Distributed Web Server Architecture: A Framework for Global, Fast, Reliable Content Delivery

Brad Block

March 1999

Abstract

Current methods of delivering content quickly and reliably over the web can be hard and expensive to implement. This paper proposes an alternative fast, easy, and economical solution that provides low-latency initial response to customer requests followed by checking for a better server and redirecting the customer to that better server if one is available. The solution utilizes aspects of Internet routing protocols to route using shortest paths first, TCP probes to determine distances between hosts, intelligently managed web server clusters, and monitors to track the performance of servers. The solution builds on round-robin DNS and the Cisco DistrbutedDirector but provides only abstract specifications and no demonstration of an actual working implementation.

6.033
Professors F. Kaashoek and S. Seshan
TA C. Boyapati
To be submitted for **Phase II** consideration

1 Introduction

Demands on Internet web servers are increasing beyond the capacities of single servers or clusters of servers. This global demand dictates that providers move from serving requests through a single point-of-presence to a distributed server architecture. Service should originate as close to the customer as possible and loads should be distributed intelligently across many clusters located at multiple points-of-presence. Current architectures aimed at meeting this demand have not been able to deliver either high capacities, close proximity to customers, or sufficiently low costs of installation or operation.

The popular and simple Round-Robin approach (Brisco, 1995) provides unintelligent load-balancing across several local servers through low time-to-live DNS records. It fails to accommodate the needs of a globally distributed architecture to satisfy requests close to their origin. An approach by Cisco called DistributedDirector (Delgadillo, [4]) is a very dollar-intensive solution that also pollutes the Internet with much overhead traffic when satisfying each customer request. The approach outlined below will fuse the strengths of these two approaches with additional refinements creating a superior distributed web server architecture. This paper will provide a high level abstract specification for a global architecture. Very specific details of protocols, data structures, and local web server cluster management are out of the scope of this paper and are mostly left to the implementers.

2 Distributed Web Server Architecture (DWSA)

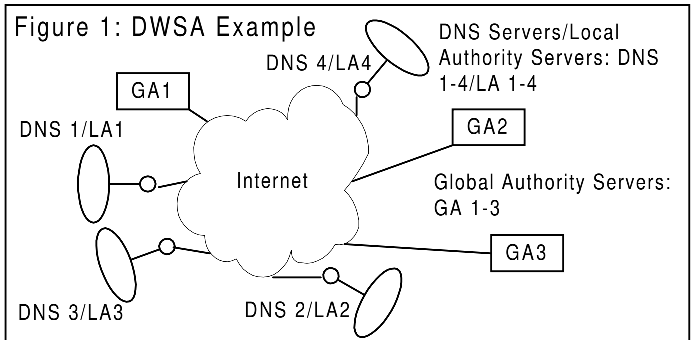
2.1 Overview of Architecture, Basic Terms, Definitions

To retrieve content over the Internet, the client or customer retrieves a URL (Berners-Lee, Masinter, and McCahill, 1994) that names the content of interest. The customer goes through several steps to retrieve the URL. Between these steps, the distributed web server architecture or *DWSA* performs behind-the-scenes calculations to service the request using the web server that is closest to the customer and that has sufficient capacity to deliver the content quickly. Throughout section 2, the description of the DWSA will use an example of a customer located in Ohio trying to retrieve URL's for www.acme.com which has web servers in California, New York, and Georgia.

The customer first resolves the web server name specified in the URL by querying a DNS (Mockapetris, 1987) server in the DWSA. The DNS server maps a web server name to a specific IP address that dictates the location of the web server. Once the customer receives a response from the DNS server, the customer issues a request directly to a particular web server using the IP address supplied by the DWSA DNS server. The DWSA provides the illusion of proximity and capacity by first delivering, with no delay, an IP address of the closest web server, then taking time to determine the best web server, and finally redirecting the customer to that best web server. In the example, the Ohio customer can contact DNS servers located in California, New York, or Georgia but would automatically be initially directed to and satisfied by the closest DNS server and web server located in New York.

Each DNS server in the DWSA (marked D1-D4 in Figure 1) has two interfaces to the Internet. One interface shares a single IP address with all other DNS servers. The second interface has an IP address unique to that DNS server. The shared address comes from a pool of addresses called the *shared virtual net* (labeled in Figure 2). The unique or private address comes from a pool of addresses unique to a DNS server called a *private net* (labeled in Figure 2). The shared address interface allows any DNS server to answer customer requests in place of any other DNS server. The private address interface allows a DNS server to uniquely name and communicate with another specific DNS server. Shared and private only indicate which servers can use the addresses on interfaces. These terms do not imply visibility from the Internet. In fact, both shared or private address interfaces may be accessed from the Internet.

Figure 1: DWSA Example



DNS servers paired with local authority servers. DNS servers have an interface with shared virtual net address and an interface with an address from the private net used by their paired local authority server.

In addition to DNS servers, shared virtual nets, and private nets, there are two classes of authority servers. A *local authority server* (marked LA1-LA4 in Figure 1) manages a cluster of web servers and distributes customer requests over the cluster. A local authority server has a unique private net address. An example of a local authority server implementation is the Cisco LocalDirector (Cisco, [8]). In this paper, local authority server is synonymous with intelligently managed autonomous cluster of web servers. Intelligently managed is defined to mean only distributing traffic to available web servers. In addition to local authority servers, there are several *global authority servers* (marked GA1-GA3 in Figure 1) that maintain a list of connected DNS servers and local authority servers and statistics about their operating state. A global authority server has a unique IP address. These global authority servers provide each DNS server with information about available servers to facilitate intelligent redirections.

In the example, there are local authority servers in California, New York, and Georgia. Each local authority server shares a unique private net of addresses with the DNS server to which it is paired in that region. The global authority servers in this example could be anywhere in the Internet but would probably best be positioned somewhere between all the DNS server/local authority server installations (e.g. Texas).

2.2 Receiving requests from the client

2.2.1 Routing customers to the closest DNS server using selection between multiple advertised routes

Each DNS server sends route advertisements to the Internet about its interface addresses. Because each DNS server shares one of its interface addresses with all other DNS servers, there appears to be many advertised routes to that single shared address (Halabi, 1997). This effectively creates a huge virtual DNS server having several interfaces to the Internet. A host on the Internet routing traffic to that shared address does not realize (or care) that the machine behind the address is actually many machines in different locations. In other words, a host on the Internet can take many different routes (labeled Route 1-3 in Figure 2) to reach the single IP address shared by the DNS servers.

However, Internet routing protocols will try to route traffic using the shortest route to the destination. Internet routers (not the DWSA) calculate route distances by counting the number of hops from the route advertisement origin (i.e. the DNS server) and by using other metrics. When there is more than one route to a destination, the Internet router usually keeps the shortest route (Halabi, 1997). DWSA takes advantage of this open-shortest-route-first policy in assuming that customers will be first directed to contact the closest DNS server.

Figure 2 shows the customer in Ohio first contacting the DNS server in New York. The customer has the option of contacting the DNS server in CA or GA because those servers are also advertising a route to the same IP address as the NY DNS server. However, the Internet routers between the Ohio customer and the DNS servers would elect to route traffic to NY because of NY's close proximity (in terms of fewest route hops) to Ohio.

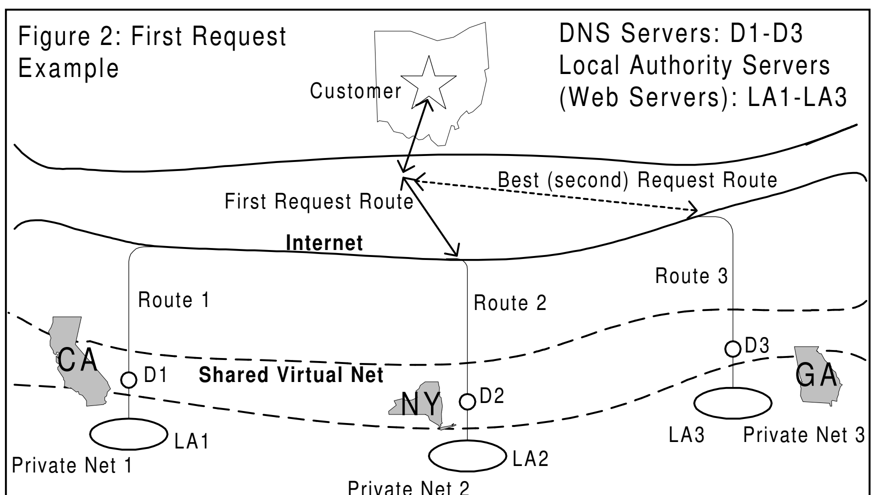


Figure 2: First Request Example

Customer's initial DNS request routed through Route 2 to D2 in NY. D2 responds with address from LA2 to serve the first URL. D2 then calculates that LA3 would be a better server and D2 tells LA2 to redirect future requests from Customer to LA3.

2.2.2 Providing low-latency first response followed by redirection if better server is available

Each DNS server is also paired with a local authority server (LA1-LA3 in Figure 2) that is considered to be the closest to that DNS server. This pairing is done statically during the design of the DWSA. When the customer initiates a DNS request and the request is sent to the closest DNS server (over the shortest route), the DNS server initially replies to the customer with the private net address of its paired local authority server.

The customer then connects to the local authority server using this address unique to that local authority server. After the initial DNS server response to the customer, the DNS server begins to determine if there is a better local authority server (other than its own paired local authority server) to satisfy the customer's request. Always serving the first request from the paired local authority server will not introduce possible slow-downs into the DWSA since if the paired local authority server is not the best server, the customer will be very quickly redirected to a better local authority server. Also, if the paired local authority server is the best server, the customer will already be there.

Figure 2 shows the customer in Ohio first contacting DNS server D2 located in New York. This DNS server immediately replies with an address unique to its paired local authority server, LA2. D2 must also determine whether another local authority server (e.g. LA1 or LA3) would be better equipped to handle the Ohio customer's request even though those local authority servers are not as close to Ohio as LA2.

To determine if there is a better local authority server, the DNS server initially looks to a cache that contains customer addresses and corresponding best-server addresses. If this cache contains no entry or an entry that is too old (e.g. > 5 seconds), the DNS server will determine the best server using several other factors. It is important that the cache exist so that the DNS server not do unnecessary redundant work, and it is important that the DNS server not use old entries that would not reflect the real state of the loads of servers or connections.

2.3 Directing requests to appropriate servers

2.3.1 DNS server data structures enable intelligent redirection

To make intelligent redirection decisions, each DNS server must maintain several lists and tables of information about other DNS servers and local authority servers (shown if Figure 3). Each DNS server stores a list of other DNS servers' unique private network addresses. It also keeps track of the two DNS servers closest to itself

Figure 3: DNS Server Summary

Interfaces:

- Shared virtual net interface
- Private net interface

Data structures:

- Static list of global authority servers
- Dynamic list of DNS servers and local authority servers
- List of two closest DNS servers
- Complete graph of DNS server and local authority server round-trip-times

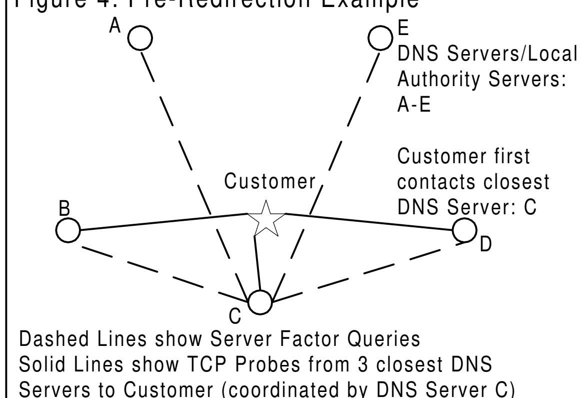
by sending periodic TCP round-trip-time probes to all DNS servers and remembering the two closest DNS servers. Each DNS server also stores a list of the private network address of all the local authority servers in the DWSA.

Each DNS server's DNS servers and local authority servers lists are obtained from the global authority servers. Each time a DNS server accesses these local lists, it checks with the global authority servers for timestamps to determine whether the local copy is current. If the local timestamp is older than the global authority servers' timestamps, the DNS server requests that the global authority servers send updates corresponding to the difference in timestamps. The DNS server may periodically request such updates even if it has no immediate need to access its lists. The DNS server uses a simple consensus algorithm in the case that global authority servers are inconsistent (Levi and Agrawala, 1994).

2.3.2 Collecting server factors to determine the state of local authority servers' conditions

A server factor is a prediction of a local authority server's ability to handle additional requests based on historical information and current conditions. Each DNS server uses its list of all local authority servers to gather server factors for all of the local authority servers. These server factors are computed by each local authority server and made available on request to DNS servers. Section 2.4 details the specifics of how each local authority server generates its server factor. Figure 4 shows how DNS server C, responding to a request from a customer, is querying all other local authority servers for their server factors.

Figure 4: Pre-Redirection Example



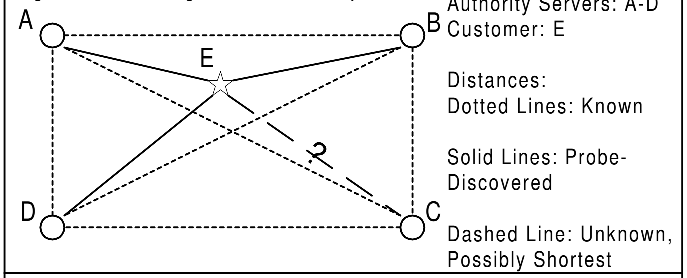
Closest to Customer, DNS server C coordinates additional TCP probing of Customer and gathering of server factors.

2.3.3 Using triangulation to estimate customer's closest local authority servers

To estimate latency and throughput (i.e. distance) between a local authority server and a customer, the DWSA uses triangulation (Matlab, [9]). The Cisco DistributedDirector (Delgadillo, [4]) uses complicated routing metrics and TCP probes from each web server to the customer to determine the "closest" server. Obtaining these routing metrics and TCP probe round-trip-times is expensive in terms of packets sent per customer request and sometimes impossible if routers do not provide this information to non-peers. Triangulation approximates a customer's closest local authority servers with less overhead than DistributedDirector.

Each DNS server maintains a completely connected graph of round-trip-times between all DNS servers and all local authority servers (shown in Figure 5 with dotted lines). This graph can be updated infrequently (order of minutes) since those round-trip-times should not change frequently. When a request comes, a DNS server sends a TCP probe to the customer and asks its two closest neighbor DNS servers to send TCP probes to the customer (shown in Figure 5 with solid lines). With these three TCP round-trip-times probes and the relatively static graph of round-trip-times between DNS servers and local authority servers, the DNS server can use triangulation (sin and cosine relations) to determine the customer's closest local authority servers.

Figure 5: Triangulation Example



The complete graph of TCP probed distances between A, B, C, D coupled with the distances between E and three other hosts allows triangulation (sin and cosine relations) to determine closest hosts to E.

This triangulation process makes a critical assumption about the routing topology of the Internet by estimating rather than testing distances of links between local authority servers and customers. If in practice, a local authority server determines that it is too far from the customer to perform the request, it could tell the DNS server to try the estimation again using a more conservative guess for the distance between the customer and that local authority server. The triangulation estimation is efficient because the overhead of 3 TCP probes per request remains constant even with many candidate local authority servers. In contrast, the DistributedDirector solution requires overhead per request that grows linearly with the number of local authority servers.

2.3.4 HTTP redirect if there exists a better server calculated from server factors and triangulation

Once the server factors have been collected and the triangulation estimations have been computed, the DNS server can instruct the local authority server to issue an HTTP redirect (Berners-Lee, Fielding, and Frystyk, 1996) upon the next customer request. The HTTP redirect will be to the private network address of the local authority server the DNS server has determined is best. A subtle point to note is that the local authority server paired with the DNS server will likely be the common case best local authority server. However, if the paired local authority server is loaded such that server latency would be greater than the additional link latency to the next best local authority server, it would make sense for the DNS server to redirect. Once the best servers are determined, they are written to the DNS server cache to be used for the next 5 seconds in the event that the customer queries the DNS server again.

In the continuing example, the Ohio customer request is initially routed to New York. After the initial reply, the New York DNS server performs server factor queries (as in Figure 4) and triangulation (as in Figure 5). It determines that the Georgia local authority server is close enough and is available to satisfy the Ohio customer request best. The New York DNS server maps the Ohio customer to Georgia in its cache and instructs its local authority server to redirect the next Ohio customer request to Georgia. This process is summarized in Figure 6.

Figure 6: DWSA: Step-by-Step

1. DNS servers advertise many routes to same IP address from shared virtual net.
2. Internet routers uses closest route to send customer's DNS request.
3. DNS server replies with address for its paired local authority server (a private net address unique to the local authority server).
4. DNS uses server factors and triangulation to look for better local authority server.
5. If found, DNS server tells its paired local authority server to redirect the customer's next request to the better server.

2.4 Collecting and propagating server statistics and information with which to make decisions

Each local authority server calculates its server factor based on a sum of weighted real-time, theoretical, and historical factors. Real-time factors are obtained by analyzing the HTTP request logs to determine current load (number of concurrent connections) and current ability to satisfy customer requests (e.g. recent throughput / retransmissions). Theoretical factors are known a priori and include server capacity (potential concurrent connections based on amount of RAM, kernel buffer sizes, local network topology, processor speed, and disk performance) and connection capacity. Lastly, historical factors are derived by studying usage patterns over time which are observed to anticipate future loads. The factor set could easily be extended to account for impact of dynamically generated content to insure that such loads are also effectively distributed.

These three sets of factors (real-time, theoretical, and historical) can be used to predict how well the local authority server will be able to cope with additional load. Furthermore, this value can be computed quickly and at low cost since all the information is readily and locally available. This information is made available to DNS servers upon request. The protocol used to transmit this information should be stateless and have low setup and takedown overhead. The protocol should involve sending simultaneous duplicate request and response packets from and to the requesting DNS server so that with higher probability, the DNS server will receive an answer in the face of network unreliability. Furthermore, if the requesting DNS server receives no reasonably fast response, it can consider the non-responding local authority server to be overloaded or otherwise unavailable due to some error.

2.5 Adding and removing servers

Adding and removing servers to and from the DWSA on-line provides the crucial ability to increase capacity and perform repairs without interruption of service. The global authority servers enable this dynamic connectedness by tracking currently connected servers. When a new local authority server or DNS server is connected, it sends a registration message to the global authority servers. This message must be successfully multicast to all global authority servers for proper operation. The global authority servers will then probe the connected servers periodically to determine if they are still alive. Specifically, the global authority servers probe local authority servers for their server factors and analyze this data to determine efficient deployment of servers (this analysis is done off-line). DNS servers are probed with TCP probes. If no answer is received from a server after some time, that server will be marked as disconnected and removed from the list of connected servers. The global authority servers record incremental changes to the list of connected servers rather than keeping a single list of connected servers. This way, DNS server requests for updates can be satisfied with information about only what has changed with respect to timestamp differences in order to save bandwidth by not resending the entire list.

3 Performance analysis

3.1 DWSA degrades poorly under excess loads

The DWSA does only so much to insure fast service to customer requests. For example, the DWSA is at the mercy of the local authority servers to actually fulfill the service request. If the local authority servers all reach capacity or the servers that are not at capacity are far away, the DWSA will be forced to direct customers to wait in long lines or commute over long distances. However, the DWSA does provide monitoring of server factors of local authority servers to determine, over time, if servers are deployed to maximize efficiency. In addition, under loads below capacities of local authority servers, the DWSA performs predictably.

3.2 Potential bottlenecks occur when local authority servers are poorly deployed

Situations such as having too few local authority servers at peak centers of customer usage can cause individual local authority servers to become bottlenecks for the DWSA. However, these sorts of bottlenecks are isolated in their impact on customers. For example, while customers in the peak usage centers will experience slow access, customers not in the scope of those local authority servers will likely see little or no degraded performance since their traffic is not routed to those highly trafficked regions. Again, DWSA can recommend better placement of local authority servers to avoid or eliminate bottlenecks as they begin to occur.

3.3 No failure in the face of much unexpected behavior

Several measures protect against slow or disabled servers or links. DNS servers attempting to determine the best server for a customer will simply ignore DNS servers and local authority servers that respond too slowly.

Outages resulting in loss of communication between the global authority servers and other servers will result in the other servers being treated as disconnected and removed from list of connected available servers. Those other servers will not reregister with the global authority servers automatically to avoid connections that may be flapping. To avoid slow service when best local authority servers becomes loaded while servicing customer requests, those best local authority servers can redirect customers back to the shared virtual net address thereby beginning the redirection process again to hopefully direct the customer to a different, sufficiently available local authority server.

Customers or search engines that "bookmark" or keep track of the best servers to which they were directed could potentially bypass attempts to always be directed to the best server. This is avoided by requiring that all local authority servers observe the HTTP referring address (Berners-Lee et al, 1996). If the referring address is not an address from the shared network or a particular private network address, the local authority server should direct the customer to the shared network address which will force it to contact a DNS server for proper redirection.

4 Conclusions and Further Considerations

4.1 The DWSA in a nutshell

The DWSA consists of a set of DNS servers, local authority servers, global authority servers, shared virtual net addresses, and private net addresses. The customer first contacts the closest DNS server as determined by intermediate Internet routers. This DNS server satisfies the request with its paired local authority server private network address. The DNS server then employs server factors and triangulation to determine if there exists a better local authority server to service the customer. If so, the DNS server instructs the local authority server to redirect the customer to the better local authority server upon the customer's next request.

The DNS server communicates with other DNS servers and local authority servers learned from the lists published by the global authority servers. DNS servers and local authority servers register themselves to be tracked by the global authority servers. Server factor histories enable global authority servers to recommend optimal placement of local authority servers.

4.2 Implementation feasibility and operational overhead

The DWSA requires simple extensions to standard DNS server and web server software to allow intercommunication and limited action based on predetermined conditions. In addition, the data structures and communication protocols used by the DNS server and web server extensions and global authority servers must be chosen for space and time efficiency. In particular most of the communication protocols should use a stateless transport and fast timeouts (if a server cannot respond quickly to its neighbor it probably would not respond quickly to the customer). The DWSA components use commonplace hardware and software and place no great additional load on the Internet in the form of administrative traffic. In addition, the DWSA can coexist with multiple instances of itself without creating excessive strain on the Internet. Furthermore, no aspect of the DWSA requires a monopoly over the Internet. In order to insure that DNS servers do not present a single-point-of-failure, several DNS servers could be used in cluster form. The DNS protocol allows more than one DNS server as authority for a domain (Mockapetris, 1987). Having multiple IP addresses in the shared virtual network of addresses would allow each DNS server installation to have several backup machines in case of server failure or overload.

4.3 Impact of fallibility of critical assumptions

Two critical assumptions were made in the DWSA. The first is that intermediate Internet routers always select the best route. The second is that distances between points in the Internet can be estimated by some knowledge of their neighbors. If these assumptions prove mostly wrong in practice, the DWSA would probably not be efficient. However, if these assumptions fail to hold only in special cases, the DWSA could take action to compensate.

References

[1] T. Berners-Lee, L. Masinter, and M. McCahill. Uniform Resource Locators (URL). Internet Request For Comments RFC 1738, December 1994.

[2] T. Berners-Lee, R. Fielding, and H. Frystyk. Hypertext Transfer Protocol -- HTTP/1.0. Internet Request For Comments RFC 1945, May 1996.

[3] T. Brisco. DNS Support for Load Balancing. Internet Request For Comments RFC 1794, April 1995.

[4] Kevin Delgadillo. Cisco DistributedDirector. http://www.cisco.com/warp/public/751/distdir/dd_wp.htm.

[5] Bassam Halabi. Redundancy, Symmetry, and Load Balancing. Internet Routing Architectures, pages 193-222, 1997.

[6] Shem-Tov Levi and Ashok Agrawala. Agreement Protocols. Fault Tolerant System Design, pages 201-230, 1994.

[7] P. Mockapetris. DOMAIN NAMES - CONCEPTS AND FACILITIES. Internet Request For Comments RFC 1034, November 1987.

[8] Cisco LocalDirector 416 and 430. http://www.cisco.com/warp/public/751/lodir/lodir_ds.htm.

[9] Matlab Delaunay Triangulation. http://ug.cs.dal.ca/matlab/techdoc/ref/delaunay.html.