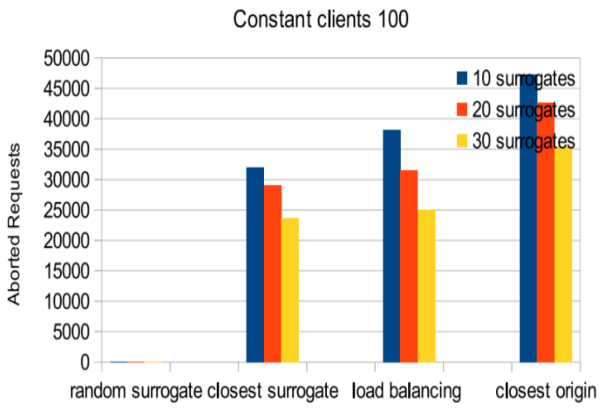

Fig. 7: Aborted requests keeping 100 constant clients

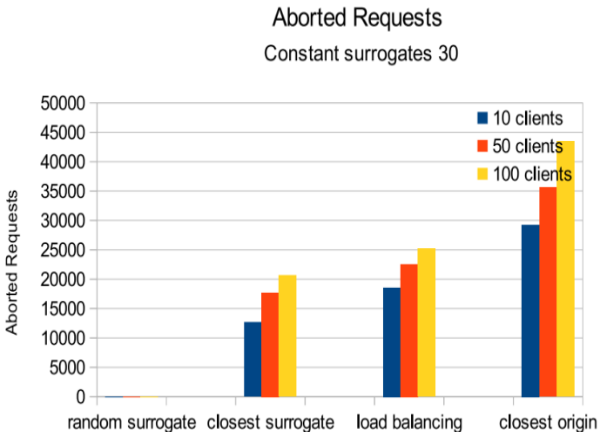

Fig. 8: Aborted requests keeping 30 constant surrogates

(Fig. 3 and 4) shows the results of mean response time for 4 different policies, where number of surogates and clients are varied. Fig. 3 shows the results of experiments performed using 4 different policies, where number of surrogates are varied while keeping number of clients constant to 100 clients. The results suggest that the response time decreases with the increase in number of surrogates. Looking at Fig. 4, we can observe that the mean response time increases with the increase in number of clients irrespective of policies used. However, the scenario where surrogates are placed randomly results in minimum response time.

(Fig. 3) and **(Fig. 4)** shows the results of mean response time for 4 different policies, where number of surogates and clients are varied. **(Fig. 3)** shows the results of experiments performed using 4 different policies, where number of surrogates are varied while keeping number of clients constant to 100 clients. The results suggest that the response time decreases with the increase in number of surrogates. Looking at **(Fig. 4)**, we can observe that the mean response time increases with the increase in number of clients irrespective of policies used. However, the scenario where surrogates are placed randomly results in minimum response time.

(Fig. 5) and **(Fig. 6)** shows the mean server utilization, where number of surogates and clients are varied. **(Fig. 5)** shows the results of experiments performed using 4 different policies, where number of surrogates are varied while keeping number of clients constant to 100 clients. Under such setting, server utilization decreases with increasing number of surrogates. Looking at **(Fig. 6)**, server utilization increases with increasing number of clients. We can observe lowest server utilization when surrogates are randomly placed. The reason for this behaviour is due to the fact the servers are active most of the time and are equally utilized. Hence, as a result minimum mean server utilization has been observed in random surrogate policy, whereas in other policies server utilization is higher.

Similar trend has been observed in **(Fig. 7)** and **(Fig. 8)**, which shows the relationship between the aborted requests and CDN policies. Aborted requests are found to be negligible in case of random surrogate policy followed by closest surrogate policy, load balancing surrrogates and closest origin scenarios.

5. CONCLUSION

Existing infrastructure of CDNs have always been evolved to provide efficient mechanism for the delivery of content to the clients with less jitter, minimum delays and better utilization of existing bandwidth. In this paper, we have analyzed the performance of CDNs using the network simulation software to evaluate the best possible scenario to design an efficient CDN. The experimental setup considers specific number of clients per surrogate server and as a result mean response time, number of aborted requests and mean server utilization are found to be different for different scenario. However, the results suggest that by placing surrogates randomly results in minimum response time, follow by the scenario where surrogates are placed closest to the clients requesting the contents or service. Moreover, these closest surrogate and random surrogate polices also show least aborted requests and low server utilization to deliever specific contents to the clients. Moreover, issue of bandwidth utilization is solved by setting a suitable number of surrogates for local area.

REFERENCES:

Bartolini, N., E. Casalicchio, and S. Tucci, (2004). A walk through content delivery networks. Lecture Notes in Computer Science, 2965, 1-25.

Chen, Y., R. H. Katz, and J. D. Kubiatowicz, (2002). Dynamic Replica Placement for Scalable Content Delivery. In Peer-to-Peer Systems: First International Workshop, IPTPS 2002 Cambridge, MA, USA, Revised Papers 306-318. Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/3-540-45748-8_29

Cronin, E., S. Jamin, C. Jin, A. R. Kurc, D. Raz, and Y. Shavitt, (2002). Constrained mirror placement on the Internet. IEEE Journal on Selected Areas in Communications, 20, 1369-1382. doi:10.1109/JSAC.2002.802066

Day, M. A., and P. Rzewski, (2003). A Model for Content Internetworking (CDI). Internet Engineering Task Force RFC 3466.

Dilley, J., B. M. Maggs, J. Parikh, H. Prokop, R. K. Sitaraman, and W. E. Weihl, (2002). Globally Distributed Content Delivery. IEEE Internet Computing, 6, 50-58. Retrieved from http://dblp.uni-trier.de/db/journals/internet/internet6.html#DilleyMPPSW02

Fortino, G., and W. Russo, (2008). Using P2P, GRID and Agent Technologies for the Development of Content Distribution Networks. Future Gener. Comput. Syst., 24, 180-190. doi:10.1016/j..06.07

Hofmann, M., and L. R. Beaumont, (2005). Content Networking - Architecture, Protocols, and Practice. Elsevier.

Jamin, S., C. Jin, A. R. Kurc, D. Raz, and Y. Shavitt, (2001). Constrained mirror placement on the Internet. INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, 1, 31-40 vol.1. doi:10.1109/INFCOM.2001.916684

Kamiya, Y., T. Shimokawa, F. Tanizaki, and N. Yoshida, (n.d.). Scalable Contents Delivery System with Dynamic Server Deployment.

Katsaros, D., G. Pallis, K. Stamos, A. Vakali, A. Sidiropoulos, and Y. Manolopoulos, (2009). CDNs Content Outsourcing via Generalized Communities. IEEE Transactions on Knowledge and Data Engineering, 21, 137-151. doi:10.1109/TKDE.2008.92

Krishnan, P., D. Raz, and Y. Shavitt, (2000). The Cache Location Problem. IEEE/ACM Trans. Netw., 8, 568-582. doi:10.1109/90.879344

Mulerikkal, J. P. and I. Khalil, (2007). An Architecture for Distributed Content Delivery Network. IEEE International Conference on Networks, 359-364. doi:10.1109/ICON.2007.4444113

Oberheide, J., M. Karir, and Z. M. Mao, (2007). Characterizing Dark DNS Behavior. In Detection of Intrusions and Malware, and Vulnerability Assessment: 4th International Conference, DIMVA Lucerne, Switzerland, 12-13, Proceedings 140-156. Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-73614-1_9

Pallis, G., A. Vakali, K. Stamos, A. Sidiropoulos, D.Katsaros, and Y. Manolopoulos, (2005). A latency-based object placement approach in content distribution networks. Third Latin American Web Congress (LA-WEB'2005), 8. doi:10.1109/LAWEB.2005.3

Sarma, S. S. and S. K. Setua, (2016). Uniform load sharing on a hierarchical content delivery network interconnection model. Innovations in Systems and Software Engineering, 12, 239-248. doi:10.1007/s11334-016-0279-5

Sidiropoulos, A., G. Pallis, D. Katsaros, K. Stamos, Vakali, A. and Y. Manolopoulos, (2008). Prefetching in Content Distribution Networks via Web Communities Identification and Outsourcing. World Wide Web, 11, 39-70. doi:10.1007/s11280-007-0027-8

Silva, F. A., A. Boukerche, T. R. Silva, L. B. Ruiz, E. Cerqueira, and A. A. Loureiro, (2016). Vehicular Networks: A New Challenge for Content-Delivery-Based Applications. ACM Comput. Surv., 49, 11:1--11:29. doi:10.1145/2903745

Suresh, V., B. Venkatesh, and A. Anjaneyulu, (2016). Emerging Cloud based Content Delivery Networks. International Journal of Engineering 1, 54-63. Retrieved from http://scigatejournals.com/publications/index.php/ijep
Ubuntu Content Delivery Networks. (n.d.). Retrieved from http://apmblog.dynatrace.com/2013/12/05/welcome-to-the-show-of-content-delivery-networks-act-1-the-what-and-why/
Ubuntu request-routing-in-cdn. (n.d.). Retrieved from http://www.slideshare.net/sankath/request-routing-in-cdn

Wang, L., V. Pai, and L. Peterson, (2002). The Effectiveness of Request Redirection on CDN Robustness. Proceedings of the 5th Symposium on Operating Systems Design and implementation Copyright Restrictions Prevent ACM from Being Able to Make the PDFs for This Conference Available for Downloading 345-360. Berkeley: USENIX Association. Retrieved from http://dl.acm.org/citation.cfm?id=1060289.1060321