International Journal of Latest Trends in Engineering and Technology
Vol.(7)Issue(3), pp. 165-170
DOI: http://dx.doi.org/10.21172/1.73.523
e-ISSN:2278-621X

Monitoring Server Based Server Selection Strategy in Content Delivery Networks

Manish Kumar Pal[1] And M A Rizvi[2]

Abstract: Content Delivery Network (CDN) services have to deal with many problems one of which one is content not available at the associated cache servers. In this work it is tried to find a optimized and reliable solution for this problem. The proposed solution is to request the content from other optimum cache server. In this paper we are suggesting to have a monitoring server to maintain availability status of the content and health parameters of all the cache server. In case of the content availability mismatch, with the help of monitoring information monitoring server selects the cache server, which will be utilized to serve the content to the client. This will increase efficiency since users are not getting the content from the origin server instead we are getting it from the one of the cache server which is closest and efficient and reliable.

1. INTRODUCTION

A content delivery network or content distribution network (CDN) is a distributed system placed at various locations across the globe across the internet. A large part of the content on the internet is served by CDN, including web object eg. Text graphics scripts, downloadable objects eg. Media files software's, applications (commercial portals), social networks & streaming media content.

Internet is a network of networks, when there is request of the content for content from the server located at other side of the globe the packets have to travel through many backbone networks. CDN multiply the use of transport networks by optimizing content delivery. This optimization is done by placing replica cache servers across globe and serving the content to the client by a cache serer i.e. closest or most appropriate to the client.

CDN involves placing of content replica servers known as edge servers at geographically tactical locations so as to serve the content to the requesting client quickly without delay. This tactically closeness of the server to the client is chosen based on parameters so that the data packets have to travel less & also able to avoid any clogged network. The decision making also involves what content can be replicated & what cannot be replicated to the edge servers. The CDN is used to serve a large number of users with reduction in download time & network traffic. A CDN system consists of content-delivery, request-routing, replication and accounting infrastructure. The content-delivery infrastructure consists of edge servers (also called surrogates or cache servers) that deliver content to end-users. The request-routing infrastructure is responsible for routing client request to appropriate cache servers. The replication infrastructure keeps the content stored in the CDN caches updated. In practice, CDNs typically host static content including images, video, media clips, advertisements, and other embedded objects for dynamic Web content.

Content delivery networks are so much prominent in today's internet that CDNs deliver majority of the today internet traffic. Massively distributed server infrastructures are deployed to replicate content and make it accessible from different Internet locations. For example, Akamai has deployed the most pervasive, highly distributed content delivery network (CDN) with more than 200,000 servers in over 110 countries and within more than 1,400 networks around the world. Google operates tens of data-centers and server clusters worldwide, and other companies such as Microsoft, Amazon, and Limelight follow similar approaches with highly distributed infrastructures. This scenario is not expected to change significantly in the next years. For instance, Google infrastructures presented a sevenfold increasing over just one year , while Cisco forecast that 51% of all Internet traffic will be served by CDNs by 2017. The vast growth of traffic and infrastructure illustrates that CDNs serve as a key part of today's Internet, and undertake heavy content delivery load. This promising growth makes CDNs a hot spot for research.

After the network and content is ready the request of the clients will begin to reach the content provider single server which in response serves the content to the client. But when the number of request increases globally the whole network will be filled by the content packets especially when the data is media file. To reduce the load on the content provider and also on the

[1] Department of CEA, NITTTR Bhopal
[2] Department of CEA, NITTTR Bhopal

network one of the solution is to move the load of the server on to the edge server which will replicate the content and serve the local client with the content. The server which contains the replicated content is also called surrogate server.

The selection of the edge server among the group plays a vital role in the performance of the CDN because a error in choosing beats the very purpose of CDN. The server selection mechanism is a field of research in which many researchers have contributed suggesting different parameters to be dominating the server selection.

To summarize CDN operation can be broadly divided into three categories Cache deployment, Request routing and Content replication. Our work in this paper works on the Request routing algorithm.

2. LITERATURE SURVEY:

In [1] Authors conducted a detailed study of the YouTube CDN with a view to understanding the mechanisms and policies used to determine which data centers users download video from. They have used week-long datasets simultaneously collected from the edge of five networks – two university campuses and three ISP networks - located in three different countries. They have employed state-of-the-art delay-based geo-location techniques to find the geographical location of YouTube servers. A unique aspect of their work is that they performed analysis on groups of related YouTube flows. This enabled to infer key aspects of the system design that would be difficult to glean by considering individual flows in isolation. The results reveal that while the RTT between users and data centers plays a role in the video server selection process, a variety of other factors may influence this selection including load-balancing, diurnal effects, variations across DNS servers within a network, limited availability of rarely accessed video, and the need to alleviate hot-spots that may arise due to popular video content.

In [2] Authors have proved that location-aware video server selection algorithm can direct a client to a less optimal content server, although there can be other better performing video delivery servers. In order to solve this problem, they propose to use dynamic network information such as packet loss rates and Round Trip Time (RTT) between an last node of a wireless network (e.g., In WiFi its an Internet Service Provider (ISP) router and in 3G network its a Radio Network Controller (RNC)) and video content servers, to find the best video content server when a video is requested. The study shows that the proposed architecture provides higher TCP performance, which leads to better viewing quality compared to location-based video server selection algorithm.

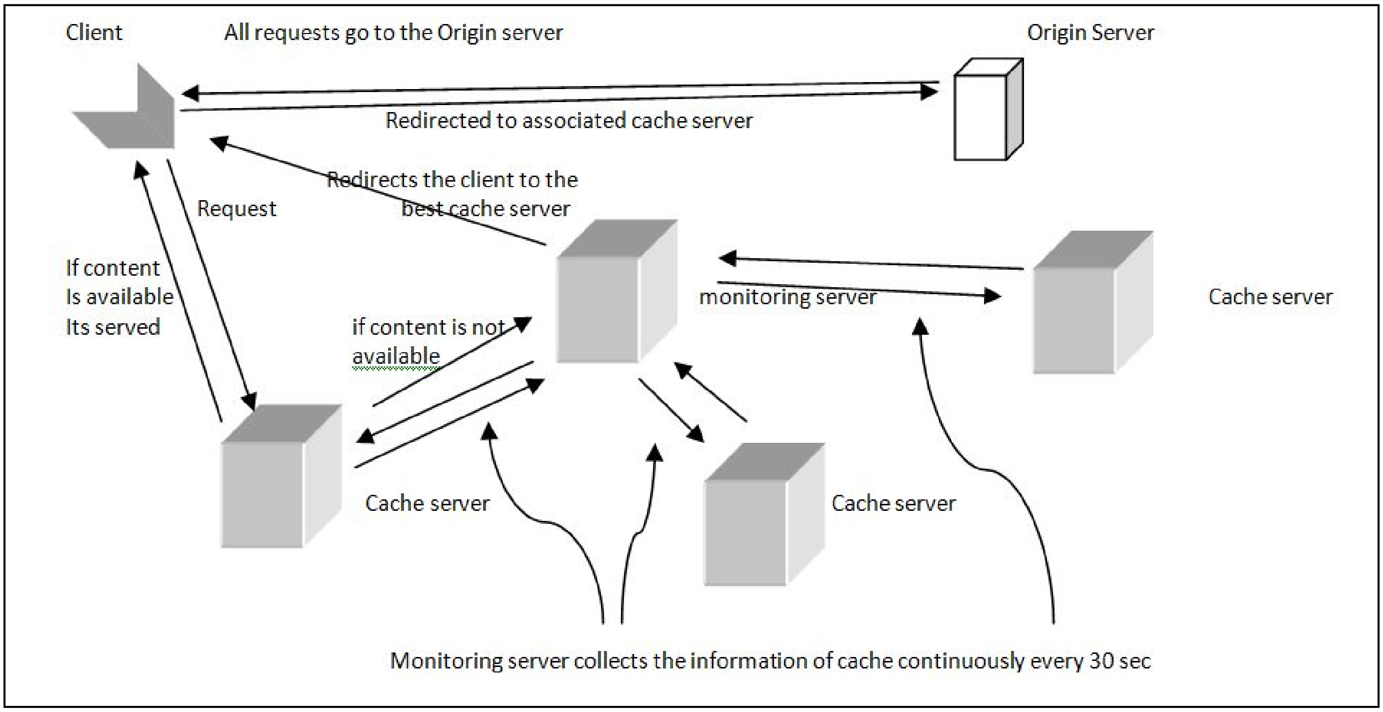
Client
All requests go to the Origin server
Origin Server
Redirected to associated cache server
Redirects the client to the best cache server
Request
If content Is available Its served
monitoring server
Cache server
if content is not available
Cache server
Cache server
Monitoring server collects the information of cache continuously every 30 sec

Fig. 1- Block diagram of CDN showing monitoring server

In [3] Authors have created local and global server and connected the global server to system manager, which is worked over Content Delivery Network to deliver and direct the user request to the nearest global edge server except local server and establish the connection between them and transfer the respective content. For optimization of edge selection process and reducing the load over content delivery network they approach local server concept in this paper. They are using Dijsktras algorithm to find the shortest distance between the edge servers and the client.

In [4] Authors propose a QoE-based server selection algorithm in the context of a CDN architecture. Using realistic characteristics of the server selection process, they have formalized their selection model as a sequential decision problem solved by the multi-armed bandit (MAB) paradigm. By using realistic experiments, they have demonstrated that their approach yields significant improvements in term of user perception instead of traditional methods (such as Fastest, Closest and Round Robin).

In [5] Authors have categorized and analyzed the working of CDN. Also they have described the organizational structure, content distribution mechanism, request redirection strategies and performance measurement mechanism. They have also surveyed and applied taxonomy to various CDNs.

In [6] Authors have suggested a methodology to copy the each content on the cache servers based on the cost of the content in turn saving the cost required for saving data. When the client request arrives and the data is not found on the associated cache server look for data in the other cache servers. Out of the other cache servers one which is having the lowest hop count ie lowest cost is selected to serve the content to the client.

In [7] Authors have suggested client side processing to choose the server among the resolved CDN servers based on load balancing. They are using mechanism called the DNS-proxy. They have done a measurement study on five major CDNs providers and compared the results with reduced web-page loading time.

In [8] Authors have investigated jointly controlling cache-server and delivery-route selection simultaneously. In the paper they have suggested using a optimization method which can be repeated continuously to maintain the desirable state for long periods. They are using the model predictive control (MPC), which has been widely used in system control. They are simultaneously optimizing both the cache-server and delivery-route selection using the MPC.

3. PROBLEM DEFINITION AND SCOPE

Problem Definition:

This work takes into consideration the research issues in relation with development of advanced, high performance and cost efficient content delivery approaches by analyzing the different strategies for server selection used to improve the working of the CDN networks.

Scope:

- Our scope of work is on request-routing system whose responsibility is to route client requests to an appropriate surrogate server which will deliver the content.
- There are a collection of network elements which support request-routing for a single CDN. They route the client requests to a replica server which is ‘closest’ to the client. However, choosing the replica server which is closest server may not be the best for servicing the client request.
- A request-routing system uses a set of metrics such as network closeness, clients perceived latency, distance, and also load on replica server in trying to direct users to the closest surrogate which can best serve the request. The selection of content and delivery techniques (for eg. full-site and partial-site) utilized by a CDN have a direct impact on the design of the request-routing system.
- The request-routing system in a CDN has two parts: one is deployment of a request-routing algorithm and second is use of a request-routing mechanism. On receiving a client request, request-routing algorithm is invoked. It specifies how edge server is selected in response to the given client request. On the other hand, a request-routing mechanism deals with informing the client about the selection. Mechanism is to first invoke a request-routing algorithm and then the result of the selection it obtains is informed to the client. It is proposes a change in server selection process in one case when the content is not found on the associated cache server.

4. OBJECTIVES:

- Study of existing server selection algorithms for CDN.
- Propose a new mechanisms for Server Selection for CDN.
- Modify the identified Server Selection Algorithms based on the new mechanism.
- Implement the existing and modified algorithm, and compare the results.

System Design

In this paper it is tried to work on request routing category of CDN operation. In the request routing as the name suggests the request are routed to the best of the cache servers. In the Operation of CDN cache nodes having replica of the content of the Original server. Each of these cache servers is associated to the demographic location of the

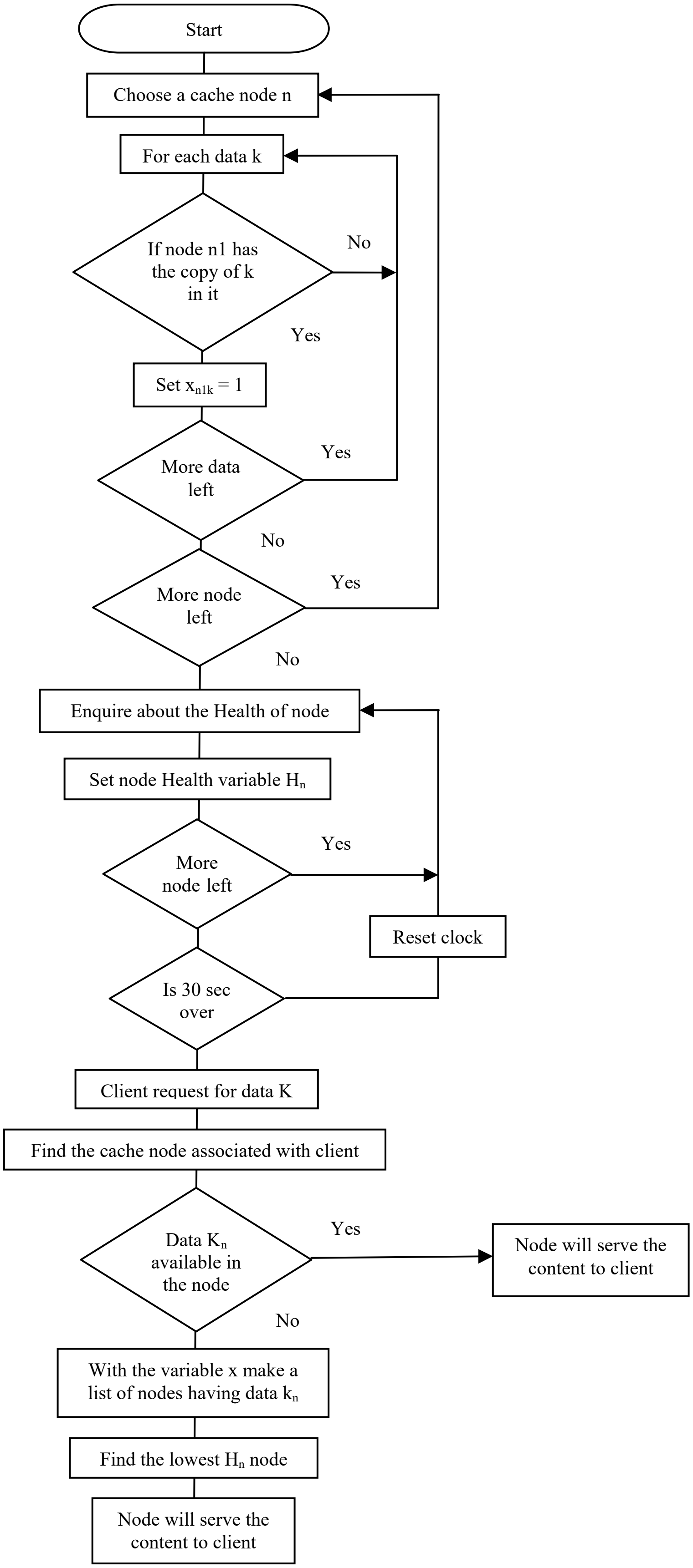
Start
Choose a cache node n
For each data k
If node n1 has the copy of k in it
No
Yes
Set $x_{n1k} = 1$
More data left
Yes
No
More node left
Yes
No
Enquire about the Health of node
Set node Health variable H_n
More node left
Yes
Reset clock
Is 30 sec over
Client request for data K
Find the cache node associated with client
Data K_n available in the node
Yes
Node will serve the content to client
No
With the variable x make a list of nodes having data k_n
Find the lowest H_n node
Node will serve the content to client

Fig. 2 Flowchart of selection of optimized server

clients. When a request is made by a client to the original server, the request is sent to the demographically associated cache server. Assuming client has requested for data k from the server. On the search of data k on the associated cache server if the data k is found, the data is served to the client by the associated cache server. If the data not found then in the original algorithm the data is requested from original server, but in a modified approach the data is checked for availability on other cache server [6].

In our methodology we are proposing a novel approach (refer Fig. 1 and 2) for handling of this situation when the requested data k is not found on the associated cache server. We are proposing to introduce an independent server called monitoring server. The monitoring server keeps on monitoring the data & health status of all the cache servers on regular interval (we are keeping this interval to be 30 sec). This monitoring needs to be done as regularly as possible so that monitoring server has the exact status of all the cache servers in the CDN network. This information will be the basis for the decision making in choosing the new cache server for serving the content k to the client. The more correct the data & health status is, more correct the decision will be. Monitoring server has the exact information on availability of data on all the cache servers. Therefore it has list of cache servers which have data k, it has to choose the cache server which has the data k and also has the best health status to accept the request from client & then serve the content. This selection of the best server chooses the fastest & the reliable cache server.
The Pseudo code for the algorithm is provided in the following

Algorithm

```
for each cache server n
    for each data k
        if n has a copy of k
            $x_{nk}$ = 1
    end for
end for
for every 30 sec
    for each cache server n
        set Health variable $H_n$
    end for
end for
A client requests for data k
if $x_{nk}$ = 1
    Cache node n serves client with data k
else
    list cache nodes with $x_{\_k}$ =1
    Choose node with lowest $H_n$ value
    Cache node n serves client with data k
end if
```

The data we are assuming with a variable k & node with a
variable n. The set of variables for data status for nodes are x_{nk} where n subscript identifies node & k subscript identifies data. The set of variables for health status for nodes are H_n where n subscript identifies node.

5. CONCLUSIONS AND FUTURE SCOPE

From the study of the papers mentioned above we can understand how much important server selection strategies are for the efficient working of the CDN. Any improvement in the server selection methodology is going to improve the performance of the CDN system. Keeping this in mind in this paper we have proposed modification on the server selection process to optimize the CDN overall performance. Our work is concentrated on request routing because after all the optimizing procedures have run the main task of actual data transfer which is large is size starts. Any optimization to this large chuck will simply affect the improvement of the performance. This sharing the status information & optimizing on that basis is not much of the overhead on the system than the benefits we are getting by having the content (which is large in size) served by efficient & reliable cache server than by the origin server

As a future field of work the performance evaluation can also to be done between combining the functionality of original server and monitoring server together in one and having a separate independent monitoring server. Also in the future work, the effect of copying the data k in the associated cache server for future requests can be evaluated.

REFERENCES

[1] Torres, Ruben, et al. "Dissecting video server selection strategies in the youtube cdn." *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*. IEEE, 2011.

[2] Nam, Hyunwoo, et al. "Towards dynamic network condition-aware video server selection algorithms over wireless networks." *Computers and Communication (ISCC), 2014 IEEE Symposium on*. IEEE, 2014.

[3] Sarddar, Debabrata, and Enakshmi Nandi. "Optimization of Edge Server Selection Technique using Local Server and System Manager in content delivery network." *International Journal of Grid and Distributed Computing* 8.4 (2015): 83-90.

[4] Tran, Hai Anh, et al. "QoE-based server selection for content distribution networks." *Computers, IEEE Transactions on* 63.11 (2014): 2803-2815.

[5] Pathan, Al-Mukaddim Khan, and Rajkumar Buyya. "A taxonomy and survey of content delivery networks." *Grid Computing and Distributed Systems Laboratory, University of Melbourne, Technical Report* (2007): 4.

[6] Lim, Kyongchun, et al. "Joint optimization of cache server deployment and request routing with cooperative content replication." *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014.

[7] Goel, Utkarsh, Mike P. Wittie, and Moritz Steiner. "Faster Web through Client-assisted CDN Server Selection." *2015 24th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2015.

[8] Kamiyama, Noriaki, et al. "Optimizing Cache Location and Route on CDN Using Model Predictive Control." *Teletraffic Congress (ITC 27), 2015 27th International*. IEEE, 2015.