ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

An Enhanced Load Equilibrium Technique in Content Delivery Network

K.Supriya[1], S.Venu Gopal[2]

Student, M. Tech CSE Dept, Vardhaman College of Engineering, Shamshabad, Hyderabad, Telangana, India[1]

Associate Professor, M. Tech CSE Dept, Vardhaman College of Engineering, Shamshabad, Hyderabad, Telangana, India[2]

Abstract: Content Delivery Network is a distribution system of servers on the Internet that accelerates practical solution of Web applications like audio, video and other Internet-based content to users throughout the globe. The aim of CDN is to deliver content from website to people more quickly and efficiently, based on their geographic location from the nearest edge server. In this paper we are going to study on different policies which give challenging issues for executing an effective new distributed algorithm for load balancing in content delivery network. Beginning from such categorization of fluid flow model which describes about network of servers queue stability. And finally conceive a new theorem which is reformulated in time-discrete form. And Specify in real time sequence, lastly, the complete method is validate by simulation process.

General Terms: Content Delivery Network, Load equilibrium.
Key Words: Reformulation, response time, continuous fluid model, control law balancing.

I. INTRODUCTION

A CDN is a large distribution network storing of web content where this content is dispersed with one or more dissemination servers called as surrogate server in order to perform reliable and efficient delivery of data to the end users. Content delivery network consist of unique server called back-end server, this back-end server consist of original information i.e. dynamic content (data that change with time) and multiple duplicate servers called surrogate servers these server are place over edge of the network to increase the content availability to the client and it consist of static content that is all the surrogate server have same type of data in them. (data that doesn't change with time) In some situations, there is another server called as redirector, which dynamically redirects client requests based on different policies. The main aim of CDN is to provide content to the end-user with high availability and high proficient.

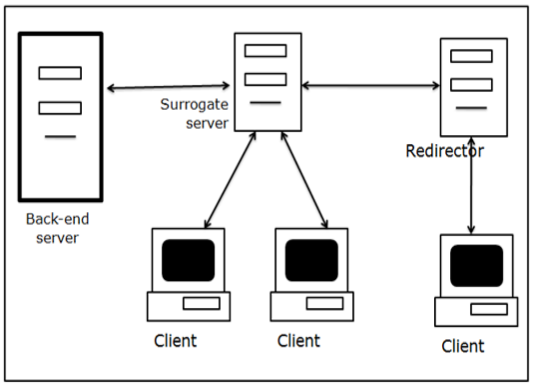

Fig 1: Architecture of content Delivery Network

The above figure shows the single network structure of content delivery network .It consist of one back-end server, surrogate server where the communication flows in two ways: one between client and surrogate server and another between surrogate server and original server. Redirector which redirects the client request to the appropriate server.

The main important performance enhancement of CDN is:
1) Over all system amount that is the average number of request served in unit time known as system throughput, is calculated according to the capability of available servers.
2) The response time proficient by increasing the capability of servers.
3) Client experience of response time after a request is issued. This will be achieved by the principle of closest and less loaded server which handles the client request. It requires a redirecting mechanism and request routing technique.

II. RELATED WORK

A CDN is a system of computers which consist of duplicate copies of data at various locations in order to maximize bandwidth for obtaining data quickly by clients throughout the network. A client can access duplicate content from nearby server. A CDN consist of different policies they are:

S.NO	Policy	Parameter	Result
1	Geographical distance	Less distance(nearby server)	Traffic congestion issues(Fails)
2	Routing Mechanisms-(DNS request Routing)(Transport request Routing)	IP address, packet tunneling Forwarding requests	Flash-crowds, effects quality of service(Fails)
3	Network management law	Discretization and non-linear solution	Load balancing can provided with local and global servers(Pass)
4	Surrogate placement	Reduce latency	provide high quality services and low CDN prices
5	Performance measurement	Bandwidth, latency, surrogate server utilization, reliability	predict, monitor and ensure content

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

III. BACKGROUND

Load Balancing is an important problem in CDN. The basis of this technology is the delivery at edge points of the network, in proximity to the request areas, to improve the user's perceived performance while limiting the costs.
In an existing queue-adjustment strategy, the scheduler is located after the queue and just before the server. The scheduler might assign the request pulled out from the queue to either the local server or a remote server depending on the status of the system queues.

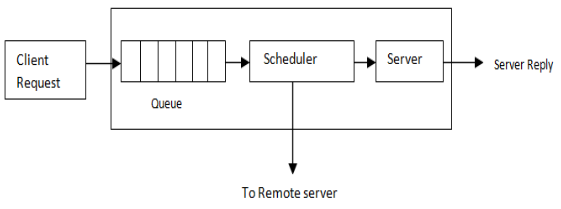

Fig (2): Queue-adjustment

In a rate-adjustment model, instead the scheduler is located just before the local queue. Upon arrival of a new request, the scheduler decides whether to assign it to the local queue or send it to a remote server.

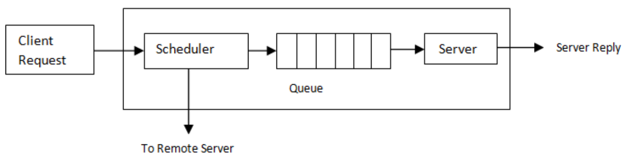

Fig (3): Rate-adjustment

In a hybrid-adjustment strategy for load balancing, the scheduler is allowed to control both the incoming request rate at a node and the local queue length. Thus in Existing systems, Upon arrival of a new request, indeed, a CDN server can either elaborate locally the request or redirect it to other servers according to a certain decision rule, which is based on the state information exchanged by the servers. Such an approach limits state exchanging overhead to just local servers.

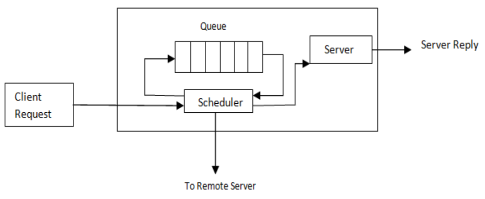

Fig (4): Hybrid-adjustment

The simplest static algorithm is the Random balancing mechanism (RAND). In such a policy, the incoming requests are distributed to the servers in the network with a uniform probability. Another well-known static solution is the Round Robin algorithm (RR). This algorithm selects a different server for each incoming request in a cyclic mode. Each server is loaded with the same number of requests without making any assumption on the state, neither of the network nor of the servers.

The Least-Loaded algorithm (LL) is a well-known dynamic strategy for load balancing. It assigns the incoming client request to the currently least loaded server. Such an approach is adopted in several commercial solutions. Unfortunately, it tends to rapidly saturate the least loaded server until a new message is propagated. Alternative solutions can rely on Response Time to select the server: The request is assigned to the server that shows the fastest response time.
The Two Random Choices algorithm (2RC) [11] randomly chooses two servers and assigns the request to the least loaded one between them. A modified version of such an algorithm is the Next-Neighbor Load Sharing [12]. Instead of selecting two random servers, this algorithm just randomly selects one server and assigns the request to either that server or its neighbor based on their respective loads (the least loaded server is chosen).
In these techniques upon arrival of new request, indeed, a CDN server can either elaborate locally the request or redirect it to other servers according to a certain decision rule, which is based on the state information exchanged by the servers. Such an approach limits state exchanging overhead to just local servers. The performance and quality of service are very poor.

IV. PROBLEM DEFINITION AND SOLUTION

Existing System

- ➢ In this existing system, we face the challenging problems of describing and implementing a control law for load balancing in Content Delivery Networks.
- ➢ An official study of a CDN system, accomplish 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, which is also reformulated in a time-discrete version

Drawbacks of Existing System

- ➢ The goal of CDN can be achieved in many different ways, not all of which provide local stability guarantees, as well as balancing of the server's queues.
- ➢ Indeed, it might happen that the overall condition is met, but one or more local server's queues overflow, thus bringing to packet losses and unavailability of the overloaded servers.

Problem Statement

- ➢ We focus exclusively on critical conditions where the global resources of the network are close to saturation.
- ➢ This is a realistic assumption since an unusual traffic condition characterized by a high volume of requests, i.e., a flash crowd, can always overfills the available system capacity.
- ➢ In such a situation, the servers are not all overloading which we have local instability conditions where the input rate is greater than the service rate.

New System Proposal

➢ We first design a suitable load-balancing law that assures equilibrium of the queues in a balanced CDN by using a fluid flow model for the network of servers.

➢

➢ We present a new mechanism for redirecting incoming client requests to the most appropriate server, thus balancing the overall system requests load.

➢ Our mechanism leverages local balancing in order to achieve global balancing. This is carried out through a periodic interaction among the system nodes.

Distributed Load-Balancing Algorithm

To overcome the above problems a suitable load-balancing law that assures equilibrium of the queues in a balanced CDN by using a fluid flow model for the network of servers is designed. Here a new mechanism for redirecting incoming client requests to the most appropriate server, thus balancing the overall system requests load. This mechanism leverages local balancing in order to achieve global balancing. This is carried out through a periodic interaction among the system nodes.

ALGORITHM: Setup Phase

1: Begin with all servers initializing its information table with neighboring node.
2: client request ai arrives at Qi queue occupancy by server node i
3: if server queue Qi > =Qmax; handle the request Else redirect to neighboring server node j (j=1…n) with Qj>Qmax.
4: update the information table with every time interval T If (load of I – peer_load of j) > 0 Load difference = load of I - peer_load of j;
5: if the client request served successfully acknowledge the same to the server.

C. Pseudo code:

```
// peer update process
prob_space[0]=0;
load_diff=0;
load_diff_sum=0;
for (j=1;j<=n;j++)
{
if(load_i-peer[j].load)
{
load_diff=load_ipeer[
j].load;build_prob_space(load_diff,prob_space);
}
update_prob_space(load_diff_sum,prob_space);
}
// balancing process
if(prob_space[ ]= =NULL)
serve_request();
else {
float x=rand();
intreq_sent=0;
int i=0;
while(prob_space[i]= =1 or req_sent= =1)
{
if(prob_space[i-1] <=x <prob_space[i])
{
send_to(peer[i-1].addr);
req_sent=1;
} i++ ; }
```

We want to derive a new distributed algorithm for request balancing that exploits the results are presented. First of all, we 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. For this reason, in the following of this section, we focus on the control law are described. 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 server's loads, as assessed by Lemma.

Algorithm Description

The newly implemented algorithm going to consists of two independent parts: a procedure that is in charge of updating the status of the neighbors‟ load, and a mechanism representing the core of the algorithm, which is in charge of distributing requests to a node‟s neighbours. The pseudocode of the algorithm is reported. Even though the communication protocol used for status information exchange is fundamental for the balancing process, in this existing system we will not focus on it. Indeed, for our simulation tests, we implemented a specific mechanism: We extended the HTTP protocol with a new message, called CDN, which is periodically exchanged among neighboring 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.

V. SIMULATION RESULTS

In this section, we will revise about some result of new proposed system for real time application. The proposed system is most reliable and effective network simulator so that they can well suited in real time implementation. The distributed algorithm is considered through a simulator by using fluid flow model. In this we try to provide an effective simulation tests and its results by using the network simulator 2- ns2. The simulation is carried out using some scenarios they are packet drop, throughput, and packet-delivery ratio.

Performance graph:

Performance is considered by the way a routing protocol deals with providing an optimal Quality of Service (QoS), packet delivery fraction, end-end delay.

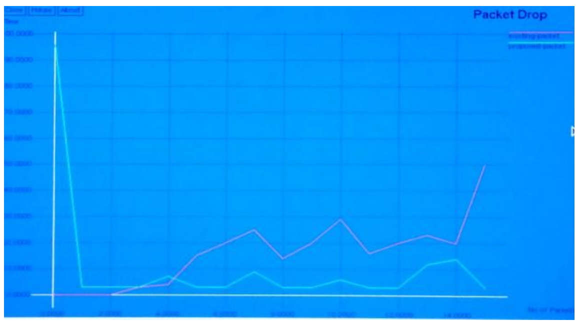

Fig (6): Packet drop

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 7, July 2015

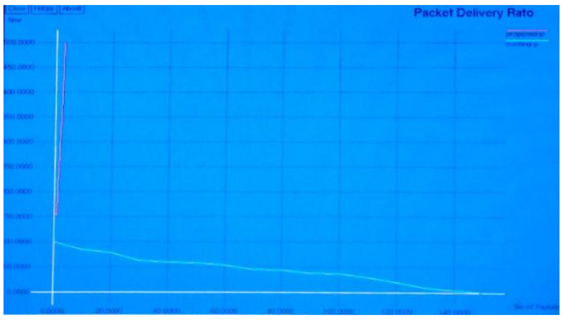

Fig (7): Packet-delivery ratio

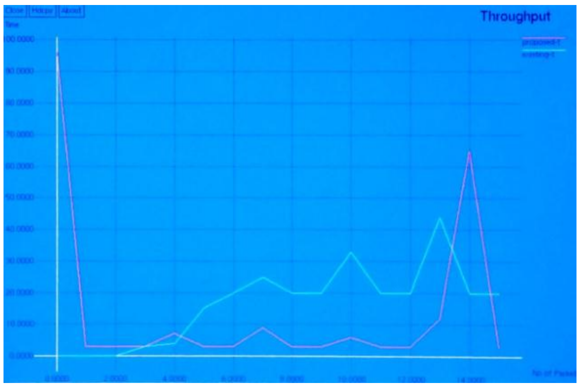

Fig (8): Throughput

VI. CONCLUSION

This project aims to achieve load balancing in the network by removing local queue instability condition through redistribution of potential excess traffic to the set of neighbors of the congested server globally and the algorithm is first introduced in its time-continuous formulation and then put in a discrete version specifically conceived for its actual implementation and deployment in an operational scenario. Through the help of simulations, we demonstrated both the scalability and the effectiveness of our proposal.

REFERENCES

[1]. G. Caro_glio, G. Morabito, L. Muscariello,"From Content Delivery Today to Information Centric Networking" January 8, 2013.
[2]. A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and S. Shenker. "Naming in content-oriented architectures" In In Proc. of ACM SIGCOMM { ICN 2011, August 2011.
[3]. Lefteris Mamatas, Tobias Harks, and Vassilis Tsaoussidis "Approaches to Congestion Control in Packet Networks" journal of internet engineering, vol.1, no.1, January 2007.
[4]. D. Chiu and R. Jain, "Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks", Journal of Computer Networks and ISDN, vol. 17, no. 1, June 1989, pp. 1–14.
[5]. D. Katabi, M. Handley, and C. Rohrs, "Congestion Control for High Bandwidth-Delay Product Networks", in Proc. ACM SIGCOMM 2002, Pittsburgh, USA, August 2002.
[6]. Zeng Zeng, and Bharadwaj Veeravalli, "Design and Performance Evaluation of Queue-and-Rate-Adjustment Dynamic Load Balancing Policies for Distributed Networks" IEEE TRANS. ON COMPUTERS, VOL. 55, NO. 11, NOVEMBER 2006.
[7]. T. Leighton, "Improving Performance on the Internet," Commun. ACM, vol. 52, no. 2, Feb. 2009, pp. 44–51.
[8]. V. Firoiu and M. Borden. A study of active queue management for congestion control. In Proceedings of Infocom 2000.
[9]. F. Dabek, R. Cox, F. Kaashoek, and R. Morris.Vivaldi: Distributed Content Delivery using Load-Aware Network Coordinates 2008 ACM 978-1-60558-266.
[10]. A.E. Kostin, I. Aybay, and G. Oz, "A Randomized Contention-Based Load-Balancing Protocol for a Distributed Multi server Queuing System," IEEE Trans. Parallel and Distributed Systems, vol. 11, no. 12, Dec. 2000.