International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 4, Issue 6, November - December 2015 ISSN 2278-6856

PROPOSED ARCHITECTURE ON- A HYBRID APPROACH FOR LOAD BALANCING WITH RESPECT TO TIME IN CONTENT DELIVERY NETWORK

[1]PrashantRewagad(Guide), [2]ArchanaChitte

[1]Head of Department Department of Computer Science and Engg GHRIEM Jalgaon

[2]ME(CSE) student Department of Computer Science and Engg GHRIEM Jalgaon

ABSTRACT

Content Delivery Network (CDN), is a network having a large distributed servers deployed in multiple data centres on the internet. The goal of this network is to serve requests from users providing high performance and less delay and QoS. The CDN reduces the risks of system failure using redirection to many replica server means copies of the contents are copied onto several mirrored web servers. So the client can directly access any server with less delay. Sometimes the servers are allocated to provide the service to request of users without having the prior knowledge about load onto that server. It may congestion occur or discard the users request. So many load balancing algorithms are used to maintain the balance on the mirrored servers. In this paper, we proposed a hybrid load balancing architecture which is a combination of least loaded algorithm and random balancing algorithm. The new term is defined here, which is a token. Token is used for maintaining the status of replica servers. According to the token information the replica server is assign to the particular users request.

Keywords- Content Delivery Network, Load balancing algorithm, distributed system.

I. INTRODUCTION

1. Content Delivery Network (CDN)

The size of web contents has grown day by day. Today an internet user also increases. Contents of file sizes range from a few megabytes to several gigabytes. Content providers use Content Delivery Network (CDN) such as Akamai, Limelight, Coral CDN to support files to millions of Internet user. A CDN combines a content delivery infrastructure, a request-routing infrastructure, a distribution infrastructure and accounting infrastructure. Usually, a CDN consists of an original server (called *back-end server*) containing new data to be diffused, together with one or more distribution servers, called *surrogate servers*.[1] The surrogate servers are actively updated by the original server(back-end server). These Surrogate servers are generally used to store static data, while dynamic information (i.e., data that change in time) is just stored in a small number of original servers. In some systems, the server is called *redirector*, which dynamically redirects client requests based on selected policies. Limelight, CDNetworks and Akamai are popular commercial CDN projects that support to the most popular Internet and media companies. Several academic projects have also been proposed, like CoralCDN at New York University and CoDeeN at princeton University, both running on the PlanetLab1 test bed[13].

A CDN is a collection of network devices arrange for delivery content to end users. A CDN network could be implemented in many architectures and topologies, which can be centralized, hierarchical, infra-structure with administrative control and decentralized. ContentDeliveryNetwork (CDN) is a geographically distributed system deployed in multiple data centres on the Internet [1]. CDN contains a replica server which provides fast and reliable services. The CDN term was introduced in the 90's. The motive behind is to provide internet services with reliable, fast performance, scalability, replication and load balancing [4]. When a single server provides services to the high volume traffics, like e-commerce sites, may slowdowns the server. According to research, if the response time for a web request exceeds 8 sec, about 30% of users leave the request. The increase in response time is directly related to performance loss, congestion and a large number of users reloading the website, making access to the website worse.[15] The replication of websites contents

From different locations are 1) reduce the response time to the nearest user because the server assign to the user which is nearest to the geographically situated, and (2) balance the load among multiple servers. A common approach is to redirect the user to the server closest to him, thus minimizing the bandwidth used, depending on the server's load; this way one can get a shorter response time [3]. One of the major problem related to CDN is, which replica server must be used? The closest server to the user is not always the best [3]. Instead, a set of parameters could be considered during this selection process, such as distance, speed, available bandwidth and server load. This type of algorithm, also known as request routing algorithm, can be divided into two categories: adaptive algorithms and no adaptive algorithms [4]. In adaptive algorithms, the choice is made based on the server's status, requiring constant

monitoring. In no adaptive algorithms, the choice is based on heuristics, and then a lightweight processing by not requiring monitoring. This paper proposes a new hybrid algorithm based on least loaded algorithm and random balancing algorithm to select the best replica server. This algorithm considers the following parameters: (1) size of requesting service queue for each replica server; (2) time require to provide the service to a request from a given URL on the replica server, (3)time require to replica server. The rest of the paper is structured as follows. In section II, we present Request Routing Algorithms, Section III, present some, problems related to static and dynamic algorithms, Section IV introduces the concepts of proposed architecture and hybrid approach and Section V concludes the paper.

II.Request Routing Algorithms

Distributing the client request to the active servers for balancing the load using request routing algorithm in a CDN among the servers in the distribution network. Different mechanisms have been used in the system. They can be categorized by static or dynamic depending on the technique used for server selection [8][9]. In Static algorithms, server is selected without having any information about the status of the system at the end of decision.

Static algorithms are not requiring any special retrieve mechanism in the system. These are the fastest algorithms which gives the fast and safe solution so they do not require any selection process[7]. They does not used any particular selection process that's why they are not able to effectively solve the problems of anomalous events like flash crowds [14].

To resolve the problem of Flash Crowd, Dynamic load-balancing strategies are an alternative solution. This strategy makes use of information coming either from the network or from the servers in order to improve the request transfer process[5]. The process of selecting the applicable server is completed through a collection and consequent analysis of several parameters which are extracted from the network fundamentals. Therefore, a data alteration process between the servers is required, which is compulsory incurs in a communication overhead.

III. Problems related to Static Algorithms

The data cannot be change frequently, at that time the static algorithms will be used in the Content Delivery Network. The static algorithms are Round Robin and Random Balancing[10]. Using this algorithm the congestion may occur at the server side because there is no any provision or not any record about the server which is heavily loaded or least loaded. In this algorithm just assign the server to the client request without any necessary information. So the static algorithms are less used. These are suitable for small network services.

Problems related to Dynamic Algorithms

In the dynamic algorithms web contents are frequently changes according to user requirement. At that time dynamic algorithms will be used. The algorithms are Least Loaded, Two Random Choices[6]. Any server can be assign to the clients request based on the least loaded information given by the surrogate server to the back end server. These algorithms are useful but there is no provision about the status of surrogate servers based on that the back end server will assign the least loaded surrogate server.

IV.PROPOSED ARCHITECTURE

The most important factor related to network traffic is congestion control. When the load on the network is greater than the capacity of network then the congestion is occur. The congestion occurrences leads to longer packet delay or sometimes it may loss the packets[11]. To control congestion and improving the efficiency of the use of resources needs to select an optimal server according to the load balancing. The selection of server is decided by the load balancing algorithm which is nearest to the user[12].

In our proposed architecture, we proposed a hybrid approach for balancing the load. This hybrid approach is a combination of Random balancing algorithm and Least Loaded algorithm. In this work, introduce a new dynamic hybrid approach with new term use i.e. token. This work is a combination of two parts. In the first, this is a combination of Random Balancing algorithm and Least Loaded balancing algorithm. The old scenario of the algorithm was that, the randomly server is selected for user's request without considering the balanced on that server which is based on static strategy. The Random selection of server may leads to congestion and delay in packet delivery[13]. In the Least Loaded algorithm, the arriving request assign the Least Loaded servers. Unfortunately, its circumstances to be saturated until a new message is circulate. The alternative solution is define in this work i.e. Hybrid approach. In hybrid approach, the Random server is accessible by user with consideration of load on that server. If that server is loaded, the another server is selected randomly. The second approach introduces the token term. Token maintain the status of the replica servers. According to the token status, the user request will be redirected to the replica server via the main server. Initially the user request is send to the main server which will maintain the queue. The server having requires less time for processing assigns replica server to the user. According to the status of token, which consist of status of replica server, assign the replica server to the user.

A.Random Balancing Algorithms

The load balancing method randomly distributes load across the servers available, picking one via random number generation and sending the current connection to it. Random numbers are numbers that occur in a sequence such that two conditions are met: (1) The values are uniformly distributed over a defined interval or set and (2) It is impossible to predict future values based on past or present ones. Random numbers are important in statistical analysis and probability theory. Based on this theorem, the random balancing algorithm is work[14].

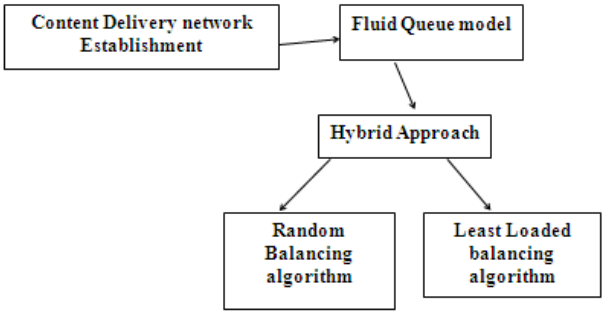
B.Least Loaded Algorithms

Load on the particular server decides based on the resources utilisation of that server. In Least Loaded algorithm, the server is selected which is least loaded. Status about the utilisation of resources can be send by replica server to the main server after providing the service to the every user.

C. Token Passing Technique

The new term is proposed i.e. Token. To provide a service to the requesting user from a particular server is depends on the Token information. Token is a variable which is updated after some time period by all the surrogate servers.The token is circulated after some time periods by the main server to getting the current status of the replica server. Every surrogate server is updates the Token by putting the load information . On the basis of this information, the least loaded surrogate servers are indentified b the main server. This process is totally carried out by the main server. To choose the least loaded surrogate servers, the algorithm is apply on the back end server. After choosing the least loaded servers the random balancing algorithm is apply. Random balancing algorithm works on the set of values. These values are nothing but the set of least loaded servers. Here servers addresses are consider for the set of values. To get the random server number, after applying the random balancing algorithm. That server is assign to the user for providing the service. It also consider the geographically nearest server to the users for fast service provide.

D.Data Flow Diagram of Proposed Architecture



V.Conclusion

In this paper, we proposed a new load balancing approach for CDN to choose the replica server for improving the QoS, reliability and reduce the packet drops and less delay for data transmission. The main benefit of this approach, is to reduce the discarding of users request if the replica servers are not available. This proposed architecture may provide the better performance in real time applications.

REFERENCES

[1]. P. Rewagad, A. Chitte, " Survey on Hybrid approach for Load Balancing With Respect to Time in Content Delivery Network", IJIFR,vol 2, Issue 10, pp.3679-3686,jun 2015.

[2]. S. Manfredi,F. Oliviero,S.P. Romano,"A Distributed control law for load balancing in Content Delivery Netwroks", IEEE/ACM trans.vol.21,no.1,pp. 55-68,Feb 2013.

[3]. F. Blanchini, R. L. Cigno, and R. Tempo, "Robust rate control for integrated services packet networks," vol. 10, no. 5, pp. 644–652, Oct. 2002.

[4]. T. Queiroz, M. Fernandez, "Fuzzy Redirection Algorithm for Content Delivery Network (CDN)",ICN 2013,pp:137-143, 2013

[5]. N. Ball, P. Pietzuch,"Distributed Content Delivery using Load-Aware Network Coordinates", ACm 2008.

[6]. Zhihui Lu, Ye Wang and Yang Richard Yang, "An Analysis and Comparison of CDN-P2P hybrid Content Delivery System and Model", VOL. 7, NO. 3, MARCH 2012.

[7]. C. V. Hollot, V. Misra, D. Towsley, andW.Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," vol. 47, no. 6, pp. 945–959, Jun. 2002.

[8]. M. D. Mitzenmacher, "The power of two choices in randomized load balancing," IEEE vol. 12, no. 10, pp. 1094–1104, Oct. 2001.

[9]. R. L. Carter and M. E. Crovella, "Server selection using dynamic path characterization in wide-area networks," Apr. 1997, vol. 3, pp. 1014–1021.

[10]. D. M. Dias, W. Kish, R. Mukherjee, and R. Tewari, "A scalable and highly available Web server," Feb. 1996, pp. 85–92.

[11]. C. V. Hollot, V. Misra, D. Towsley, and W. bo Gong, "A control theoretic analysis of red," 2001, pp. 1510–1519.

[12]. M. Dahlin, "Interpreting stale load information," vol. 11, no. 10,pp. 1033–1047, Oct. 2000.

[13]. Z. Xiang, Q. Zhang, W. Zhu, Z. Zhang, Y. Zhang, "Peer-to-Peer Based Multimedia Distribution Service", IEEE trans on multimedia, vol.6,no.2,April 2004.

[14]. H. Yin, X. Liu, G. Min and C. Lin, "Content Delivery Networks: A Bridge between emerging applications and future IP networks", IEEE Netw., vol. 24, no. 4, pp- 52-56, Jul-Aug 2010.

[15]. B. Davison, "A web caching primer," Internet Computing, IEEE, vol.5,no.4,pp.38-45,jul/aug 2001