ISSN No.: XXXX-XXXX
Pages 01-08

JOURNAL OF ADVANCED COMPUTER APPLICATIONS

ISSN: XXXX-XXXX (Online)
Vol: 01 **Issue:** 01
Contents available at: https://www.swamivivekanandauniversity.ac.in/jaca/

Edge Server Selection using Request Manager System in Content Delivery Network

Ranjan Kumar Mondal[1] and Ganesh Rajak[2]

1 Department of Computer Science and Engineering, Swami Vivekananda University, Barrack-pore-700121, WB, INDIA

2 Department of Computer Science, University of Kalyani, Kalyani-741235, WB, INDIA

ranjankm@svu.ac.in

ABSTRACT

Cloud computing is a smart way of providing web services to the user from any where by using internet connection on pay-per-use basis. As it uses web 2.0 technology the data travels along different paths through different network devices and finally arrives at the client end by overcoming the network traffic and delays, the network traffic and the associated delays is based on the geographic location of the original web-server, network latency, and the content which is being served. CDN (Content delivery network) is a globally distributed network which caches the static content of the website into its edge servers and serve them to the geographically nearest requesting user. As static content does not require processing so it will be served immediately from its nearest edge server. We proposed an efficient way that employs request manager system over the content delivery network to serve and direct the user request to the nearest edge server and establish the connection between them to transfer the static contents.

Handle on demand network popularity of the Content Delivery Network and solve the flash crowd problem, caching web content at the internet's edge server has been emerged. Therefore finding the nearest edge server to a particular web user is an open research problem and it ensures a faster response time and download time of the requested content due to reduced latency.

Keyword: Cloud computing, Content Delivery Network, Nearest Neighbor Queries, Edge server, Request manager system, Edge server selection.

1. INTRODUCTION

With advancement in the network technology day by day the no of clients are increasing and demanding more smooth and secure services on the way, as on the same way many IT companies are focusing to satisfy the clients with their requirements and trying to invest more on creating new

* Authors for Correspondence

things than to invest in maintenance, so companies are looking for cloud computing. Even though cloud computing has a great effort in allowing unlimited user access to a site and auto scalability but it suffers from network delay in serving the contents to client; as because the data packets has to travel from different routes through different network devices from original webserver to the end user. So the question arises why not we serve the client from their nearest location? What if we can reduce the network latency by putting an edge server to that region? Solution to this a Content Delivery Network (CDN) : It is a geographically distributed network of webservers which caches the web content and deliver the web content to the requested clients based on their geographical locations. When a user sends a requests to the website then the original server redirects the request to the registered CDN, which then serve the static contents such as images ,JavaScript ,CSS files or any other contents which is being embedded in site pages. This approach generally helps in serving the content faster but it has certain limitations such as selecting optimal edge server, and also keeping track of end user request.

In this paper we devise an optimal edge server selection algorithm which selects and stores the results in recently used list, which can be used to find the future edge server selection request within the same algorithm.

2. BACKGROUND STUDY

a. Content delivery network

A content delivery network (CDN) is a large geographically scattered system of servers deployed in multiple data centers across the Internet. The goal of a CDN is to serve content to end-users with high availability and high performance. CDNs serve a large fraction of the Internet content today, including web objects (text, graphics and scripts), downloadable objects (media files, software, and documents), applications (e-commerce, portals), live streaming media, on-demand streaming media, and social networks. When various users from different geographical locations request a website then the CDN selects the nearest edge server to the end user, and delivers the static content, and helps reduce the network latency time which is depicted in Figure 1.

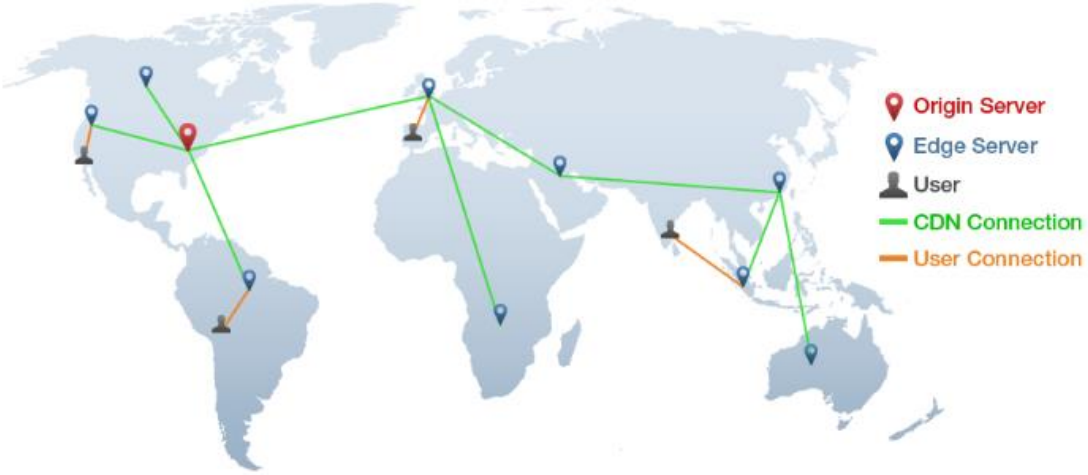

Fig. 1. Content Delivery network

Content Delivery Network (CDN) comprises of a series of servers and Points-of-Presence around the globe, which help in improving the page-load time as well as reduce the download time of the content by bringing it closer to the users. In the simplest way, a CDN can be explained as a set of mirrors for your website, hosted in such a way that the content is served to your targeted audience from the server closest to them.

3. METHODOLOGICAL ASPECTS

a. Server Selection using Request Manager System

In this paper our main goal is to find the nearest edge server to the requesting client, and help serve the static contents to the client. In this approach instead of implementing the algorithm all over the CDN, we have implemented the concept of keeping a request manager system(RMS), which basically consists of hardware (Hard disks, processor, Memory, GPS, network device) and the software which will handle the entire incoming request from users of various geographic locations requesting for various web sites. This system is placed centrally over the CDN and it basically keeps track of all the edge servers within a CDN, it holds all the relevant information of the connected edge servers and the associated databases of the web-sites for which the request is to be handled.

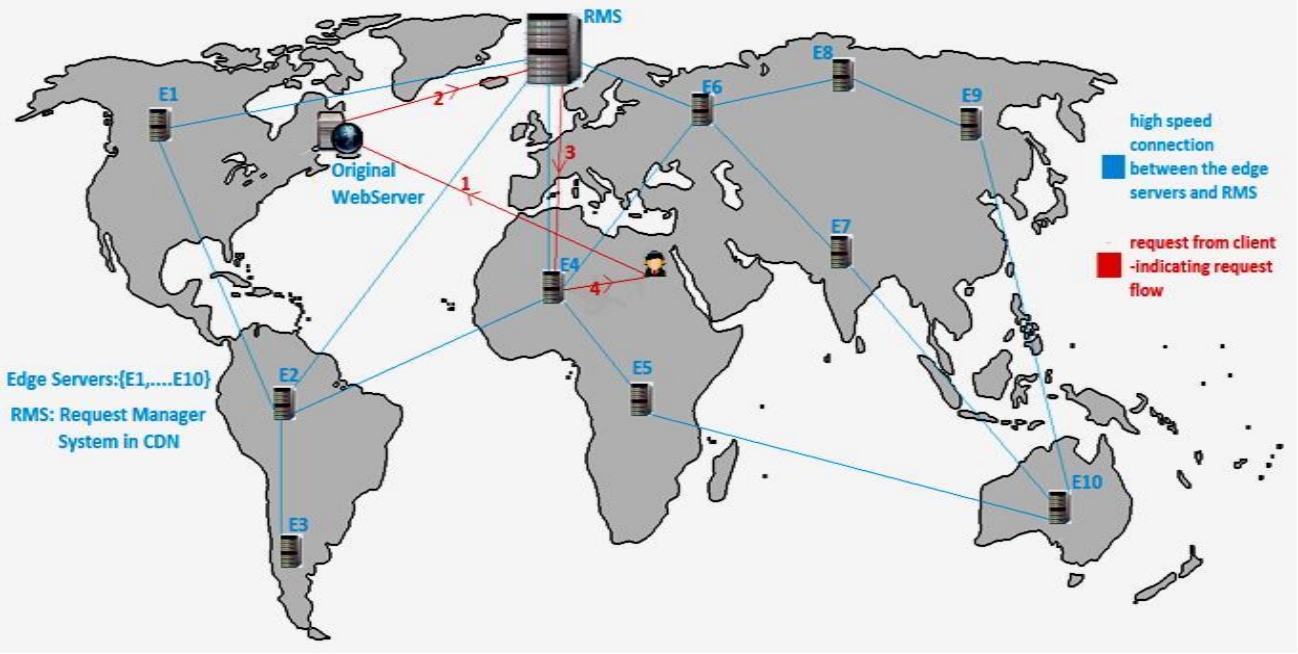

Fig. 2. Edge server selection using RMS

NOTE: Lines connecting the edge server, RMS and Web Server may contain powerful network devices and in between them DNS (Domain Name System as user will type the domain name instead of IP Address) is placed.

We assumed that CDN is registered for the specific domain with DNS and the following requirements of the CDN holds—

1. Request manager system is placed centrally and is connected to each edge server with high speed connection.
2. Request manager system maintains a table of all the edge servers that contains parameters such as workload, network latency, average response time, storage capacity, channel capacity etc. which is periodically updated. And let this table be called as edge parameter table.
3. Each edge server runs the services for periodically broadcasting the average workload, network latency, and storage capacity and also employs a search query service to search for contents within the CDN.
4. Request manager also employ a recently used list, which asynchronously updated whenever a user from unique geographic location and IP address allotted an edge server. This list contains geographic coordinates, IP address and the corresponding selected edge server's IP address.

b. Flow chart

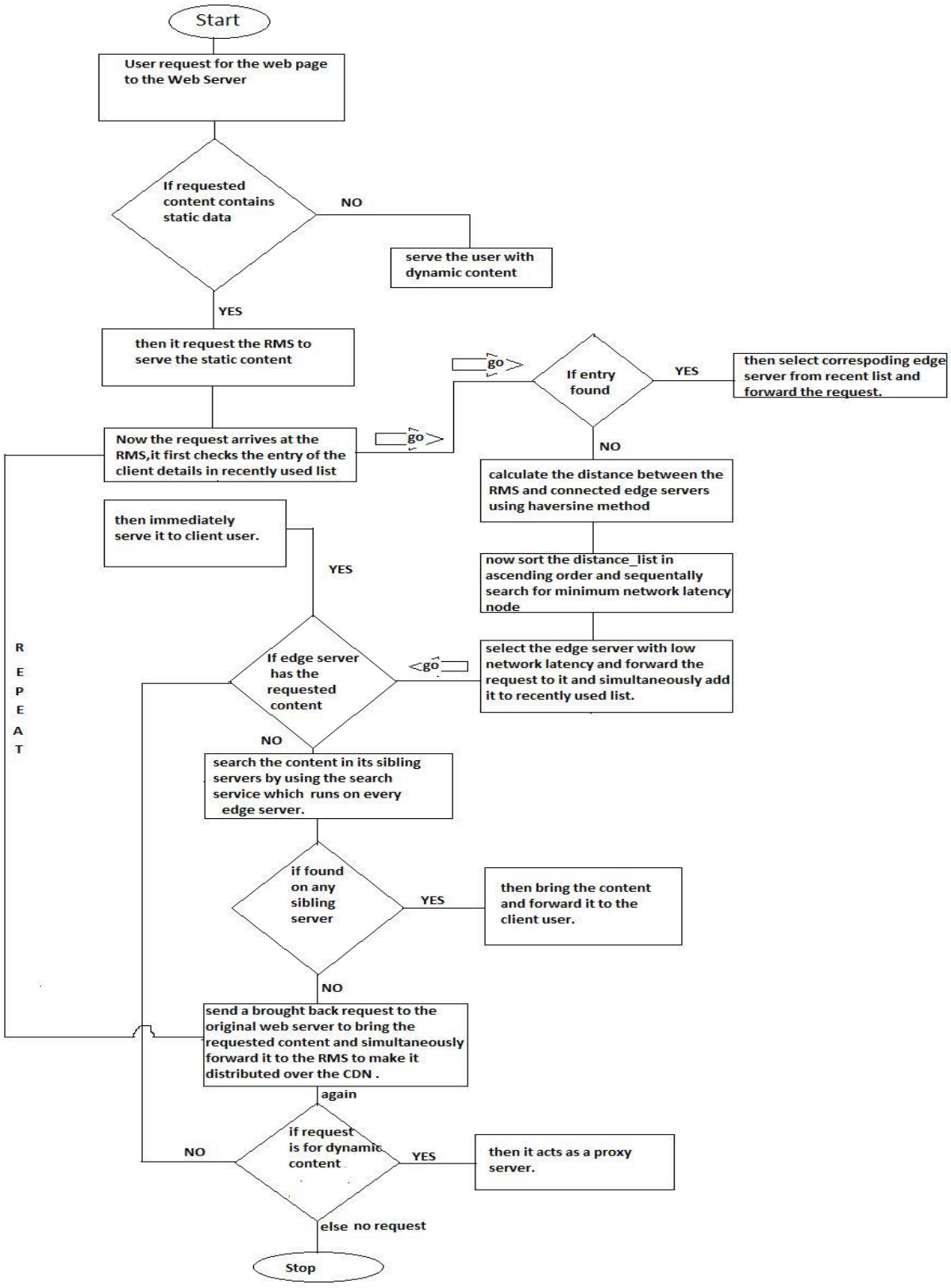

Fig. 3. Flow chart

c. Algorithm for Edge Server selection

1. Let U={u_1,u_2,u_3,u_4…….u_n} be the set of users from different geographic locations.
2. And E={e1,e2,e3,…..en} be the set of edge servers and RM be the request manger in a CDN.
3. When a user from set U sends a request to a specific domain (say www.xyz123.com). Then the DNS finds the address and redirects the request to the original server.
4. When a request arrives at the original web server, it first identifies which type of content is to be served, for static content it sends the request to the request manager system of the registered CDN.
5. Now request manager first identifies the user location i.e. longitude, latitude and IP address and then it looks up the recently used list ,to check whether this user recently requested for any content or not.
   a) If it finds the IP address and associated geo-coordinates in the recently used list then it immediately associate the corresponding edge server, which was last used for serving the content.
   b) Else goto step 6;
6. Now the request comes from a new user
   a) Calculate the distance between the user and edge servers of the CDN using haversine distance calculation method, and store the list in data structure named as **distance_list**(contains edge server's IP address and the users geo-coordinates, and distance between them).
   b) Sort the **distance_list** in ascending order of the distance.
7. Now choose first edge_server from distance_list
   a) For each temp_server in distance_list
      i.If (getNetworkLatency(edge_server)<getNetworkLatency(temp_server)) then
            selected_edge_server = edge_server;
      ii. Else selected_edge_server = temp_server;
      End for
8. After the selection of the edge server, the parameters of the selected_edge_server and the user are written to the recently used list.
9. Now the original request of the user is forwarded to the selected_edge_server and then selected_edge_server transmit the static contents immediately.
10. Now the connection between the selected_edge_server and the user is established.
11. If user again sends a request for either some dynamic content or for static content then

a. If request is for dynamic content then:

i. selected_edge_server acts as the proxy and fetch the dynamic content from the original web server and sends it to the user.

b. If request is for static content then :

i. If content is available then transmit the static content immediately.

ii. Else if it is not available in selected_edge_server then it will search for content by sending a search query to its siblings and if found redirects it to the user.

iii. Else if it is not available within the CDN, then a brought back request is sent to the original server and the static content is fetched, simultaneously content is distributed over the CDN with the help of RMS.

12. For each request repeat step 5 to step 11
13. End

For adding and searching content in the recently used list we can use hashing and for sorting the distance list we can use any stable sorting algorithm.

4. CONCLUSION

Now days CDN becomes an efficient way of delivering static content to the client user, it also helps in providing live streaming, video-on-demand(VOD) ,faster data sharing (used by online storage provider) services in the world of internet. In our proposed work we used a centrally dedicated request handler system (RMS) which keeps track of all the connected edge servers of a CDN and monitor the status of each edge servers, this will boost the network performance and the average workload of the edge servers for the following reasons-

1. RMS protects from unnecessary network traffic to the edge servers by running the edge server selection algorithm centrally on RMS.
2. RMS keeps track of all the edge servers and monitors the status of the edge server and helps in handling faults.
3. Easy to extend the number of edge servers and for any modification to algorithm needs to be modified on RMS only, as algorithm runs centrally.
4. It also offers different running services such as FTP, broadcaster, SMTP etc and a special service called brought back request, which helps in caching the file from the original web server and distribute over the CDN, if and only if the content doesn't exist within the CDN. This works automatically as it is a part of algorithm.
5. This methodology serves faster to the frequent users as they are stored in recently used list, no need to run the whole algorithm.

Finally, we conclude that our proposed work can handle huge requests from different users who are keen to access static contents such as videos, images, or any other files. We assume that this CDN is hosted either as a cloud system to provide CDN as-a-service or hosted at any cloud vendors. And our approach is as practicable to be implemented to help serve better.

REFERENCE

1. Nygren, R. K. Sitaraman and J. Sun, ―The Akamai Network: A Platform for High-Performance Internet Applications,‖ ACM SIGOPS OSR, vol. 44, no. 3, pp. 2–19, 2010.
2. T. Repantis, J. Cohen, S. Smith and J. Wein, ―Scaling a Monitoring Infrastructure for the Akamai Network,‖ ACM SIGOPS Operating Systems Review, vol. 44, no. 3, July 2010.
3. J. Parikh, H. Prokop, R. Sitaraman, J. Dilley, B. Maggs and B. Weihl, ―Globally Distributed Content Delivery,‖ IEEE INTERNET COMPUTING, pp. 50-58, September-October 2002
4. F. Aurenhammer, ―Voronoi Diagrams—A Survey of a Fundamental Geometric Data Structure,‖ ACM Computing Surveys, vol. 23, no. 3, pp. 345–405, September 1991.
5. P. Guillaume, and S. M. Van "Globule: a collaborative content delivery network." IEEE Communications Magazine 44.8 (2006): 127-133.
6. K. P Al-Mukaddim Khan, and B. Rajkumar "A taxonomy and survey of content delivery networks." Grid Computing and Distributed Systems Laboratory, University of Melbourne, Technical Report (2007).
7. S. Saroiu, Gummadi, K. P., Dunn, R. J., Gribble, S. D., & Levy, H. M. (2002). An analysis of internet content delivery systems. ACM SIGOPS Operating Systems Review, 36(SI), 315-327.

8. D. Cranor, C, M. Green, C. Kalmanek, D. Shur, S. Sibal, E. Van der Merwe, J.&J. Sreenan, C. (2001). Enhanced streaming services in a content distribution network. Internet Computing, IEEE, 5(4), 66-75.
9. Wikipedia “Content Delivery Network”.