International Journal of Computer Science
ISSN:2348-6600
Oddity...Probe...Reviste...
Published by SK Research Group of Companies(SKRGC)

http://www.ijcsjournal.com Volume 3, Issue 1, No 4, 2015. ISSN: 2348-6600
Reference ID: IJCS-087 PAGE NO: 499-503.

Performance Enhancment in Content Centric Network Using Load Balancing

K.V.VINODKUMAR[#1] , Mrs T.MANORANJITHAM[*2],
[#]*K.V.VINODKUMAR(M.Tech),Dept of Computer Science and Engineering.SRM University, CHENNAI*
[*]*Mrs T.MANORANJITHAM, Asst Prof, Dept of Computer Science and Engineering.SRM University, CHENNAI*

[1]kommi.vinodkumar@gmail.com
[2]manoranjitham.t@ktr.srmuniv.ac.in

Abstract— In Content Delivery Networks (CDNs) the challenging issue is defining and implementing an effective law for load balancing. In the proposed system an formal study of a CDN system is carried out through the exploitation of a fluid flow model characterization of the network of servers. It provides a lemma about the network queues equilibrium. The result is leveraged in order to devise a novel distributed and time-continuous algorithm for load balancing, which is also reformulated in a time-discrete version .The discrete formulation of the proposed balancing law is eventually discussed with the actual implementation in a real-world scenario. Finally, this is validated by the means of simulations.

Index Terms— **Content Delivery Network (CDN), control theory, request balancing.**

I. INTRODUCTION

A content delivery network (CDN) is a large distributed 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, documents) ,applications (e-commerce, portals), live streaming media, on-demand streaming media, and social networks.

Content providers such as media companies and e-commerce vendors pay CDN operators to deliver their content to their audience of end-users. In turn, a CDN pays ISPs, carriers, and network operators for hosting its servers in their data centers. Besides better performance and availability, CDNs also offload the traffic served directly from the content provider's origin infrastructure, resulting in possible cost savings for the content provider. In addition, CDNs provide the content provider a degree of protection from DoS attacks by using their large distributed server infrastructure to absorb the attack traffic. While most early CDNs served content using dedicated servers owned and operated by the CDN, there is a recent trend to use a hybrid model that uses P2P technology. In the hybrid model, content is served using both dedicated servers and other peer-user-owned computers as applicable. single server distribution and CDN Scheme of distribution shown in figure 1.

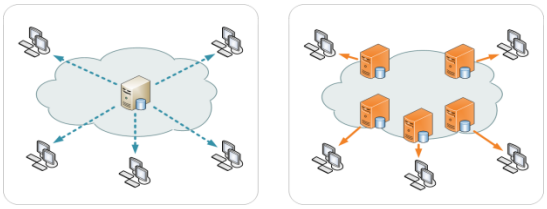

Fig 1: Single server distribution, CDN scheme of distribution

1.1 Operation

Most CDNs are operated as an application service provider (ASP) on the Internet (also known as on-demand software or software as a service). An increasing number of Internet network owners have built their own CDNs to improve on-net content delivery, reduce demand on their own telecommunications infrastructure, and to generate revenues from content customers. This might include offering access to media streaming to internet service subscribers. Some larger software companies such as Microsoft build their own CDNs in tandem with their own

International Journal of Computer Science
ISSN:2348-6600
Oddity...Probe...Reviste...
Published by SK Research Group of Companies(SKRGC)
http://www.ijcsjournal.com Volume 3, Issue 1, No 4, 2015. ISSN: 2348-6600
Reference ID: IJCS-087 PAGE NO: 499-503.

products. Examples include Microsoft Azure CDN and Amazon Cloud Front.

Here content (potentially multiple copies) may exist on several servers. When a user makes a request to a CDN hostname, DNS will resolve to an optimized server (based on location, availability, cost, and other metrics) and that server will handle the request.

1.2 Content Delivery Network Technology

CDN nodes are usually deployed in multiple locations, often over multiple backbones. Benefits include reducing bandwidth costs, improving page load times, or increasing global availability of content. The number of nodes and servers making up a CDN varies, depending on the architecture, some reaching thousands of nodes with tens of thousands of servers on many remote points of presence (PoPs). Others build a global network and have a small number of geographical PoPs.

Requests for content are typically algorithmically directed to nodes that are optimal in some way. When optimizing for performance, locations that are best for serving content to the user may be chosen. This may be measured by choosing locations that are the fewest hops, the least number of network seconds away from the requesting client, or the highest availability in terms of server performance (both current and historical), so as to optimize delivery across local networks. When optimizing for cost, locations that are least expensive may be chosen instead. In an optimal scenario, these two goals tend to align, as servers that are close to the end-user at the edge of the network may have an advantage in performance or cost.

Most CDN providers will provide their services over a varying, defined, set of PoPs, depending on the geographic coverage desired, such as United States, International or Global, Asia-Pacific, etc. These sets of PoPs can be called "edges" or "edge networks" as they would be the closest edge of CDN assets to the end user.

The CDN's Edge Network grows outward from the origin/s through further acquisitions (via purchase, peering, or exchange) of co-locations facilities, bandwidth, and servers.

1.3 Content networking techniques

The Internet was designed according to the end-to-end principle. This principle keeps the core network relatively simple and moves the intelligence as much as possible to the network end-points: the hosts and clients. As a result the core network is specialized, simplified, and optimized to only forward data packets.

Content Delivery Networks augment the end-to-end transport network by distributing on it a variety of intelligent applications employing techniques designed to optimize content delivery. The resulting tightly integrated overlay uses web caching, server-load balancing, request routing, and content services. These techniques are briefly described below.

Web caches store popular content on servers that have the greatest demand for the content requested. These shared network appliances reduce bandwidth requirements, reduce server load, and improve the client response times for content stored in the cache.

Server-load balancing uses one or more techniques including service-based (global load balancing) or hardware-based, i.e. layer 4–7 switches, also known as a web switch, content switch, or multilayer switch to share traffic among a number of servers or web caches. Here the switch is assigned a single virtual IP address. Traffic arriving at the switch is then directed to one of the real web servers attached to the switch. This has the advantage of balancing load, increasing total capacity, improving scalability, and providing increased reliability by redistributing the load of a failed web server and providing server health checks.

A content cluster or service node can be formed using a layer 4–7 switch to balance load across a number of servers or a number of web caches within the network.

Request routing directs client requests to the content source best able to serve the request. This may involve directing a client request to the service node that is closest to the client, or to the one with the most capacity. A variety of algorithms are used to route the request. These include Global Server Load Balancing, DNS-based request routing, Dynamic metafile generation, HTML rewriting, and any casting. Proximity choosing the closest service node is estimated using a variety of techniques including reactive probing, proactive probing, and connection monitoring.

CDNs use a variety of methods of content delivery including, but not limited to, manual asset copying, active web caches, and global hardware load balancers.

II. RECOMMENDED SYSTEM

Routing in a CDN is usually concerned with the issue of properly distributing client requests in order to achieve load balancing among the servers involved in the distribution

International Journal of Computer Science
ISSN:2348-6600
Oddity...Probe...Reviste...
Published by SK Research Group of Companies(SKRGC)
http://www.ijcsjournal.com Volume 3, Issue 1, No 4, 2015. ISSN: 2348-6600
Reference ID: IJCS-087 PAGE NO: 499-503.

network. Several mechanisms have been proposed in the literature. They can usually be classified as either static or dynamic, depending on the policy adopted for server selection.

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 selection 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.

The redirection mechanisms can be implemented either in a centralized or in a distributed way. In the former, a centralized element, usually called dispatcher, intercepts all the requests generated into a well-known domain, for example an autonomous system, and redirects them to the appropriate server into the network by means of either a static or a dynamic algorithm. Such an approach is usually adopted by commercial CDN solutions. With a distributed redirection mechanism, instead any server receiving a request can either serve it or redistribute it to another server based on an appropriate (static or dynamic) load-balancing solution.

Depending on how the scheduler interacts with the other components of the node, it is possible to classify the balancing algorithms in three fundamental models a queue-adjustment model, a rate-adjustment model, and a hybrid-adjustment model.

In a 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: If an unbalancing exists in the network with respect to the local server, it might assign part of the queued requests to the most unloaded
remote server. In this way, the algorithm tries to equally
balance the requests in the system queues. It is clear that in order to achieve an effective load balancing, the scheduler needs to periodically retrieve information about remote queue lengths.

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. Once a request is assigned to a local queue, no remote rescheduling is allowed. Such a strategy usually balances the request rate arriving at every node independently from the current state of the queue. No periodical information exchange, indeed, is requested.

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. Such an approach allows to have a more efficient load balancing in a very dynamic scenario, but at the same time it requires a more complex algorithm. In the context of a hybrid-adjustment mechanism, the queue-adjustment and the rate-adjustment might be considered respectively as a fine-grained and a coarse-grained process. Both centralized and distributed solutions present pros and cons depending on the considered scenario and the specific performance parameters evaluated. As stated in, although in some cases the centralized solution achieves lower response time, a fully distributed mechanism is much more scalable. It is also robust in case of dispatcher fault, as well as easier to implement. Finally, it imposes much lower computational and communication overhead.

In the following, we will describe the most common algorithms used for load balancing in a CDN. Such algorithms will be considered as benchmarks for the evaluation of the solution we propose in this paper.

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.

he 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.

International Journal of Computer Science
ISSN:2348-6600
Oddity...Probe...Reviste...
Published by SK Research Group of Companies(SKRGC)
http://www.ijcsjournal.com Volume 3, Issue 1, No 4, 2015. ISSN: 2348-6600
Reference ID: IJCS-087 PAGE NO: 499-503.

The Two Random Choices algorithm(2RC) 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. 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 Section III, we will present an alternative solution for load balancing, falling in the class of rate-adjustment approaches. We propose a highly dynamic distributed strategy based on the periodical exchange of information about the status of the nodes in terms of load. By exploiting the multiple redirection mechanism offered by HTTP, our algorithm tries to achieve a global balancing through a local request redistribution process.

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.

III. Load-Balanced CDN: Model Formulation

In this section, we will 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. We assume a fluid model approximation for the dynamic behavior of each queue. We extend this model also to the overall CDN system. Such approximation of a stochastic system.

Actually, this approximation cannot be exploited in a real scenario: The requests arrive and leave the server at discrete times ,hence in a given time interval, a discrete number of re- quests arrives at and departs from each server in the system 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, as assessed by Lemma.

IV. Proposed Distributed Load Balancing Algorithm

The implemented algorithm consists of two independent parts: a procedure that is in charge of updating the status of the neighbor's load, and a mechanism representing the core of the algorithm, which is in charge of distributing requests to a node's neighbors' based on servers. In the pseudo code of the algorithm is reported. Even though the communication protocol used for status in-formation exchange is fundamental for the balancing process, in this paper 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 place, in which are nonetheless out of the scope of the present work. Every second, the server sends its status information to its neighbors and, at the same time, waits for their information. After a well-defined interval, the server launches the status up- date process. We suppose all the information about peers' load is already available during such a process.

Algorithm:

```
//peer status update
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);
             load_diff_sum=load_diff_sum+load_diff; }
  update_prob_space(load_diff_sum,prob_space);
}

//balancing process
if(prob_space[]==null)
          server_request();
else{
     float x=rand();
     int req_sent=0;int i=0;
     while(prob_space[i]==1 or req_sent==1){
          if(prob_space[i-1]<=x<prob<prob_space[i]){
                    send_to(peer[i-1].addr);
                    req_sent=1;
          }
           i++;
     }
}
```


Published by SK Research Group of Companies (SKRGC).

International Journal of Computer Science
ISSN:2348-6600
Oddity...Probe...Reviste...
Published by SK Research Group of Companies(SKRGC)

http://www.ijcsjournal.com Volume 3, Issue 1, No 4, 2015. ISSN: 2348-6600
Reference ID: IJCS-087 PAGE NO: 499-503.

V.CONCLUSION

In this paper, we presented a novel load-balancing law for cooperative CDN networks. We first defined a model of such networks based on a fluid flow characterization. We hence moved to the definition of an algorithm that aims at achieving load balancing in the network by removing local queue instability conditions through redistribution of potential excess traffic to the set of neighbors of the congested server. 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, which outperforms most of the potential alternatives that have been proposed in the past. The present work represents for us a first step toward the realization of a complete solution for load balancing in a cooperative, Distributed environment.

VI.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] M. Colajanni, P. S. Yu, and D. M. Dias, "Analysis of task assignment policies in scalable distributed Web-server systems," IEEE Trans. Parallel Distrib. Syst., vol. 9, no. 6, pp. 585–600, Jun. 1998.

[6] D. M. Dias, W. Kish, R. Mukherjee, and R. Tewari, "A scalable and highly availableWeb server," in Proc. IEEE Comput. Conf., Feb. 1996, pp. 85–92.68 IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 21, NO. 1, FEBRUARY 2013.

[7] C. V. Hollot, V. Misra,D. Towsley, andW.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.

[8] 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.

[9] 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.