Meeting Worldwide Demand for your Content

Evolving to a Content Delivery Network

A Lucent Technologies White Paper
By L. R. Beaumont

4/25/01

Table of Contents

List of Figures

Introduction

Many Internet content providers start out with a simple "self hosting" network. With this basic configuration, content exists on one or more co-located servers. Users from around the world (or at least from around the market area) are then directed to these origin servers by a Domain Name System (DNS). This arrangement works well until the demand for content begins to exceed the server or network capacity. At that point, problems start to arise in the form of poor response for users who are located far from the origin servers.

This paper shows how Web Caching, Web Switching, and intelligent DNS server equipment can provide a path for you to evolve your simple "self-hosting" networks into efficient, globally responsive content delivery networks.

DNS Basics

Before moving forward, it is necessary to grasp the basics on how a DNS works. The DNS plays a critical role in Internet communications as the standard Internet protocol and distributed database, defined by RFC 1035 and many other documents. A DNS translates alphanumeric host names such as *www.foo.com* into an IP address such as *135.3.22.224*.

Each DNS works as a hierarchical network of cooperating servers where the local server resolves the host name if it can. If it cannot resolve the name, the local server returns a Name Server (NS) record to identify another DNS server that can likely resolve the name. This process continues until the authoritative DNS server for the host is queried.

The authoritative server informs the system of the IP addresses for the hosts for which it is the authority. This is accomplished by returning an address (A) record and assigning it a time to live (TTL) value. Each DNS server in the path from the client to the authoritative DNS retains this address record for the length of time specified by the TTL. This greatly reduces the number of requests that need to be handled by the authoritative DNS server.

Relieving Server Congestion

One common approach for evolving simple "self-hosting" networks is to install server accelerators to relieve server congestion. Server accelerators[1], also known as reverse proxy or surrogates, are placed in front of your server farm to handle most of the load. Figure 1 (following) provides an illustration of this approach.

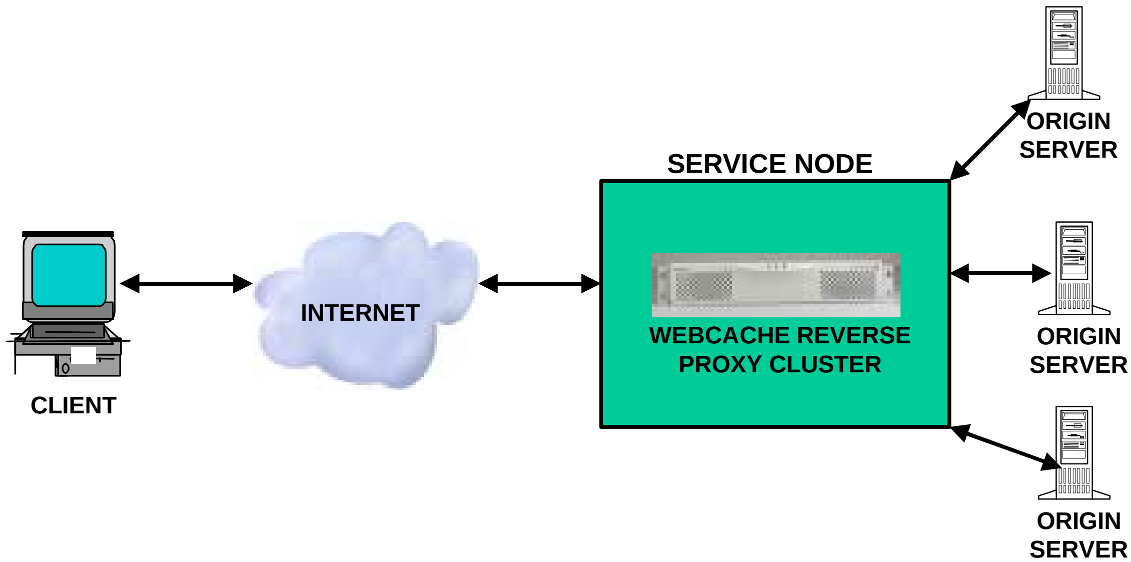

Figure 1 – Relieving Server Congestion with Reverse Proxy Approach

As shown in Figure 1, the Reverse Proxy approach works as follows to relieve server congestion:

1. Using a Web browser, the client requests content from the origin server.
2. The Web browser is directed to the appropriate WebCache Reverse Proxy server, which obtains the requested content from the origin server.
3. The WebCache Proxy Server then forwards a copy of the requested content to the browser and creates a copy of the content in its cache store.
4. Subsequent requests for the same content are then served by the WebCache Proxy Server without requiring access to the origin server.

The key component in the server accelerator is a Web cache[2,3] device that stores your content and then retrieves it on request. By serving most of the requests in place of the origin server, the Web cache can help you improve server performance, increase content delivery speed, and easily handle spikes in demand. In this example, the Web cache device used is the Lucent Technologies ***imminet*** WebCache[4].

While the reverse server proxy approach does provide some significant benefits, it also introduces the Web caches as a single point of failure and limits your capability to scale beyond a single Web cache.

Scaling up

The next step in network evolution is to scale up to a load-balancing device, such as the Lucent ***imminet*** WebDirector[5]. This device allows the reverse proxy solution to be scaled to handle very large traffic loads. The WebDirector can also provide health checks and reconfigure your network to remove any failed Web cache devices. The addition of a load-balancing device accommodates switched reverse proxy configurations, as shown in Figure 2:

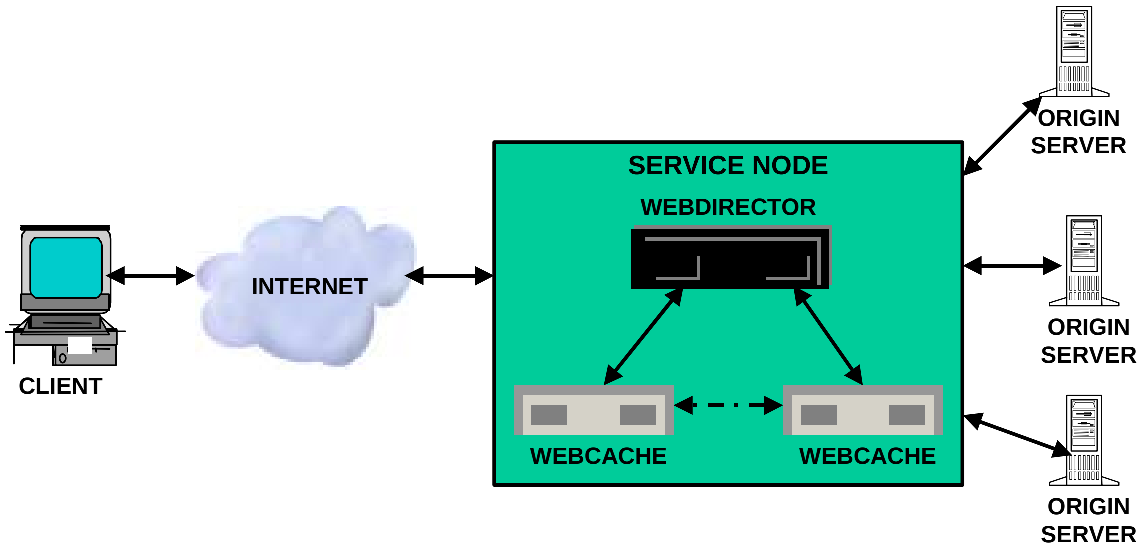

Figure 2 - Relieving Server Congestion with Load Balancing

In this diagram (Figure 2), the WebDirector is added to load balance the WebCaches and provide additional redundancy and performance. With this configuration, the client is directed to the WebDirector for access to the WebCaches. A configuration with greater availability can be built using two WebDirector switches (an active and standby) for a highly reliable and scalable platform.

While this approach provides you with an excellent solution for relieving server congestion, it does not resolve network congestion or provide content close to users located far from the origin server. To solve these problems requires you to place server accelerators across the service area and navigate requests to those servers.

Going Global

To truly “go global,” you must set up “server accelerators” or service nodes around the world. Then, you need an intelligent device that can route content requests to the closest service node. With this type of arrangement, your capacity could be scaled without limit and your users would be assured of always getting content from the service node nearest to them. Figure 3 illustrates how content would be served in this global configuration.

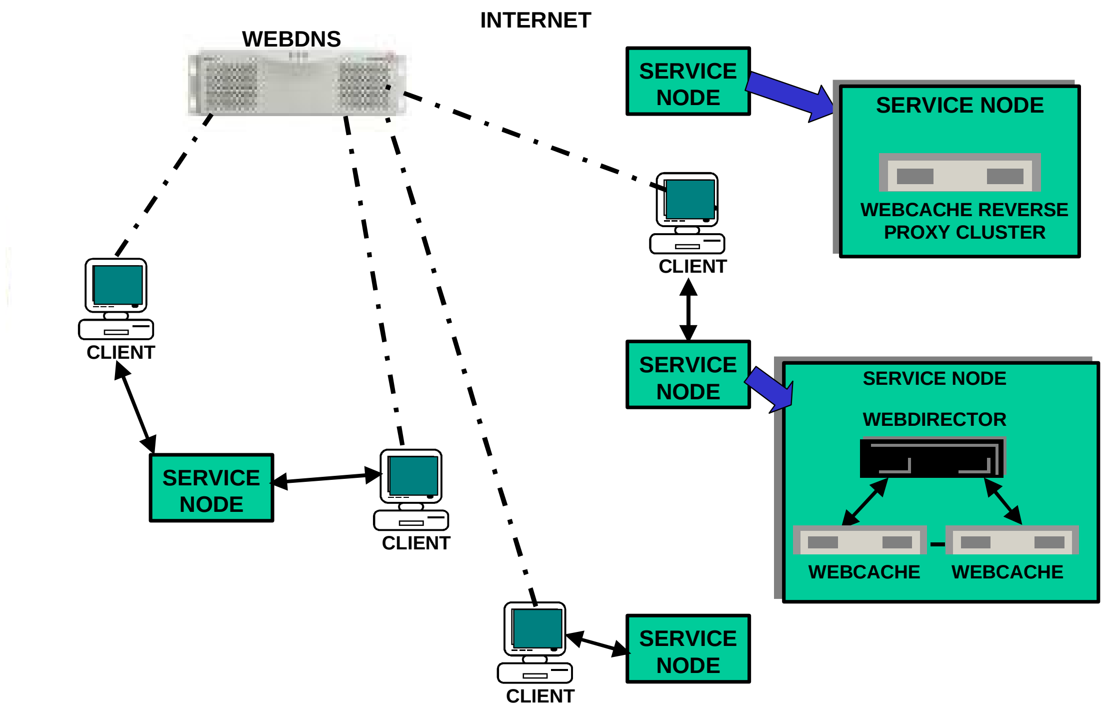

Figure 3 – A Global Approach for Content Delivery

You can build this efficient Content Delivery Network solution today with the ***imminet*** WebDNS product. To deliver the fastest response time to clients around the world, you can place service nodes in strategic sites worldwide to move content closer to the users. And by configuring the WebDNS as the authoritative DNS[6] server for your content site, you can help ensure that client requests are directed to the service node that is closest to each client. This request navigation capability allows you to:

- Increase content availability
- Expand network scalability
- Relieve congestion
- Handle peak traffic loads
- Reduce the load on the origin server

Figure 4 illustrates how the WebDNS product enables this processing to work in conjunction with an existing DNS.

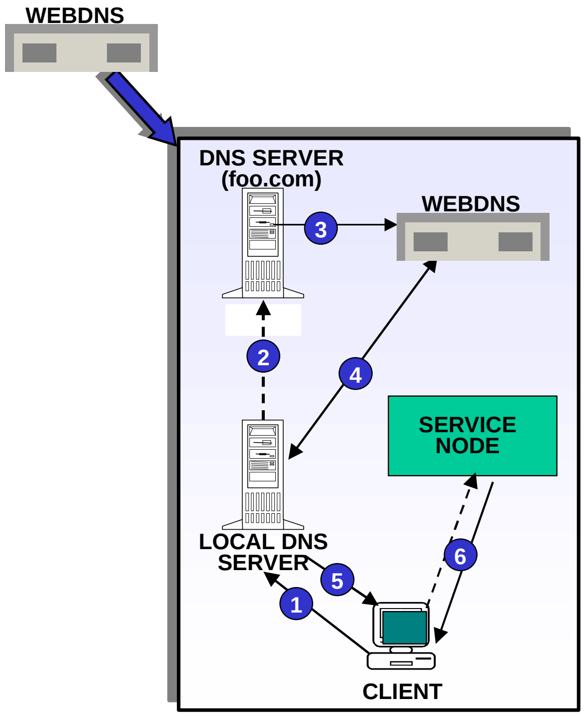

Figure 4 – WebDNS Processes for Content Delivery

As shown in Figure 4, the WebDNS performs the following processing steps:

1. The client wants to access a page and embedded objects at http://www.foo.com/ so it sends a DNS request to the local DNS server.
2. The local DNS server contacts the authoritative name server for the domain “foo.com.”
3. Rather than return an address record, the authoritative name server redirects the request to the WebDNS at a Data Center.
4. The WebDNS determines the correct service node for the client to go to and returns this information (as an address record containing the IP address of the Service Node) to the local DNS. In this step, the WebDNS bases its decision on the network address for the client’s local DNS, rather than the client’s address. Typically, the client will still be mapped to an optimal service node.
5. The local DNS server forwards the response to the client.
6. The client requests content from the optimal service node.

As shown, the WebDNS, in conjunction with your local DNS servers, offers you a versatile, global solution that can help you:

- Reduce end-user latency by directing requests to the closest service node

- Monitor your network for unavailable or out-of-service nodes, resulting in the highest network availability
- Detect failures and automatically route requests to the best service node
- Share loads across a distributed array of service nodes
- Perform easy configuration for full compatibility with existing Domain Name Systems
- Interwork seamlessly with ***imminet*** WebCache and WebDirector solutions
- Continuously monitor the Internet to learn where the closest node is to each client
- Collect statistics on the operation of the WebDNS and each service node

Conclusion

Lucent ***imminet*** products, including the WebCache, WebDirector, and WebDNS products, let you seamlessly migrate from basic self-hosting configurations to globally responsive content delivery network solutions. These solutions offer effective Web caching, load balancing, and authoritative DNS to help you relieve server congestion and evolve your existing network to meet worldwide demand for content. And, these solutions are scalable, easy to install and administer, provide a manageable growth path, and deliver exceptional service to end users, worldwide.

Glossary

Cache: Taken from the French *to store*, in computing, caching refers to the storage of recently retrieved computer information for future reference. The stored information may or may not be used again. A cache is a program's local store of response messages and the subsystem that controls its message storage, retrieval, and deletion. A cache stores cacheable responses in order to reduce the response time and network bandwidth consumption on future, equivalent requests. Any client or server may include a cache, though a server that is acting as a tunnel cannot use a cache. Note: When used alone, the term "cache" often denotes a "caching proxy."

Caching proxy: A proxy with a cache, acting as a server to clients, and a client to servers. Caching proxies are often referred to as "proxy caches" or simply "caches". The term "proxy" is also frequently misused when referring to caching proxies.

CDN: Short for Content Delivery Network, a network capable of delivering content to users.

Content: Content is an abstract representation of a particular sequence of digital bits that can be rendered by a computer program. Content is comprised of one or more digital data streams of bytes that can be translated by a rendering machine for human use. A simple example is an HTML Web page. Another example might be a James Bond movie comprised of several resolution encoding and different language tracks. Application programs are also considered content.

DNS: Short for Domain Name System, an Internet service that translates domain names into IP addresses. While domain names are alphabetic, and easier to remember, the Internet is really based on IP addresses. As a result, every time you use a domain name, a DNS service must translate the name into the corresponding IP address. For example, the domain name www.example.com might translate to 198.105.232.4. A comprehensive group of DNS resources is available at: http://www.dns.net/dnsrd/.

Request navigation: Routing of the client request to an origin server or Web cache from which the requested content can be retrieved.

Reverse Proxy: A configuration where a cache adopts the IP address sought by the client so that the DNS resolves the server name to the Web cache, rather than to the origin server. If the requested object is available in the cache, it is delivered to the client. If it is not available, then the cache queries its associated origin servers. A reverse proxy is usually run by the same organization that runs the associated origin servers. This term is a synonym for server accelerator.

Server accelerator: A synonym for reverse proxy.

Switched Reverse Proxy: A configuration where a Layer 4 switch adopts the IP address sought by the client so that the DNS resolves the server name to the Layer 4 switch, rather than to the origin server or reverse proxy. Based on a load-balancing algorithm, this switch then forwards the request to one of several caches that are acting as non-switched reverse proxies. (Contrast with reverse proxy.)

Surrogate: A synonym for reverse proxy.

References

[1] ***imminet*** Server Acceleration, offer description available at: http://www.lucent.com/serviceprovider/imminet/images/imminet-ServerAcceleration.pdf

[2] *Web Caching*, O'Reilly, Duane Wessels, to be published April, 2001

[3] *Web Proxy Servers*, Prentice Hall, Ari Luotonen, 1998

[4] Lucent ***imminet*** WebCache, http://www.lucent.com/serviceprovider/imminet/webcache.html

[5] Lucent ***imminet*** WebDirector, http://www.lucent.com/serviceprovider/imminet/webdirector.html

[6] *DNS and BIND*, O'Reilly, Paul Albitz & Cricket Liu