A FRAMEWORK FOR MINIMIZING LATENCY IN CDN USING URL REQUEST ROUTING APPROACH

S.Manikandan[1], A.Chitra[2] , P.Venketesh[3]

[1] Senior Lecturer, [2]Professor, [3]Lecturer

Dept of Computer Science and Engg, PSG College of Technology,

Coimbatore -641004, India.

manigandan_me@yahoo.co.in

Abstract

Increase of web data in recent years is a crisis for distributing and managing the content of a websites. Content Distribution Network (CDN) serves to provide some special solution to these issues. Content Distribution Network maintains large number of replicas to act on behalf of origin servers. The main issues in designing Content Distribution Network are Request routing, Object replication, replication consistency and Server Distribution. Request routing is a familiar technique to progress the accessibility of web sites. It normally minimizes the client latency and increases content availability. This paper presents an exploration on different Request routing approaches and proposes a framework to minimize client latency using URL based request routing method.

Keywords: Content Distribution Network, Request Routing, Object replication, Replication consistency, Latency time, URL Request Routing.

Introduction

A proxy server acts as an intermediary in the Web page loading process, accepting a request from a browser, implementing authentication and filtering policies, and managing the request through, to the Web server. It can also give reports of user's activity and also smart caching system i.e. to read documents "unplugged" to the Internet. Proxy servers provide tighter control at network boundaries, because proxy is only connected to the Internet. A proxy server has proven to be an effective solution for controlling network access. As the demand for Internet content has increased it leads to several problems that are discussed later.

CDNs burst onto the scene in 1999 to address the fact that the Internet was not designed to handle large transmissions of Web content over long distances. Network congestion and traffic bottlenecks, worsen by growing payloads of Web traffic and degrade both individual Web site and network performance [1]. CDNs address the problem by storing and serving the content from many distributed surrogate servers that are geographically apart rather than from a centralized origin server. Using caching

technology, CDNs store replicas of content near users, rather than repeatedly transmitting identical versions of the content from an origin server thus improving the response time. As a result CDN accelerates and improves the quality of content delivered to end users, while lowering network congestion and bandwidth costs for ISPs.

CDNs replicate and deliver only content that the owners specify from surrogate servers throughout the Internet. The main issues in designing Content Distribution Network are Request routing, Object replication, replication consistency and Server Distribution.

Request routing methods select the appropriate surrogate server for the required clients request and redirect. The selection is based on information about surrogates load and on network metrics collected by various ways such as routing protocol information, RTTs (round trip times) measured by network probes etc. CDNs employ routing intelligence to guide user requests to local servers. Object replication can be used to improve availability in the face of network or server failure, to create numbers of concurrent accesses, and to allow users to access to close copies of objects, thus limiting the effects of network congestion. The replicated objects are to check with its consistency pattern. The consistency pattern is decided with the consideration of consistency models and policies designed by individual companies. Yet another challenge in CDN is to position the surrogate server in the right location. The locating problem includes surrogate server placement and surrogate content placement.

The Building blocks for a single CDN consist of several surrogate servers, a distribution system, a request routing system and an accounting system (used for billing). This paper presents a report based on the investigation on different request routing methods and proposed URL request routing method with its performance with DNS based request routing system.

The rest of this paper is organized as follows. Section 2 describes various request routing methods that are deployed in CDNs. The proposed URL request routing method with its equations are mentioned in Section 3. Section 4 presents the comparison of different methods. Section 5 concludes the paper with final remarks.

Request Routing

A typical CDN contains active components like Request Routing System (RRS), Distribution System (DS) and Accounting System (AS) and Surrogate server as shown in the fig1. A request-routing system [2][3] facilitates the activity of directing a client request to a suitable surrogate server. It consists of a set of network elements called Request routers. A Distribution system consists of a collection of network elements called content-distributor. It supports the activity of moving a publisher's content from the origin server to one or more Surrogate servers (using either Push or Pull algorithm).

The Accounting system supports the measurement and recording of content distribution and delivery activities. Information recorded by the accounting system is used as a basis for the transfer of money, goods, and obligation among the network service providers and the content providers.

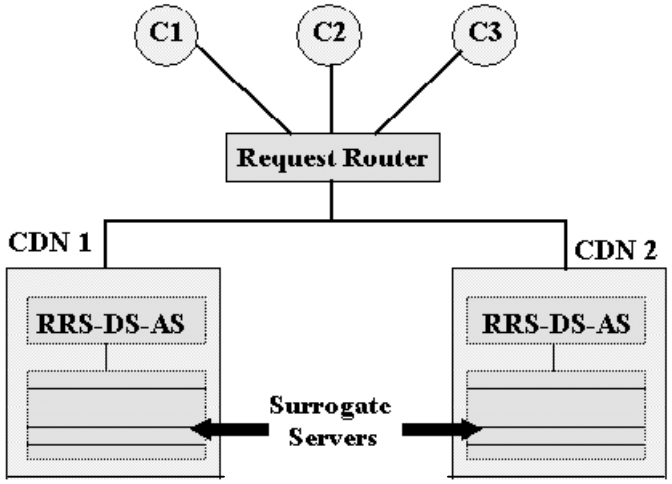

Fig1: CDN Architecture.
C1, C2, C3-Clients**, DS**-Distribution System,
RRS-Request Routing System, **AS**- Accounting System

Servers in a CDN are located at different locations in the Internet called surrogate servers. Client typically access content from surrogate servers by first contacting a request router. The request router makes a server selection decision and returns a server address to the client. The client then retrieves the content from the specified surrogate servers. A primary issue for a request router is how to direct client requests for an object served by the surrogate servers within the network. From the database, Request routers

choose the best server using static and dynamic information of various surrogate servers. Surrogate servers pass the information such as metrics of the server, network conditions and client proximity to the Request router. This section describes different techniques like DNS based, Transport layer based, Application layer based and content based request routing used in CDN.

DNS based Request Routing

Domain Name Server (DNS) based request routing is extensively deployed in the Internet at present. DNS based request routing [4] is also deployed as a directory service in CDNs for resolving the client request to appropriate surrogate server address. Specialized DNS server included in DNS system does the DNS resolution process [5-7]. DNS server is capable of resolving single or multiple surrogate address to handle the domain name of the desired website or content.

The client request for a web site or content in the Internet, subsequently the request moves towards the nearest DNS. The DNS resolves the request and returns the apt surrogate server address that can be either single or multiple replies. A single reply CDN server is authoritative for the entire DNS domain or sub domain. Client gets the surrogate address as a reply and contacts the surrogate server for website or content. Then the content is transferred from the surrogate servers to the client. If multiple surrogate server address is reply to the client, Client site DNS server decides to which surrogate server it should select from the reply. RFC 2782 (DNS SRV) provides guidance on the use of DNS for load balancing [8]. The aforesaid methods are single-level DNS server resolution system. Instead of single-level DNS resolution system multi-level DNS server resolution is in practice. This is to fetch the IP address from the next level of DNS server. A hierarchical architecture is deployed for multi-level DNS server resolution approach and in this, the most common mechanism used to insert multiple requests routing DNS server, in a single DNS resolution is by employing Name Server (NS) and Canonical NAME (CNAME).

In NS redirection, records are redirecting the authority to next hierarchical level. Here, to implement NS mechanism, Multiple DNS server is implicated in the name resolution. For example, a client site DNS server resolving sample.test.edu would

eventually request a resolution of sample.test.edu from the name server, authoritative for test.edu. The name server authoritative for this domain might be a Request-Routing NS server. In this case the Request-Routing DNS server can either return a set of A records or can redirect the resolution of the request sample.test.edu to the DNS server that is authoritative for example.com using NS records.

In CNAME redirection [4], the Request-Routing DNS server returns a CNAME record to direct resolution to an exclusively new domain. In principle, the new domain might use a new set of Request-Routing DNS servers. Multiple physical DNS servers that combine request routing and metric measurement can share an anycast IP address [9]. The packet containing the DNS resolution request will reach one of these DNS servers, which is the closest to the client site DNS server. After receiving the packet, the DNS server knows that it is the closest and can use this information in making routing decision.

Transport Layer Request-Routing

In Transport layer request routing technique closely inputs the first packet of the client's request to select the appropriate surrogate server for high-level granularity. As shown in the fig2, the first packet of the client request contains data about services, client, IP address, and port information and layer four protocols used in client side. Using these information's and integrating with user-defined policies, surrogate servers are selected.

In practice, the transport layer request routing using DNS server, which chooses forward flow traffic. But reverse flow traffic has larger data than forward flow traffic, so it takes a direct route instead via DNS.

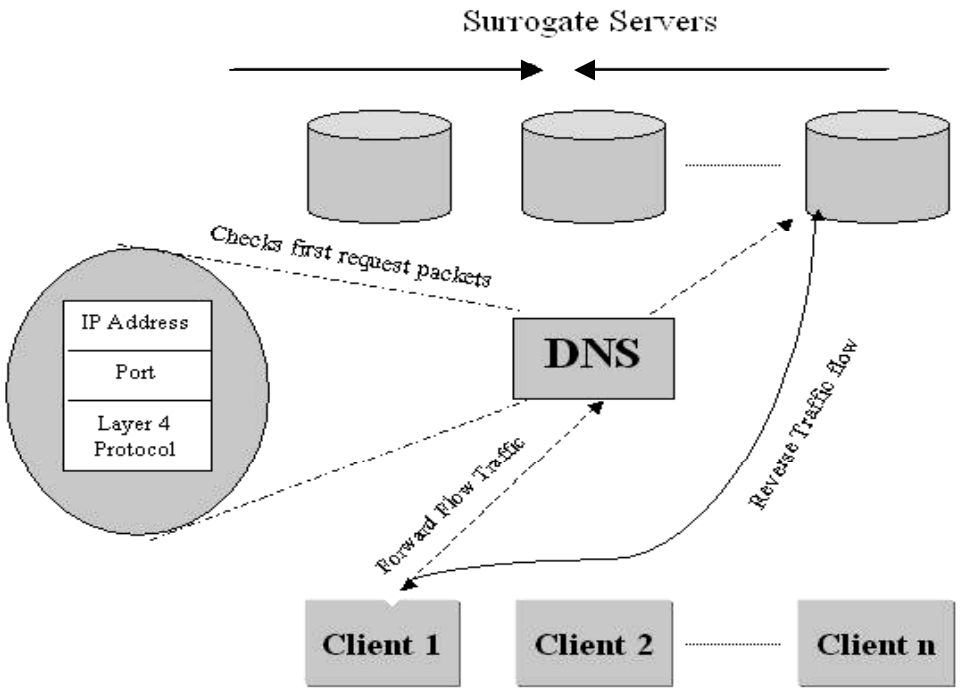

Fig2: Transport Layer Request-Routing Architecture

Application layer Request Routing

Application layer based request routing is implemented in DNS as like transport layer request routing approach. In this even more fine-grained request routing method is established for better efficient control. The first packet is imported even more closely than transport layer header, which exposes to the client IP address and with objects enable system for better selection of surrogate

Application layer based request routing is classified into two types

1. Header Inspection
2. Content Modification

Header Inspection

In HTTP [10], RTSP [11] and SSL application level protocol's first portion of the header gives information about how request should be directed towards the surrogate. So it is easy to categorize the different service request and direct to appropriate surrogate server. Basically two different methods are used to inspecting the header Universal Resource Locater (URL) and MIME request header based implementation.

Application level protocol such as HTTP and RTSP [11] describe the URL [7] methods, which uses prefix URL for content request in decision-making. The URL based requesting routing uses 302 redirection and in path element method for its routing and directing towards the surrogate server.

MIME header helps in identifying the type of client device request, like, voice browser, PDA, Cell phone or wireless nodes, which needs special content delivery. These issues are implemented with the help of cookie languages, user agents for decision-making. In MIME header site-specific identifier method helps to authenticate and identify a session form the client that is used in application level protocol like SSL.

Content Modification

Content routing [12] sustain in the central part of the Internet performed by content routers. This technique is used to take routing decisions without any special switching device between the server and the client. The main advantage of this content routing is the client can access the origin server directly. Only gateways, firewalls and Border Gateway Protocol (BGP) level routers have to be content router. Clients instigate content requests by contacting a local content router. Each content router preserves a set of name-to-next-hop mappings in a routing table.

In broad-spectrum, the technique takes advantage of content objects that consist of basic structure that consist of references to additional object called embedded objects. Most web pages comprise HTML document (plain text) that contains some embedded objects, like JPEG images. Embedded HTML directives contain embedded objects that are used as a reference in the particular web pages. Hence, only these meticulous objects will be retrieved from the surrogate server. Now, the content provider with respect to the embedded objects that are retrieved from the best origin server will modify HTML web pages. The technique is also referred as URL rewriting. In general two types of URL rewriting are executed, namely A-priori URL Rewriting, On-Demand URL Rewriting.

Content server's authenticity verifies the signature on initial routing update. If a content peer becomes unreachable, then all the contents available through that peer are unreachable as well. Routing advertisement from content servers also includes a measure of the load at that server, specified in terms of the predictable response latency. Content

modification techniques must not violate the architectural concepts of the Internet [13]. Special deliberation is made to guarantee the task of modifying the content is performed in a consistent manner with RFC 3238, whether it checks operations or modifications on content is done [13].

URL Based Request Routing

In the Default DNS based approach (First approach), If client wants to request for a content from a surrogate server which is located apart, the client will establish a TCP connection with nearest DNS server for resolving the URL to IP address. The DNS server gives the IP address back as a response to the requested client. The client creates HTTP header with the resolved IP address and sent to default gateway. The IP address in the HTTP header is look up in the routing table and if it's in the routing table, the HTTP header request is then forwarded to the surrogate server via the particular interface as shown in the fig3.

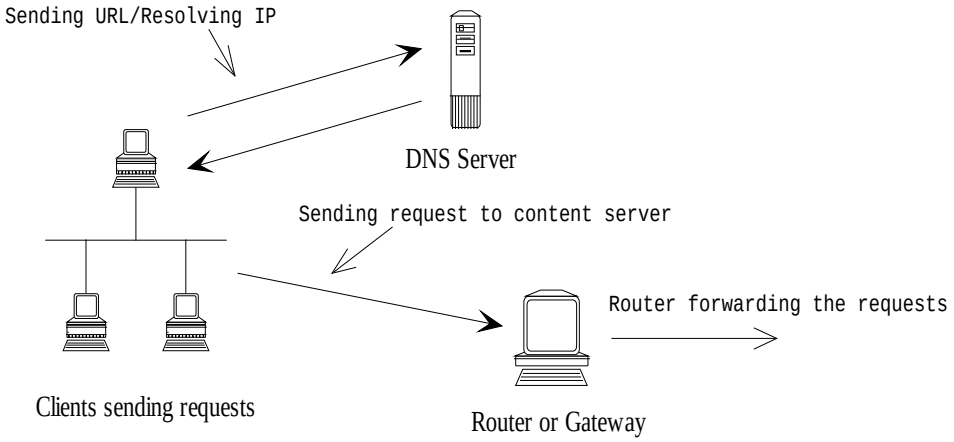

Fig3: Default DNS to resolve IP and sending HTTP requests to Router.

In the second approach as shown in the fig3 the DNS look up table for resolving the URL to IP address is implemented using Message Digest 5 algorithm [14]. MD5 algorithm use digests to reduce variable length URL string in to fixed length digest value. The Digest value is used as a key in the DNS hash table to map corresponding IP address. But in default DNS server, the entire URL is stored as the key and to retrieve the IP address, linear searching algorithm is implemented.

In the new URL routing [15] approach as shown in the fig4, instead of sending the resolved IP address back to the client, the client request is modified and the resolved IP address is included in the HTTP request in URL router itself and the corresponding IP address is looked up in the routing table. The proposed new method reduces the latency time and also reduces the traffic between the clients and the URL router. Using the proposed method, we have reduced the IP resolving time and reduce the latency of the client by approximately 50%.

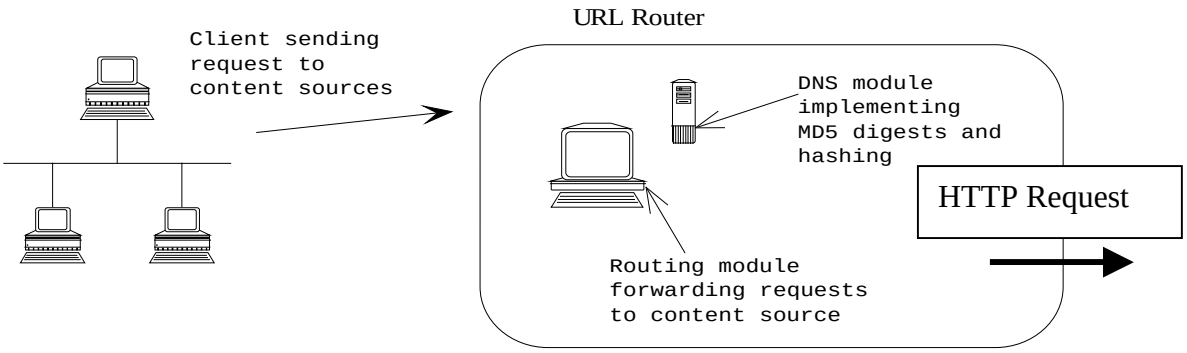

Fig4: URL routing where DNS table is Inbuilt in the Router

General Equation

$$Do + DLo + Ro + RLo \quad \text{-----} \quad \text{Equation (1)}$$

Where,

Do - TCP connection Establishing Time (millisecond) from client to DNS server.

DLo - DNS table Look-up Time (millisecond) to resolve IP address from URL.

Ro - TCP connection Establishing Time (millisecond) from client to Router.

RLo – Routing table Look-up Time (millisecond).

Co is a constant for all the three above mentioned methods and given *by* $Co = Ro + RLo$

Default DNS:

$$Do + DLo + Co \quad \text{-----} \quad \text{Equation (2)}$$

MD5 Hashing DNS table:

$$Do + Ho + Co \quad \text{-----} \quad \text{Equation (3)}$$

URL Routing:

$$Ho + Co \quad \text{-----} \quad \text{Equation (4)}$$

Here *Ho* is a MD5 Hashing method for DNS look up table.

Discussion

DNS based request routing is easy and simple to implement but some of its limitations are: Domain level resolution is only possible in DNS based request routing. Not all DNS realization is standard. Name server based DNS request routing system supports only with the information of client site server, which has the database of short time-to-live values (TTL). Some time it can cause timeouts and lead to exception handling condition. So, choosing the value of TTL is very critical. DNS server can allow recursive resolution of DNS name. For example, Content Network can resolve lesson.test.edu, but the request for the resolution might come from dns1.test.edu as a result of the recursion and might allow additional overheads. DNS based request routing techniques can suffer from serious limitations. The use of such techniques can overburden third party DNS servers, which should not be allowed [16]. RFC 2782 provides warnings on the use of DNS for load balancing [8].

The overhead associated with transport-layer Request-Routing [16] is healthier suited for long-lived sessions such as RTSP [11] and FTP [17]. In general, transport-layer Request-Routing can be combined with DNS based techniques. Hence, the DNS based methods could be used as a first step in deciding on an appropriate surrogate with more accurate refinement made by the transport-layer Request-Routing system.

An application-layer based request routing system is using application-layer *anycasting* [18]. The process could be performed in real time at the time of the object request. Object-specific control of server loading is done using URL based application layer request routing. Header based application routing can be used to direct traffic to a language-specific delivery node.

Content based approach experience many limitations, such as, the initial request from a client and all the embedded objects of the HTML web pages are to be served from the specific site of the origin server [19]. Non-cacheable pages can be marked to be cacheable only for a relatively short period of time. Rewritten URLs on cached pages can cause problems, because they can get outdated and point to surrogates that are no longer available or no longer good choices. Even though content based routing

has a limitation, the overall goal is to improve scalability and the performance for delivering the modified content, including all embedded objects.

Combination of different mechanisms can be beneficial and advantageous over using one of the mechanisms alone. Content modification can be used together with DNS Request-Routing to overcome the resolution granularity problem in DNS Request-Routing. Using DNS Request-Routing, requests for those objects can now dynamically be directed to different surrogates. With content modification, references to different objects on the same origin server can be rewritten to point into different domain name spaces.

In the equation (1) the TCP connection Establishing Time from client to Router (*Ro*) and Routing table Look-up Time (*RLo*) is constant (*Co*) for all the three methods. Using default DNS server the equation (2) represents the time to pass HTTP request header from the nearest router to surrogate server. In default DNS the searching time for resolving the IP address depends on number of URL entries in the DNS table i.e. *O(n)*. In equation (3) MD5 hashing method used in DNS server, *Ho* is the resolving time or computational time of the given URL. By means of hashing, the IP address is resolved in *O(1)* from the DNS table. Comparing the MD5 hashing method with default DNS server method *Do and Co* are constant. The difference in searching time for both methods is *[O(n) – O(1)]*. The latency can be reduced by *[O(n) – O(1)]* times to the users in the internet. When comparing with URL routing, the DNS hashing is inbuilt in the URL router. So in the equation (4), *Do* can be ignored and using URL routing the latency is reduced by *{Do + [O(n) – O(1)]}* times to the internet users.

Conclusion

This paper discusses a comparison on different types of request routing methods used in content distribution networks. The importance of DNS based and application based request routing schemes with all the merits and limitations of different approaches are compared. A novel framework for URL routing with MD5 hashing method used in DNS look-up table has been proposed in this paper. It is also identified that URL routing scheme would improve the performance and scalability of request routing in CDN.

References

[1] www.akamai.com -“Internet Bottlenecks”.

[2] B. Cain, F. Douglis, M. Green, M. Hofmann, R. Nair, D. Potter, and O. Spatscheck, *Known CDN Request-Routing Mechanisms*", November 2000.

[3] M. Day, B.Cain, and G. Tomlinson, "*A Model forCDN Peering*", November 2000.

[4] Md. Humayun Kabir, Eric G. Manning, Gholamali C. Shoja., “Request-Routing Trends and Techniques in Content Distribution Network” Parallel, Networking, Distributed Applications (PANDA) Laboratory, University of Victoria , Canada.

[5] Eastlake, D. and A, Panitz, "Reserved Top Level DNS Names", BCP32, RFC 2606, June 1999.

[6] Mockapetris P.,"Domain names - concepts and facilities", STD13, RFC 1034, November 1987.

[7] Mockapetris P., "Domain names - concepts and facilities", STD13, RFC 1035, November 1987.

[8] Gulbrandsen A., Vixie, P. and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2002.

[9] A.Barbir et al., “Known CN request-routing mechanisms.” Internet Draft (draft-ietf-RFC 3568, July 2003.

[10] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[11] Schulzrinne, H., Rao, A. and R. Lanphier, "Real Time Streaming Protocol", RFC 2326, April 1998.

[12] Mark Gritter, David R. Cheriton, "An Architecture for Content Routing Support in the Internet", http://www.dsg.stanford.edu/papers/contentrouting/2001

[13] Floyd, S. and L. Daigle, "IAB Architectural and Policy Considerations for Open Pluggable Edge Services", RFC 3238, January 2002.

[14] R. Rivest, “The MD5 Message Digest Algorithm”, November1992.

[15] Zornita Genova , Kenneth, “ Managing routing tables for URL routers in CDN, International Journal of Networking Management, 2004.

[16] Shaikh A., "On the effectiveness of DNS-based Server Selection", INFOCOM 2001, August 2001.

[17] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, October 1985.

[18] Huston, G., "Commentary on Inter-Domain Routing in the Internet", RFC 3221, December 2001.

[19] K. Johnson et al., "The measured performance of content distribution networks", Proceedings of the Fifth International Web Caching Workshop and Content Delivery Workshop 2000, May 2000.