Content Delivery Networks: An Introduction

White paper

Contact:

Networking Products Division
HCL Technologies Ltd.
49-50 Nelson Manikkam Road
Chennai- 600 029, INDIA

May 2002

http://cdn.hcltech.com

Contents

Introduction: What are Content Delivery Networks?

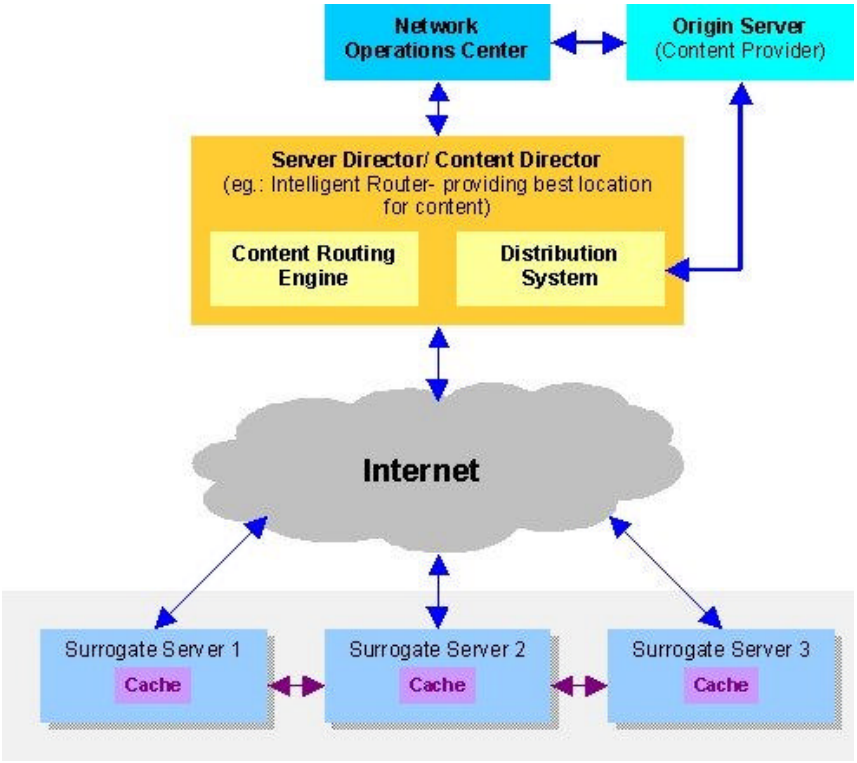
Content Delivery Network, as the name suggests, is a network of machines which delivery content, which may be static or dynamic data on the web. CDN encompasses many different technologies, all with a common goal of improving the Internet performance. CDNs require the ability to detect changes to existing data or to detect availability of new content at origin servers.

CDN is necessarily a complex system with many components. These components are distributed across different nodes of a network in a possibly heterogeneous environment. These components are servers with replicated content in them all over the world. CDNs typically take care of redirecting customer requests to a server topologically placed *near* the customer. Thus, customer gets the advantage of getting data requested at a much faster rate, from the nearest server.

Two of the major concerns addressed by CDNs are ensuring efficient content distribution and freshness of content given to the customer. An enterprise of ISPs would never accept misses on certain objects. Retrieving large objects across the Internet during bandwidth-constrained hours could result in unacceptable latencies. Pushing time-sensitive data (e.g. news, share prices, entertainment, live sporting events) to servers to ensure fresh content is a challenge, which CDN solution providers face. Most important of them is efficient distribution of all content to the respective servers.

Typical CDN Architecture

Essential infrastructure for CDNs would typically follow the ideology that it is necessarily a complex system with many components. This requires a management entity (i.e. network operations center), which intelligently monitors and manages the whole system.



Then there are servers (i.e. shown as surrogate server 1, surrogate server 2, surrogate server 3 in the diagram) cache associated with them. The cache actually cache the data so that when a customer views the same content next time, it is furnished fast. In case the content is not present in the cache, it is the responsibility of the respective server associated with cache to get content from a source which is topologically nearer to it.

Metrics for redirecting requests could include network proximity derived from network routing tables (e.g. border gateway protocol), topological proximity (i.e. depending on region), load balancing with servers (i.e. finding out the server with less load in a particular region). In case the respective server finds that content is not actually present in nearby locations, it sends a request to origin server and gets content. CDNs also process logs from these servers and use it for billing purposes.

Server Director/Content Director

This intercepts all requests and directs them to servers topologically nearest to the client or end-user. Server director avoids situations where access to Internet is impossible as a result of server problems. Assuming content is present in three servers in a region (i.e. surrogate server1, surrogate server 2 and surrogate server 3) and a request comes from a client topologically nearer to surrogate server 1 with the requested content. When request arrives, assume that surrogate server 1 is down.

In a situation like this, server director would not let the request go to surrogate server 1. Instead, it passes on the request to surrogate server 2 or surrogate server 3 depending on factors like topological proximity, network proximity etc.

Server director keeps track of health of all servers in the server farm and is able to detect failure immediately and take appropriate action. Server director handles directing requests to a least loaded server. In case all servers are out of service, then requests should be sent to origin server. There could arise a situation where content is being fetched from one of the origin servers, and at this instant, one of the servers in server farm comes up. During this time, server director should be able to switch/redirect the request to the server with the content cached and has come up. This leads to efficient utilization of bandwidth and faster access of content to client.

Mechanism for direct access to origin server

Many a times it happens that client needs access to updated information (e.g. financial news). In such a case the server director would have a mechanism to redirect all requests to origin servers.

Single point of control

Server director needs to take care of all servers, which would be distributed over great distances from a single point of control. All servers in server farm have a cache associated with them, which cache content delivered over HTTP, FTP or NNTP. These are characterized by having some proportion of static content. Essentially, servers should potentially be able to detect whether content requested is present in cache or not. Based on this, server should pass on the request to origin server. On getting response from origin server, the content should be delivered to end-user, and should cache a copy of it in server's cache.

Efficiency with cache

With the increase in usage of Internet on a day-to-day basis, we realize that cache hit rates are growing exponentially with respect to growth of web content. It is also unlikely

that capacity of cache increases with respect to growth of web content. So efficient utilization of cache is a must for better performance and with hundreds and thousands of caches over the Internet, performance improvement could be substantial.

Effective cache management involves finding out the objects which are to be present in cache and which need not. Most of algorithms which come into picture here take into account the probability of the requested object being accessed a number of times in recent past.

The CDN advantages

- Faster response time due to factors like geographical proximity, network proximity etc.
- Providing support for different types of content including on-demand, streaming media
- Providing support for secure delivery of content
- Efficient distribution of content to all the resources (typically all the servers in the content delivery network)
- Provide unique ways to improve performance of network thereby providing ways to utilize bandwidth efficiently

CDN service providers

Some of the service providers who specialize in content delivery networks:

Akamai	: www.akamai.com
Digital Island	: www.digitalisland.com
Globix	: www.globix.com
Mirror-Image	: www.mirror-image.com
Ibeam	: www.ibeam.com
CacheWare	: www.cacheware.com
Inktomi	: www.inktomi.com
Cache Flow	: www.cache-flow.com

Peering between CDNs

Content peering allows multiple content delivery network solution providers to inter-operate with each other. It is very much possible that two CDNs are do not have the same underlying technology / architecture implementation. It is also highly unlikely that a single CDN solutions provider is spanning multiple geographies. From this, it is obvious that content peering or CDN peering is a must in terms of current trends.

Assume that there are two CDNs peering. Both CDNs would have their own independent content directors, which handle request routing, load balancing, and surrogates furnish data to client or end-user. For peering to take place, there has to be some form of communication between components of independent CDNs. This requires a gateway for communication.

The end-result would be a virtual network in which all components of different CDNs have work in unison to deliver content to end-users. There is an alliance of service providers and content providers namely content alliance, which created content-peering group. The content-alliance was formed to facilitate interoperability of independent CDNs. Although content alliance group is focussed on standards that would help CDNs peer, the content-bridge alliance is focussed to new models that offer content providers optimal performance and network reach.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.