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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 5, May 2014

Efficient and Distributed Control Mechanism for load Handling in Content Distributed Network

Sampath K V [1], Ragavendra T S [2]

Student- M.Tech, Department of Computer Networking Engineering, Cambridge Institute of Technology, India[1]

Asst. Professor, Dept of Computer Science &Engg., Cambridge Institute of Technology, Bangalore, India [2]

Abstract: A content distribution network (CDN) or content delivery network is a distributed and internetworked system of multiple-severs. Content distribution network provide web based services to clients throughout the globe. A content distribution network (CDN) technology is content redundancy or multiple copy of content that provides a fool-safe service. Content distribution network is much suitable for web application like streaming audio, video, and Internet television programming.A Content distribution network is an effective solution to the emerging Web applications. Unfortunately it also faces a higher risk of degradation in overall performance of entire distributed network when high number of request arrives from client flash crowd. In this research paper we propose an efficient control law for handle the load on individual servers by using efficient request routing mechanism which can handle worst case scenario in the existing system.

Keywords: Content distributed network (CDN), surrogate servers, request queue, flash crowd.

I. INTRODUCTION

A content distribution network [1, 3] provides an effectively solution to coordinate the web service provided by a scalable distributed multiple-servers. A content distribution network as shown in Fig (1), constitute of few number of backend servers [1,2] (original servers) which has original information, along with many distribution servers which are known as surrogate servers[1]. To increase availability of server content (web content) to the clients the same is replicated in the surrogate servers. So surrogate servers are redundancy of original servers to some extent. Redundant servers contain the static information (data don't change with time). Whereas the original server contains the dynamic information (data change with time).

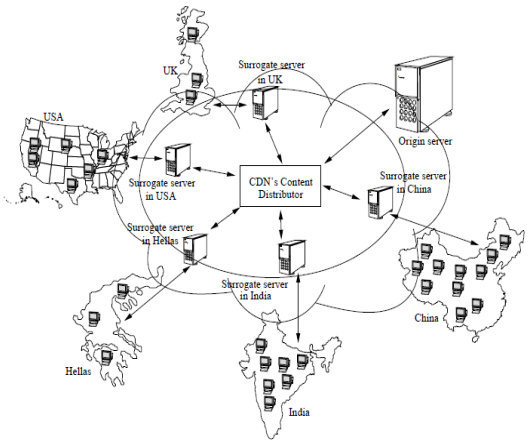


Fig (1)

A. The major performance improvements discussed in this paper are:

1. The average number of requests handled in a time unit by which overall system performance is increased.
2. The improvement is the basis of the request handling capabilities of the servers.
3. Client experience of Response time after a request is issued; this follows the principle of closest and less loaded server handle the client request. It requires a powerful redirecting and request routing mechanism.

II. RELATED WORK

A content distribution network (CDN) technology is content redundancy or multiple copy of content that provides a fool-safe service. Content distribution network is much suitable for web application like streaming audio, video, and Internet television programming. Content distribution network is much suitable for web application like streaming audio, video, and Internet television programming (IPTV).In CDN request routing is done by disseminate client requests to the redundant servers.

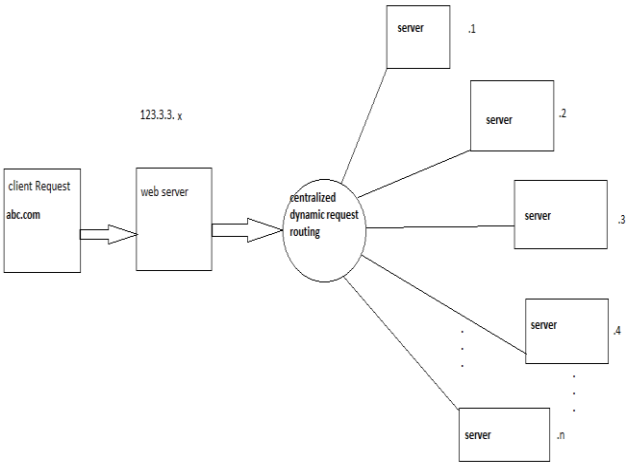
There are many centralized approaches currently used in existing system:

They can be generally classified as 1) hardware approach 2) static approach and 3) dynamic approach. In hardware approach the costly multiple switches are used to redirect.Request is served by server closest to client. Static approach chooses a server which does not forward to any information relate to system at decision time. Static approach chooses a server without considering any information about the status of the system at decision time. Static algorithms effectively increase the system availability and overall performance. However they are not capable of handling the abnormal situations like flash crowds.Dynamic approach as shown in Fig (2) give a valid alternative to static and hardware approach which make use of information of servers and network .Such approaches give a good result and make use of essential information returning either from the servers or network thereby increases the overall performance. The selection of the suitable server is finished through a collection and

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 5, May 2014

future analysis of the many parameters extracted from the network



Dynamic approach: Fig (2)

The static approach and dynamic approach we discussed earlier is a part of centralized approach ,that is a centralized system always monitor the redundant servers and it is the key component which over seen the request routing and redirection. In case of the centralized system failure then there is a chance of whole network failure or degradation in system performance.

III. PROPOSED SYSTEM

A. Proposed system Design*:*

- Initially content of original server is replicated into multiple-surrogate servers.
- Surrogate servers are initialized with request queues with request capacity of 5(req_queue<=5).
- Each severs in network have information about neighboring or adjacent servers
- Each severs maintain an information table.
- Considered at each time interval *T* the server updates its information table.
- Any Client requests are redirected to the closest server.

B. Proposed Algorithm Description :

The proposed system has two algorithms both are independent on each other. Though clients are involved in our proposed system network they have no significant role other than requesting for service to the closest server. All the surrogate 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 Q_{max}, and the $Qi_{(t)}$ be the queue occupied by server i at time consider the arrival rate be $a_{i(t)}$ service rate id $g_{i(t)}$

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

$$g(Qi(t)) = \sum_{j \epsilon Ne_i} a_{ij(t)} \quad (1)$$

for i = 1…n.

Ne_i= {adjacent j of node i}, and $a_{ij}(t)$ 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

$$\sum_{j \epsilon Ne_i} .a_{ij(t)} = \sum_{j \epsilon Ne^{+}_i} .a_{ij(t)} + \sum_{j \epsilon Ne^{-}_i} .a_{ij(t)} \quad (2)$$

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

ALGORITHM: Setup Phase

1: Begin with all servers initializing its information table with neighboring node.

2: client request a_i arrives at Q_i queue occupancy by server node i

3: if server queue $Q_i >= Q_{max}$; handle the request
Else redirect to neighboring server node j (j=1…n) with $Q_j > Qmax$.

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_i-
peer[j].load;build_prob_space(load_diff,prob_space);
}
update_prob_space(load_diff_sum,prob_space);
}
```

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 5, May 2014

```
// 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++ ; }
```

IV. Simulation Results

In this simulation section we discuss and try to bring some result so the proposed system can be brought for real time application. We simulated the proposed system in well known and most reliable network simulators so that they can well fitted in real time implementation.The efficiency of our algorithm is considered through a simulator also grouping the existing techniques like static and dynamic approach. In this we try to provide an effective simulation tests and its results by using the network simulator 2- *ns2.* There are many simulator tools dedicated for CDN but we consider the widely excepted. The simulation is carried out using some scenario topology

A. Balancing Performance

We 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. We need to model each server nodes as a Ma/Ma/1 queue with a rate of service of S_i request rate of a_i. 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 .3

As discussed earlier overhead due to redirections can increase impact on overall performance in distributed network. The analyze of the impact on the performance due to the redirection overhead is as shown in Fig 4

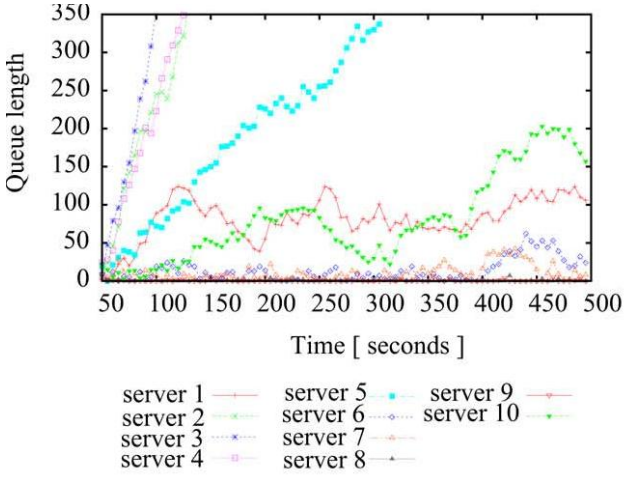


Fig (3)

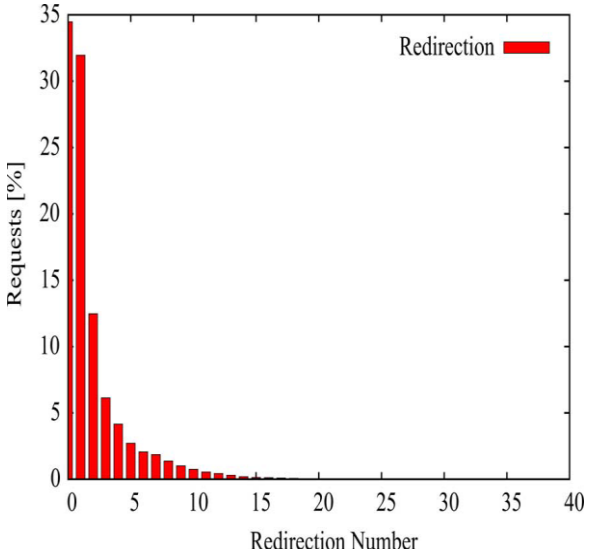


Fig (4)

Requests receiving a specified number of redirections

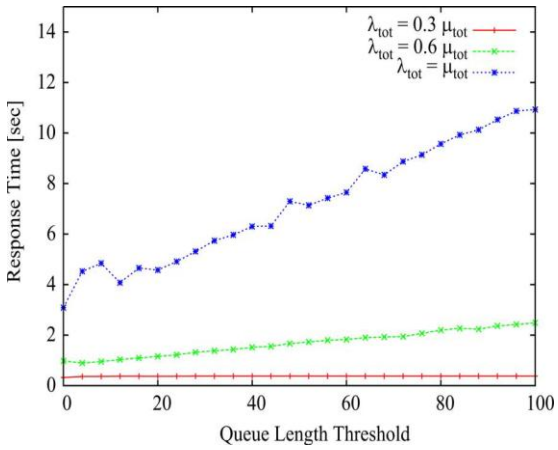


Fig (5)
Response time as a function of the lower queue threshold

IV. CONCLUSION

This paper discusses the importance of handling load using cooperative control law of surrogate servers. The simulation results show that the proposed system can successfully handle the all draw back that can be faced in existing system. All results and proof proved that the proposed system can out-perform the alternative system presently implemented. Finally we conclude with the result that the proposed system can be implemented in real time.

V. FUTURE WORK

Our future work is dedicated to implement the proposed system in real time. As we seen the proposed system have shown much reliable even in worst case scenario during the simulation experiments and all results proved so the proposed system can be implemented in real world.

ACKNOWLEDGMENT

The author would like to thank The Principal Dr. Suresh.L Cambridge Institute of Technology for giving him the opportunity to work on this project. He is also thankful to his HOD Dr. Satyanarayanreddy.K (Department of MCA, MTECH (CNE)) for his constant encouragement and moral support. Also he would like to thank Prof. Raghavendra.T.S for his support and encouragement, which helped me in correcting my mistakes and proceed to complete the paper with the required standards.

REFERENCES

[1] 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.M. Clerc, "The Swarm and the

[2] T. Leighton, "Improving Performance on the Internet", Commun. ACM, vol. 52, no. 2, pp.44-51, Feb. 2009.

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

[4] S. Manfredi, F. Oliviero, and S. P. Romano, "A distributed control law for load balancing in content delivery network ," in Proc. IEEE GLOBECOM Workshop, Miami, FL, Dec. 2010, pp. 579–583.

[5] K.R.Rao,Z.S.Bojkovic, D.A.Milovanovic,"Introduction to Multimedia Communications: Applications, Middleware, Networking", Wiley, New Jersey, USA, 2006.

[6] Akamai:"Akamai," 2013 [Online]. Available: http:// www.akamai.com/index.html.

[7] Networks: "Limelight-Networks," 2013 [Online]. Available:http://.uk.llnw.com.

[8] CDNetworks, :"CDNetworks," 2013 [Online].Available: http:// www.us.cdnetworks.com/index.php

[9] Coral: "The Coral Content Distribution Network," 2013 [Online]. Available: http://www.coralcdn.org.

BIOGRAPHIES

Sampath K V is an M.Tech student of CAMBRIDGE INSTITUTE OF TECHNOLOGY, Bangalore, India Presently he is pursuing his M.Tech [CNE] from this college and he received his B. E from Sri Dharmasthala Manjunatha institute of technology(sdmit-ujire), affiliated to VTU University, Dakshina Kannada in the year 2011. His area of interest includes Computer Networks, HCI, Algorithms, web 2.0 etc. Cloud computing and Object oriented Programming languages, all current trends and techniques in Computer Science.

Ragavendra T S is currently Assistant Professor in the Dept. of computer Science and Engg at Cambridge Institute of Technology, Bangalore. He has received M.Tech degrees affiliated to VTU. His Area of Expertises are Computer Architecture, Microprocessor, cloud computing and Wireless sensor networks.