A Distributed Control Law for Load Balancing in Content Delivery Networks

Mr. Shendage Swapnil Sunil[1]
PG Student, Department of CSE,SIETC,
Paniv, Solapur University,Maharashtra,
India[1]
swapnil.shendage@gmail.com

Prof. Dhainje Prakash B.[3]
Vice-Principal, SIETC, Paniv, Solapur
University,Maharashtra, India[3]

Dr. Deshmukh Pradeep K[2]
Principal, SIETC, Paniv,
SolapurUniversity,Maharashtra, India[2]
principalsietc@gmail.com

***Abstract*—**Large Internet-scale distributed systems deploy hundreds of thousands of servers in thousands of data centers around the world. Internet-scale distributed system to have emerged in the past decade is the content delivery network (CDN, for short) that delivers web content, web and IP-based applications, downloads, and streaming media to end-users (i.e., clients) around the world. This paper focuses on the main research areas in the field of CDN, pointing out the motivations, and analyzing the existing strategies for replica placement and management, server measurement, best fit replica selection and request redirection. In this paper, I face the challenging issue of defining and implementing an effective law for load balancing in Content Delivery Networks. A formal study of a CDN system, carried out through the exploitation of a fluid flow model characterization of the network of servers. This result is then leveraged in order to devise a novel distributed and time-continuous algorithm for load balancing.

Keywords: *Content Delivery Network, Fluid flow model, Load balancing Algorithm.*

*****

I. INTRODUCTION

The over-evolving nature of the Internet brings new challenges in managing and delivering content to users, since for example, popular Web service soften suffer congestion and bottlenecks due to the large demands posed on their services.

A Content Delivery Network (CDN) is a collaborative collection of network elements spanning the Internet, where content is replicated over several mirrored Web servers in order to perform transparent and effective delivery of content to the end users. Collaboration among distributed CDN components can occur over nodes in both homogeneous and heterogeneous environments.

The typical functionalities of a CDN include: • Request redirection and content delivery services, to direct a request to the closest suitable CDN cache server using mechanisms to bypass congestion, thus overcoming flash crowds or SlashDot effects.

• Content outsourcing and distribution services, to replicate and cache content from the origin server to distributed Web servers.

• Content negotiation services, to meet specific needs of each individual user (or group of users).

• Management services, to manage the network components, to handle accounting, and to monitor and report on content usage.

Figure 1.1 shows the model of a CDN where the replicated Web server clusters spanning the globe are located at the edge of the network to which end users are connected. A CDN distributes content to a set of Web servers, scattered over the globe, for delivering content to end users in a reliable and timely manner. The content is replicated either on-demand when users request for it, or it can be replicated beforehand, by pushing the content to the distributed Web servers.A user is served with the content from the nearby replicated Web server.

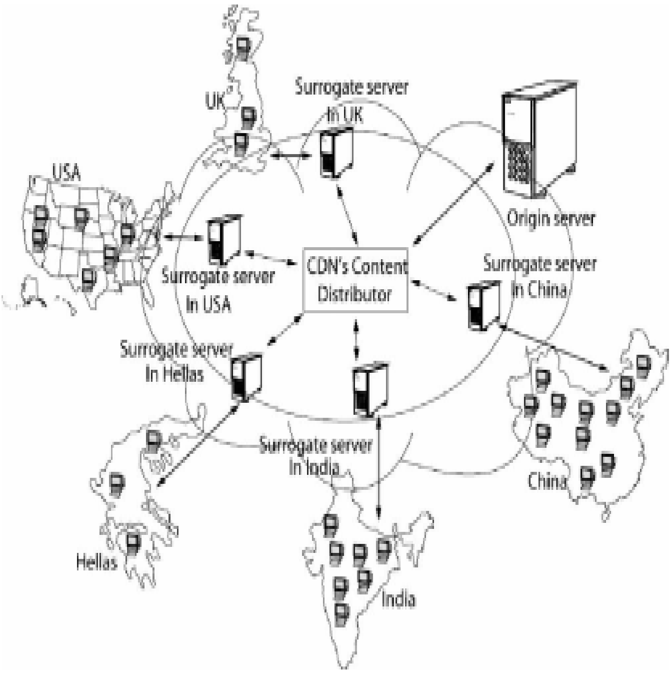
UK
Surrogate server In UK
USA
Origin server
Surrogate server In USA
CDN's Content Distributor
Surrogate server In China
Surrogate server In Hellas
Surrogate server In India
China
Hellas
India

FIG.1.1 CONTENT DELIVERY NETWORK

The three main entities in a CDN system are the following:

1. content provider
2. CDN provider,
3. end users.

A content provider or customer is one who delegates the Uniform Resource Locator (URL) name space of the Web objects to be distributed. The origin server of the content provider holds those objects. A CDN provider is a proprietary organization or company that provides infrastructure facilities to content providers in order to deliver content in a timely and reliable manner. End users or clients are the entities who access content from the content provider's Web site.

CDN providers use caching and/or replica servers located in different geographical locations to replicate content. CDN cache servers are also called edge servers or surrogates. The edge servers of a CDN are called Web cluster as a whole. CDNs distribute content to the edge servers in such a way that all of them share the same content and URL. Client requests are redirected to the nearby optimal edge server and it delivers requested content to the end users. Thus, transparency for users is achieved

II. LOAD-BALANCING STRATEGIES

The challenging issue of defining and implementing an effective law for load balancing in Content Delivery Networks.The content delivery networks technique is one of the successful virtual networks rapidly developed over the last decade with the specific advantage of optimizing the Internet. Nowadays, the CDN has become one of the most important parts of the Internet architecture for content distribution. In this article I highlight the innovative technologies in CDNs and present their evolution triggered by ever newer emerging applications. By resenting in-depth discussion about the architecture, challenges, and applications of CDNs, I demonstrate their importance for the future Internet.

Akamai's DNS-based load balancing system continuously monitors the state of services and their servers and networks. To monitor the entire system's health end to end, Akamai uses agents that simulate the end user behaviour by downloading Web objects and measuring their failure rates and download times. Akamai uses this information to monitor overall system performance and to automatically detect and suspend problematic data centres or servers.

Request-routing mechanisms inform the client about the selection of replica server generated by the request-routing algorithms. Several mechanisms have been proposed in the literature. They can usually be classified as either *static* or *dynamic*, dependingon the policy adopted for server selection [20].Static algorithms select a server without relying on any information about the status of the system at decision time. Static algorithms do not need any data retrieval mechanism in the system, which means no communication overhead is introduced. These algorithms definitely represent the fastest solution since they do not adopt any sophisticated selection process. However, they are not able to effectively face anomalous events like flash crowds.

Dynamic load-balancing strategies represent a valid alternative to static algorithms. Such approaches make use of information coming either from the network or from the servers in order to improve the request assignment process. The election of the appropriate server is done through a collection and subsequent analysis of several parameters extracted from the network elements. Hence, a data exchange process among the servers is needed, which unavoidably incurs in a communication overhead.

III. FLUID QUEUE MODEL

Now introduce a continuous model of a CDN infrastructure, used to design a novel load balancing law. The CDN can be considered as a set of servers each with its own queue. I assume a fluid Model approximation for the dynamic behavior of each queue. I extend this model also to the overall CDN system. Such approximation of a stochastic system.

Let $q_i(t)$ be the queue occupancy of server *i* at time *t*. Iconsider the instant arrival rate α_i(t) and the instant service rate δ_i(t). The fluid model (Fig. 2) of CDN servers' queues is given by

$$\frac{dq_i(t)}{dt} = \dot{q}_i(t) = \alpha_i(t) - \delta_i(t) \quad \text{...... (1)}$$

For *i* = 1…..N.

Equation (1) represents the queue dynamics over time. In particular, if the arrival rate is lower than the service rate, I observe a decrease in queue length. On the other hand, the queue increases whenever the arrival rate is greater than the service rate. In the latter case, the difference in (1) represents the amount of traffic exceeding the available system's serving rate.

In DNS-based request-routing, a domain name has multiple IP addresses associated to it. When an end user's content request comes, the DNS server of the service provider returns the IP addresses of servers holding the replica of the requested object. The client's DNS resolver chooses a server among these. To decide, the resolver may issue probes to the servers and choose based on response times to these probes. It may also collect historical information from the clients based on previous access to these servers. Both full and partial-site CDN providers use DNS redirection. The performance and effectiveness of DNS-based request-routing has been examined in a number of recent studies. The advantage of this approach is the transparency as the services are referred to by means of their DNS names, and not their IP addresses. DNS-based approach is extremely popular because of its simplicity and independence from any actual replicated service. Since it is incorporated to the name resolution service it can be used by any Internet application [89]. In addition, the ubiquity of DNS as a directory service provides advantages during request-routing. The disadvantage of DNS-based request-routing is that, it increases network latency because of the increase in DNS lookup times. CDN administrators typically resolve this problem by splitting CDN DNS into two levels (low-level DNS and high-level DNS) for load distribution. Another limitation is that DNS provides the IP address of the client's Local DNS (LDNS), rather than the client's IP address

IV. DISTRIBUTED LOAD-BALANCING ALGORITHM

In this section, I want to derive a new distributed algorithm for request balancing that exploits the results. It is a hard task to define a strategy in a real CDN environment that is completely compliant with the model proposed. As a first consideration, such a model deals with continuous-time systems, which is not exactly the equal to the traffic received at node from node if no requests are lost during the redirection process.

A. ANALYSIS OF ALGORITHM DESIGN

In this project, I want to derive a new distributed algorithm for request balancing that exploits the results are presented. First of all, I observe that it is a hard task to define a strategy in a real CDN environment that is completely compliant with the model proposed. As a first consideration, such a model deals with continuous-time systems, which is not exactly the case in a real packet network where the processing of arriving requests is not continuous over time. The objective is to derive an algorithm that presents the main features of the proposed load-balancing law and arrives at the same results in terms of system equilibrium through proper balancing of servers‟ loads.

B. ALGORITHM DESCRIPTION

The implemented algorithm consists of two independent parts:

1. a procedure that is in charge of updating the status of the neighbours‟ load,

2. a mechanism representing the core of the algorithm, which is in charge of distributing requests to a node’s neighbours.

Even though the communication protocol used for status information exchange is fundamental for the balancing process.

I implemented a specific mechanism: I extended the HTTP protocol with a new message, called *CDN*, which is periodically exchanged among neighbouring peers to carry information about the current load status of the sending node. Naturally, a common update interval should be adopted to guarantee synchronization among all interacting peers. For this purpose, a number of alternative solutions can be put into places, which are nonetheless out of the scope of the present work.

Though clients are involved in our proposed system network they have no significant role other than requesting for service to the closest server. All thesurrogate servers initialized to handle the request raised by the client. Our proposed algorithm will enhance the functionality of the surrogate servers and mainly the overall system performance as whole.

Let the request queue maximum capacity be Qmax, and the Qi(t) be the queue occupied by server i at time consider the arrival rate be ai(t) service rate id gi(t)

Then I have exchange of request among the server node, which is given by following equation;

g(Qi(t)) = Σ aij(t) ………….. (1)
jϵNei
for i = 1…n.

Nei= {adjacent j of node i}, and aij(t) takes the portion of request injected from node i into node j. The above equation works on the principle of request redirecting by a high loaded server i to a neighbouring less loaded server j. The neighbouring server j will handle the request behalf of server i. Equation (1) can be written in term associated with client incoming request at server i and term associated with request redirected from server to its neighbours

Σ.aij(t)= Σ.aij(t) + Σ.aij(t) ..……….(2)
jϵNeijϵNe+ijϵNe-i

Ne+i(t) ={ adjacent j of server node i: Qj< Qi } and Ne-i(t) ={ adjacent j of server node i: Qj> Qi } are set adjacent server whose queue is respectively less loaded and high loaded than queue at server i**.**

V. BALANCING PERFORMANCE

I need to connect 10 server nodes to the interconnected network, also 10 client nodes, each of the client node should connect to a single server. I need to model each server nodes as a Ma/Ma/1 queue with a rate of service of Si request rate of ai. For every second t, the server exchange its status information data to its neighboring server at the same time it gets its information table updated. By this for every standard time interval server node will do the status update process. So each sever in network have the knowledge of load in the network. So this distributed network works fine in the simulator because each individual server node have complete status of network. At last the flash crowd scenario, simulations demonstrate that our proposed well performs the analyzed existing algorithms in terms of overall system performance that is availability, response time and queue length handling as shown in fig.5.1.

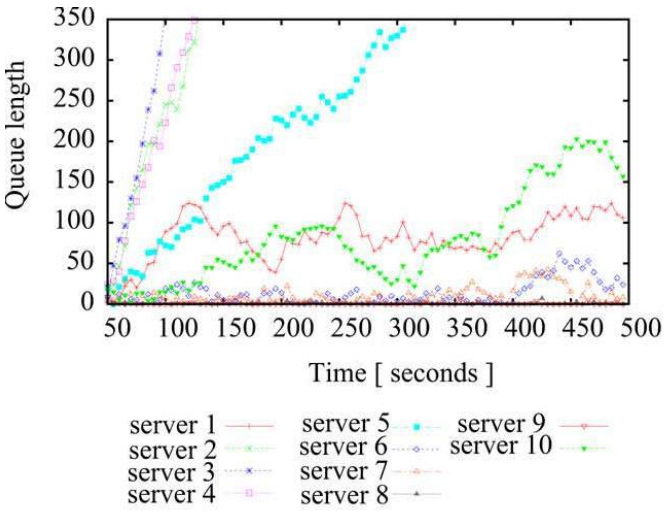

Fig.5.1 system performance

VIII. References

[1] S. Manfredi, F. Oliviero, and S. P. Romano, “Distributed management for load balancing in content delivery networks,” in *Proc. IEEE GLOBECOM Workshop*, Miami, FL, Dec. 2010, pp. 579–583.

[2] 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.

[3] J. D. Pineda and C. P. Salvador, “On using content delivery networks to improve MOG performance,” *Int. J. Adv. Media Commun.*, vol. 4, no. 2, pp. 182–201, Mar. 2010.

[4] D. D. Sorte,M. Femminella, A. Parisi, andG. Reali, “Network delivery of live events in a digital cinema scenario,” in *Proc. ONDM*,Mar. 2008, pp. 1–6.

[5] Akamai, “Akamai,” 2011 Online. Available: http://www.akamai.com/index.html

[6] Limelight Networks, “Limelight Networks,” 2011 Online. . Available: http://.uk.llnw.com

[7] CDNetworks, “CDNetworks,” 2011 Online. . Available: http://www.us.cdnetworks.com/index.php

[8] Coral, “The Coral Content Distribution Network,”2004Online Available: http://www.coralcdn.org

[9] Network Systems Group, “Projects,” Princeton University, Princeton, NJ, 2008 Online. . Available: http://nsg.cs.princeton.edu/projects

[10] A. Barbir, B. Cain, and R. Nair, “Known content network (CN) request- routing mechanisms,” IETF, RFC 3568 Internet Draft, Jul. 2003 Online. . Available: http://tools.ietf.org/html/rfc3568

[11] T. Brisco, “DNS support for load balancing,” IETF, RFC 1794 Internet Draft, Apr. 1995 Online. . Available:http://www.faqs.org/rfcs/rfc1794.html

[12] M. Colajanni, P. S. Yu, and D. M. Dias, “Analysis of task assignment policies in scalable distributedWeb-server systems,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 9, no. 6, pp. 585–600, Jun. 1998.

[13] D. M. Dias, W. Kish, R. Mukherjee, and R. Tewari, “A scalable and highly availableWeb server,” in *Proc. IEEE Comput. Conf.*, Feb. 1996, pp. 85–92.

[14] C. V. Hollot, V. Misra,D. Towsley, and W.Gong, “Analysis and design of controllers for AQM routers supporting TCP flows,” *IEEE Trans. Autom. Control*, vol. 47, no. 6, pp. 945–959, Jun. 2002.

[15] C. V. Hollot, V. Misra, D. Towsley, and W. bo Gong, “A control theoretic analysis of red,” in *Proc. IEEE INFOCOM*, 2001, pp. 1510–1519.

[16] J. Aweya, M. Ouellette, and D. Y. Montuno, “A control theoretic approach to active queue management,” *Comput.Netw.*, vol. 36, no. 2–3, pp. 203–235, Jul. 2001.

[17] F. Blanchini, R. L. Cigno, and R. Tempo, “Robust rate control for integrated services packet networks,” *IEEE/ACM Trans. Netw.*, vol. 10, no. 5, pp. 644–652, Oct. 2002.

[18] V.Misra,W. Gong, W. bo Gong, and D. Towsley, “Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to red,” *Proc. ACM SIGCOMM*, pp. 151–160, 2000.

[19] D. Cavendish, M. Gerla, and S. Mascolo, “A control theoretical approach to congestion control in packet networks,” *IEEE/ACM Trans. Netw.*, vol. 12, no. 5, pp. 893–906, Oct. 2004.

[20] V. Cardellini, E. Casalicchio, M. Colajanni, and P. S. Yu, “The state of the art in locally distributedWeb-server systems,” *Comput. Surveys*, vol. 34, no. 2, pp. 263–311, Jun. 2002.

[21] Z. Zeng and B. Veeravalli, “Design and performance evaluation of queue-and-rate-adjustment dynamic load balancing policies for distributed networks,” *IEEE Trans. Comput.*, vol. 55, no. 11, pp. 1410–1422, Nov. 2006.

[22] V. Cardellini, M. Colajanni, and P. S. Yu, “Request redirection algorithms for distributed Web systems,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 14, no. 4, pp. 355–368, Apr. 2003.

[23] M. Dahlin, “Interpreting stale load information,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 11, no. 10, pp. 1033–1047, Oct. 2000.

[24] R. L. Carter and M. E. Crovella, “Server selection using dynamic path characterization in wide-area networks,” in *Proc. IEEE INFOCOM*, Apr. 1997, vol. 3, pp. 1014–1021.

[25] M. D. Mitzenmacher, “The power of two choices in randomized load balancing,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 12, no. 10, pp. 1094–1104, Oct. 2001.

[26] C.-M. Chen, Y. Ling, M. Pang, W. Chen, S. Cai, Y. Suwa, and O. Altintas, “Scalable request routing with next-neighbor load sharing in multi-server environments,” in *Proc. IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Mar. 2005, vol. 1, pp. 441–446.

[27] R. A. Horn and C. R. Johnson, *Topics inMatrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1995.

[28] F. Cece, V. Formicola, F. Oliviero, and S. P. Romano, “An extended ns-2 for validation of load balancing algorithms in content delivery networks,” in *Proc. 3rd ICST SIMUTools*, Malaga, Spain, Mar. 2010, pp. 32:1–32:6.