WPI-CS-TR-02-27 October 2002

A Client-Based Study of Clustered and Distributed Web Content

by

Craig E. Wills
Nitin John

Computer Science Technical Report Series

WORCESTER POLYTECHNIC INSTITUTE

Computer Science Department
100 Institute Road, Worcester, Massachusetts 01609-2280

Abstract

As the Web has grown, the amount of traffic to popular Web sites has increased to the point where a single server machine cannot handle the request load and a cluster of machines at these sites is often used. Much research work on load balancing and content distribution techniques has been published with many offerings available in the marketplace that provide these services. However, little work has been published on what load balancing techniques are being *used*. A key contribution of our work is a longitudinal study on the use of load balancing techniques at a set of popular Web sites. This study was done over a two-year period using a client-based methodology that can be applied to any site. We also examine the interaction between these techniques and the cache validation and persistent connection features of HTTP.

Our results show that over 60% of 120 popular sites continue to map client requests to a single IP address indicating that transparent load balancing amongst a cluster of machines is likely being used to handle requests. We found some inconsistencies in HTTP headers returned by different machines in the same cluster. About half of the remaining sites use DNS redirection to map client requests to multiple machines with distinct IP addresses. The remainder use HTTP redirection with a smaller set of these sites actually using the redirection for load balancing.

Another contribution of our work is a study on the extent of content distribution from the origin server for a site to auxiliary servers local and outside of the site. We found an increase in the number of embedded objects being served by auxiliary servers from 50% of the studied sites in April 2000 to 69% in May 2002. In a more detailed study of 51 sites, using additional data, we found a steady increase in the percentage of sites using auxiliary servers, from 51% in November 1999 to 73% in May 2002. This increase is primarily due to the use of auxiliary servers local to a site. The use of known CDN servers increased from 8% in November 1999 to 20% in May 2001, but dropped back to 16% of the 51 sites in May 2002.

Keywords: Web clusters, load balancing, content distribution.

1 Introduction

As the Web has grown, the amount of traffic to popular Web sites has increased to the point where a single server machine at these sites cannot handle the request load. In response, these sites use clusters of machines to serve content at the site or distribute the content to other servers within or outside of the site.

A number of techniques for balancing the load amongst multiple server machines at a site have been proposed in the research literature [2, 5]. Such a collection of server machines are called server farms or web clusters and the technique by which the client is directed to one of the machines in a cluster is called load balancing. Load balancing techniques can be broadly classified as transparent and non-transparent. The basis of transparency is from the client perspective. Sites that do transparent switching do it in a way that the response appears to come from the same server regardless of the specific machine that handled the request. Non-transparent switching involves the use of application layer protocols to direct a client to one of the multiple server machines forming the cluster at a Web site.

A motivation for our work lies in understanding the correlation between the suggested load balancing techniques in research and the techniques in use by popular Web sites. We also study the amount of content distribution used by Web sites where more than just the origin server for a URL is used to serve the content of a page. These auxiliary servers serve particular types of content such as images or ads, or are part of content distribution networks (CDNs), which serve portions of content from servers located closer to clients [12].

One approach to study the techniques in use is to contact the administrators of popular Web sites directly and inquire about the approach used. While useful, particularly for sites employing transparent load balancing, we did not believe this approach would lead to much response. Administrators are busy and naturally reluctant about providing details of their installation. As an alternate, we use a client-based approach for studying the load balancing and content distribution techniques used by popular Web sites. The advantage of this approach is that it can be applied uniformly to any set of sites. We use information obtained from DNS (Domain Name System) queries, HTTP headers and HTML content to classify the techniques used by a particular site.

The obvious disadvantage of this approach is in identifying details of sites using transparent load balancing. However an inability to identify the details of these sites is itself a reflection on the correctness of the implementation at those sites. In cases where we can identify non-transparency for sites that do transparent load balancing, we may identify conditions that cause problems for other Web entities such as caches.

Another aim of our study is to identify problems with interactions between load balancing with features of HTTP. For example, client caches use HTTP headers such as `Last-Modified` and `Etag` to validate cache contents. In a previous study [20] it was found that the Etag for an object varied even though the object remained the same. This situation occurred because different server machines were serving the same content, but using server-specific Etag specifiers.

Another feature of HTTP/1.1 is the capability for a client and server to maintain a persistent connection so that a client can retrieve multiple objects over a single TCP connection. The provision for persistence in HTTP/1.1 is done to speed up the page retrieval as compared to HTTP/1.0, where a page is retrieved by establishing a separation connection for each object. However, as Web sites distribute content amongst multiple servers, the usefulness of persistent connections is diminished.

We studied the use of cluster load balancing and content distribution techniques by 120 popular Web sites in April 2000 and May 2002. We also carried out a more detailed study of content distribution for 51 sites with additional data from November 1999 and May 2001. This longitudinal study allows us to assess any trends in the techniques being used. It also allows us to examine the ongoing interaction between these techniques and features of HTTP.

In Section 2 of the paper we discuss research on load balancing techniques in Web clusters, along with previous work on content distribution and relevant HTTP features. In Section 3 we motivate the approach used in this work followed by the methodology we employed in Section 4. In Section 5 we present the results of our work followed by a summary of our findings and contributions as well as directions for future work in Section 6.

2 Related Work

There have been numerous research papers and products on load balancing systems for requests coming to a cluster of machines at a Web site. An excellent summary of this work is presented in [5]. The authors of [5] present a taxonomy of server architectures that include 1) a Web cluster where the set of Web server machines reside behind a switching device and only the IP (Internet Protocol) address of the device is visible to clients; 2) a Virtual Web cluster where a set of Web server machines share the same virtual IP address; and 3) a distributed Web system where the IP addresses of the Web server machines are visible to clients.

Architectures of the first type can either use content-blind (Layer 4) or content-aware (Layer 7) switching to decide how to direct client requests. Specific policies and implementations for both content-blind [7, 10] and content-aware [3, 6] switches have been explored. Architectures of the second type use MAC-layer, content-blind routing to direct requests to a particular server [8, 19]. For purposes of our work we consider both of these types of architectures as *transparent* because only one IP address is visible to clients. These architectures are advantageous because they give sites fine-grain control of directing client requests to appropriate machines.

Architectures of the third type are *non-transparent* and use other means to route requests. Three approaches described in [5], which we investigate in our work, are DNS routing, HTTP redirection and URL rewriting. In DNS routing, the authoritative DNS server for a server name selects a different IP address for each request reaching it [4]. Alternately, the DNS server can return a cacheable set of IP addresses for a server name. Because DNS mappings are cached for some amount of time, this approach provides a coarser grain of control for sites to map client requests to different server machines. The dynamics of the mapping is controlled by a time-to-live (TTL) attached to the mapping by the authoritative DNS server.

When using HTTP redirection a server responds to a request with a 302 HTTP response redirecting the client to another server [9]. When used for load balancing, this approach allows sites to select which server to use, but forces clients to make an additional request to the selected server. With URL rewriting, traversal or embedded links on a page are dynamically changed to direct subsequent client requests to an alternate server. Such an approach can be used for local routing of requests within a Web cluster, but it requires dynamic content generation in handling the request.

Rather than serve all content for a Web page from the origin server for the page, content can also be distributed amongst different servers for both load balancing and practical reasons. For example, a particular server may serve images for a site or another server may be used to serve ads that appear on a Web page. Content Distribution Networks (CDNs) have also been created consisting of a collection of servers that attempt to offload work from origin servers by delivering content on their behalf [12, 11]. The servers belonging to a CDN commonly are located at different (relative to the origin server) locations in the network, with some or all of the origin server's content cached or replicated amongst the CDN servers. Requests are commonly mapped to CDN servers using DNS redirection, but other techniques such as URL rewriting are also used [12].

Two potential issues with the distribution of content among servers in a cluster or on a wider scale is how the distribution interacts with features of HTTP. First, the HTTP headers `Last-Modified` and `Etag` can be part of the response returned by a Web server to indicate the last modified time or an entity tag for the object. These headers can be saved by a client cache and used on subsequent requests to conditionally retrieve the same object only if the object contents have changed from its previous version. If multiple servers are serving the same objects then there is potential that they serve the same content, but different header values. In fact a previous study [20] found this condition to exist for a small number of popular Web sites.

The other interaction issue occurs with the HTTP/1.1 feature where a client and server can maintain a persistent connection so that a client can retrieve multiple objects over a single TCP connection [9]. Previous studies have shown that persistent connections, particularly when used with pipelining, can provide better performance than parallel HTTP/1.0 requests [16, 14, 13]. Although pipelining and persistent connections are not always supported on all HTTP/1.1 servers, their support in Web clusters can be problematic. Some work has been done to support persistence in switch-based architectures [2]. If Web sites distribute content amongst multiple servers at a site or geographically then a separate connection must be established by a client to each server.

3 Approach

Our work is not about proposing a new load balancing or content distribution technique. Such techniques have been widely explored in the research literature and in the range of products and services available in the marketplace. This work takes a different approach by examining what techniques are being *used* and whether their use has changed over time. We seek to answer questions about the use of transparent versus non-transparent load balancing techniques as well as the degree to which distribution of Web site content is being used. We are not aware of previous work that has examined the use of load balancing techniques and are only aware of one previous piece of work that has specifically examined the prevalence of content distribution [14].

Without access to the servers at such sites or knowledge of what techniques are being used we are constrained to analyzing data that is available to us in response to HTTP requests and DNS resolutions. The constraint however puts us in the same domain in which typical web users would find themselves and allows us to study a Web server from the point of view of clients. This approach allows us to study non-transparent techniques directly and to detect when transparent load balancing is done incorrectly. The latter occurs when client-based tests can detect differences in transparently-served content.

In our work, we studied the use of load balancing techniques for clustered server machines at a Web site and for the distribution of content amongst separately named servers. The goal is to examine the prevalence of each technique as well as its interaction with HTTP. We identified the following sets of sites for how servers are used.

1. Single IP: Servers for sites in this set always return the same IP address for all DNS queries from a client. These sites either use a form of transparent switching or just use a single server to handle requests. While the latter is possible, we expect it to occur little if we focus our study on popular sites.

2. Multiple IP: These sites use DNS redirection where the server for the site is mapped to multiple IP addresses in response to DNS queries.

3. HTTP Redirection. These sites return a HTTP 302 response to redirect subsequent requests to another server. In some cases, the client is always redirected to the same server while other sites may redirect a client to a set of servers as a form of load balancing.

4. URL Rewriting. The URLs in HTML content served for these sites are modified to contain different server names so that retrieval of embedded objects or subsequent traversal of links can be load balanced amongst a set of servers.

5. Content Distribution. Sites using content distribution use servers other than the origin server for a page to serve embedded objects. We term these servers as *auxiliary* because they aid the origin server in serving the content for the page.

4 Methodology

Having identified what to study, we also needed to identify a set of Web sites to examine. Because we expect the use of Web clusters and content distribution to be most prevalent in popular Web sites, we focused on these sites by constructing a list of 120 popular sites based on ratings of www.100hot.com[17] in early 2000. While the notion of popularity does change over time, we used this set of sites for all tests for consistency. The set of sites we used are listed in Appendix A.

All tests were conducted from a client machine at WPI. We always retrieved the home page for a site. The first set of tests was conducted in April 2000 with a follow-up study in May 2002. The same set of 120 sites was used in both tests. The two-year gap between the two tests allows for examining differences in use over time. Tests were conducted with a simple HTTP retrieval client written in C with scripts written in Perl to control the testing and parse the content. For each site in the list, our script performs the following steps:

1. The script first performs a DNS resolution on the server name for the home page of the site to obtain an IP address.

2. The HTTP client retrieves the home page for the server using a HTTP/1.0 GET request. For our tests the use of persistent connections is not needed.

3. The HTTP response received as part of the reply is examined to see if HTTP redirection is being used. If a 302 code is found then the server is put on a list of servers using HTTP redirection.

4. If the server does not fall in the above category, our script examines the number of IP addresses returned by the DNS name resolver. If there is more than one IP address we put the server into the multiple IP address list. If not then the server is put in the single IP list.

After classifying each server in the list based on this initial methodology, we have three sets: the single IP set, the multiple IP set and the HTTP redirection set. The remainder of this section describes our methodology for data collection and analysis for each of these three sets of servers. We also discuss our methodology for measuring the use of URL rewriting and content distribution.

4.1 Single IP

The Single IP set constitutes the set of servers for which DNS mapped the server name to a single IP address and the HTTP request for the server home page returned a 200 (HTTP OK) response. We expect popular sites with only a single IP address visible to clients to be using some form of transparent load balancing. While we cannot test what load balancing technique is being used from a client, we designed our experiment to see whether inconsistencies in server responses could be detected. For this investigation we made multiple (five is the number we used in our tests) retrievals of the home page from the given server using our client. We store the headers and contents for each retrieval. We also compute an MD5 checksum on the contents to determine if changes occur between retrievals.

Once we have all retrievals, we analyze them by comparing the HTTP headers and MD5 checksum of the first retrieval with those of subsequent retrievals. With a transparent mechanism, all machines in the cluster should return the same set of headers if the content itself does not change. We specifically examine the `Etag` and `Last-Modified` headers (if available) between retrievals. If we find that either of these headers differs between retrievals while the checksum remains the same then the validation of this object by a client cache would result in an unnecessary retrieval of the object contents. At the minimum, a change in these headers, with the checksum remaining the same, indicates the site is using some form of transparent load balancing.

We also looked for changes in other headers such as the server software used or server-supplied date. A significant difference in the date between consecutive retrievals could indicate that the different retrievals are handled by separate server machines with their clocks not synchronized. Differences in server software are not likely to cause problems for clients, but unsynchronized clocks could affect how client caches operate.

4.2 Multiple IP

For the multiple IP set we know that these sites are using DNS-based redirection to direct client requests to different machines. Although the investigative steps are similar to the Single IP set, the tests are done on each individual IP address returned by the DNS server.

We use the HTTP client to retrieve the HTML container object for the home page from each of the IP addresses that form part of the cluster. Once the HTML object is retrieved we use similar techniques as the Single IP set to identify variations in the HTTP header responses from different server machines.

4.3 HTTP Redirection

Servers returning a HTTP 302 response in our initial test were put in a set using HTTP redirection. HTTP redirection is a potential approach to load balancing amongst a set of servers or it may simply be used by a site to always redirect requests to another URL as well as performing other actions such as returning a cookie. This approach does introduce more delay because a client must make another request to actually fetch the content.

This set is investigated by first sending a request to the server for the home page, which responds with a 302 status and a new URL specified in the HTTP `Location` header. We continue to make requests to the redirected server location until the server returns a 200 status. We repeat this process of following redirections until a 200 response is returned a total of five times to see if a request is always redirected to the same server or if different requests are redirected to different servers. The focus of our analysis for this set of servers is to detect if it is used for load balancing. We did not investigate potential differences between HTTP headers for responses in this set.

4.4 URL Rewriting

We also used the results for each of the sites in the first two test sets to see if we could detect the use of URL rewriting. Because we used the results of the first retrieval, we did not use results from the sites in the HTTP redirection set. With URL rewriting, the server names of traversal links and embedded objects in an HTML page are modified by the server to direct subsequent client requests to different servers. URL rewriting can also be used for content distribution as described in [12].

We tested for URL rewriting for each site by first identifying all servers referenced in a traversal or embedded link in the initial home page retrieval. We compute this list of servers for subsequent retrievals and compare the initial list with subsequent lists. If there is a difference then we investigated further to see if it is a case of load balancing or just that the contents of the page have changed.

4.5 Content Distribution

In addition to checking for URL rewriting in the contents of the first two test sets, we also used the results to quantify the amount of distribution used for embedded objects on these pages. These objects are retrieved automatically by most browsers when a page is requested.

In doing the analysis, we determined both the number of embedded objects on the page as well as the number of these objects that are served by servers other than the origin server. We also determined the number of auxiliary servers serving these objects. Content distribution is interesting because it potentially allows a client to retrieve embedded content from a server closer to the client, but it also interacts negatively with persistent connections because a client must establish a TCP connection with a different server.

After obtaining content distribution results for April 2000 and May 2002, we correlated these results with data from a study done in November 1999 as part of work described in [14]. The authors in [14] found that approximately 15% of 665 servers used auxiliary servers to serve embedded content. They further classified an auxiliary server as a *local* server if the network portion of the IP address for it matched the network portion of the IP address for the origin server, They classified a server as *ad* if the string "ad" was found in the server name. The category *akamai* was used for servers if the string "akamai" was found in the server name indicating it belonged to Akamai [1], a commercial content distributed network. Unclassified servers were marked as *other*.

Following up on this work, we identified the subset of 51 of our 120 server sites that were also studied in the November 1999 study. We also obtained data for these 51 server sites from a similar, unpublished study carried out in May 2001. The subset of 51 sites are identified in Appendix A.

Using the classification of [14] as a guideline we classified auxiliary servers into four sets. We defined a server as *local* if the name of the server contained the same company name or if the authoritative DNS server for the auxiliary server matched the origin server. We identified *ad* servers as containing the string "ad" or being in a list of servers known to serve ads [15]. We defined *CDN* servers as coming from Akamai [1] or Speedera [18]. Unclassified servers are marked as *other*.

This portion of our work allows us to both understand changes in the amount of content distribution over a period of a few years and also to examine the extent to which CDNs are being used by the subset of popular Web sites.

5 Results

Table 1 shows the number and percentage of sites classified into each of the initial three classification sets. In the April 2000 study, two-thirds of the sites returned a single IP address for the site server. The number went down a bit in May 2002. The number of sites returning multiple IP addresses for the site server remained constant between the two studies at just under 20% of all sites. The number of sites using HTTP redirection went up from 15% in April 2000 to almost 20% in May 2002.

To better understand the consistency in approach used over time, Table 2 shows the transitions in server classification between the two runs. By adding up the diagonal in the table we find that 89 out of the 120 (74%) sites remained in the same set between the two runs. The table shows there are at least two changes between all classification set pairs with the biggest shift between the two runs being the 9 sites that used a single IP address in April 2000, but used HTTP redirection in May 2002.

In the remainder of this section, we examine each of these three sets in more detail as

Table 1: Classification of Web Sites

	Number of Sites (Pct. of Total)	
Web Site Set	**April 2000**	**May 2002**
Single IP	80 (67)	75 (63)
Multiple IP	22 (18)	22 (18)
HTTP Redirection	18 (15)	23 (19)

Table 2: Change in Web Classification Between the Two Test Runs

Test Run	**May 2002**				
	Classified Set	Single IP	Multiple IP	HTTP Redir.	Total
April 2000	Single IP	64	7	9	80
	Multiple IP	7	13	2	22
	HTTP Redir.	4	2	12	18
	Total	75	22	23	120

well as examine the use of URL rewriting and content distribution.

5.1 Single IP

As indicated in Section 4, for each of the sites in the Single IP set we investigated the response headers for variations. Table 3 shows the results of this investigation. The table shows that for three of the sites in April 2000 and two of the sites in May 2002, we found evidence that the `Etag` header changed, but the content did not. We found a similar case with the `Last-Modified` header for one site in May 2002. These cases not only indicate that more than one server machine is serving the same content, but that a client caching this content will unnecessarily retrieve the same content again if it tries to validate the cached content. We also noticed an increase in the number of Web sites modifying content on each request in the May 2002 results so changes in these headers are expected.

We did not see any significant date change between any responses, but as shown in the table, we did find a few sites that returned different values for the server software header in response to repeated requests. Table 3 also shows two sites in May 2002 with variation in the `Content-Location` header, indicating multiple servers are being used. The content-location is described by RFC 2616 [9] as below.

"The Content-Location value is not a replacement for the original requested URI; it is only a statement of the location of the resource corresponding to this particular entity at the time of the request. Future requests MAY use the Content-Location URI if the desire is to identify the source of that particular entity."

Overall, the results in Table 3 show that 5-6% of sites classified as Single IP exhibit some amount of inconsistency in at least one returned HTTP header (some sites exhibit variation in more than one header). While this percentage is relatively small, it represents sites where the load balancing is not transparent. In a few cases, these inconsistencies can lead to poor interaction of client caches where cached content that is still fresh will be discarded when the client validates it with the server.

Table 3: Header Variation with Same Content for Single IP Set

	Number of Sites (Pct. Single IP)	
Header	**April 2000**	**May 2002**
Etag	3 (4)	2 (3)
Last-Modified	0 (0)	1 (1)
Server	1 (1)	3 (4)
Content-Location	0 (0)	2 (3)
Any Header	4 (5)	5 (6)

5.2 Multiple IP

Table 4 shows that in April 2000 over 30% of the sites using multiple IP addresses for server load balancing exhibited some form of variation in header values, although the percentage drops a lot for the May 2002 results. Sites in the April 2000 results had problems with changing `Etag` headers for the same content, but all of these sites fixed this problem for the later results. We also found changes in the `Content-Location` header for different requests, but these changes corresponded to the IP address we were requesting the content from and hence were not of interest.

Table 4: Header Variation with Same Content for Multiple IP Set

	Number of Sites (Pct. Multiple IP)	
Header	**April 2000**	**May 2002**
Etag	4 (18)	0 (0)
Server	3 (14)	1 (5)
Any Header	7 (32)	1 (5)

5.3 HTTP Redirection

Table 1 shows that the number of sites using HTTP redirection increased for the May 2002 results, but there was not a corresponding increase in the number of sites using this technique for load balancing. In April 2000, we found four (22%) of the sites using HTTP redirection to redirect client requests to multiple servers. In May 2002, we found three (13%) such sites. In the latter results we did detect another site always redirecting to the same server, but using a site-defined header to indicate multiple server names.

5.4 URL Rewriting

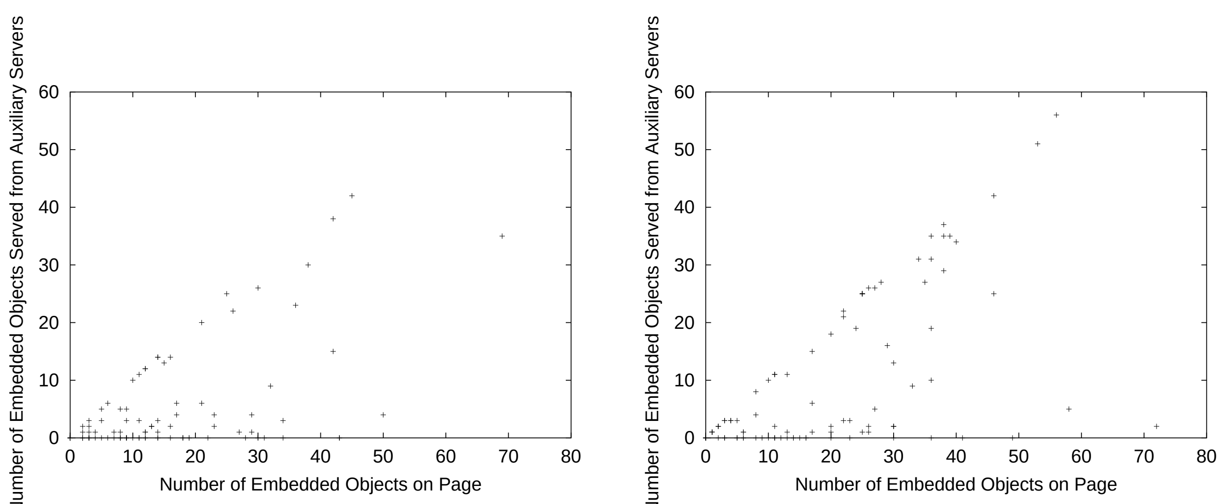
We examined the set of sites in the Single and Multiple IP address sets for evidence of URL rewriting and found none. It is possible that sites in our list employ URL rewriting, but we found no evidence of it simply because all tests were from the same client site. Future work should look at using a similar methodology from more than one client location.

5.5 Content Distribution

The last portion of our study examined the prevalence of using multiple servers to serve the content on the home pages of sites in our list. We parsed the contents of the HTML content for these pages to determine the number of embedded objects on these pages. JavaScript code embedded on these pages is not executed so objects needing to be retrieved because of its execution are missed. We also determined the number of these objects served by auxiliary servers and the total number of these auxiliary servers used. Median and mean results for these characteristics are given in Table 5. Scatter plots showing the number of embedded objects on the home page of a site relative to the number served by auxiliary servers are shown in Figures 1a and 1b.

Table 5: Median/Mean for Embedded Object Distribution

Characteristic	**April 2000**	**May 2002**
Number of Embedded Objects	12/14.8	17/20.0
Number of Embedded Objects from Auxiliary Server	1/4.7	2/9.8
Number of Auxiliary Servers Used	1/1.0	1/1.4



a. April 2000 b. May 2002

Figure 1: Relative Distribution of Embedded Objects on Web Site Home Pages

The results show both a clear increase in the number of embedded objects on these pages over a two-year period and an even bigger increase in the number of these objects being served by auxiliary servers. Data points on the x-axis in Figures 1a and 1b indicate sites where all objects are served by the origin server while data points on the diagonal indicate all embedded objects are served by one or more auxiliary servers. As another indication of the increase in auxiliary server use, Figure 2 shows the cumulative distribution for the number of auxiliary servers used by a site. As shown, the number of sites using an auxiliary server increased from 50% in April 2000 to 69% in May 2002. While this increase in distribution potentially allows clients to retrieve objects in parallel, it also reduces the potential benefit of persistent connections because the client must establish connections to more server machines.

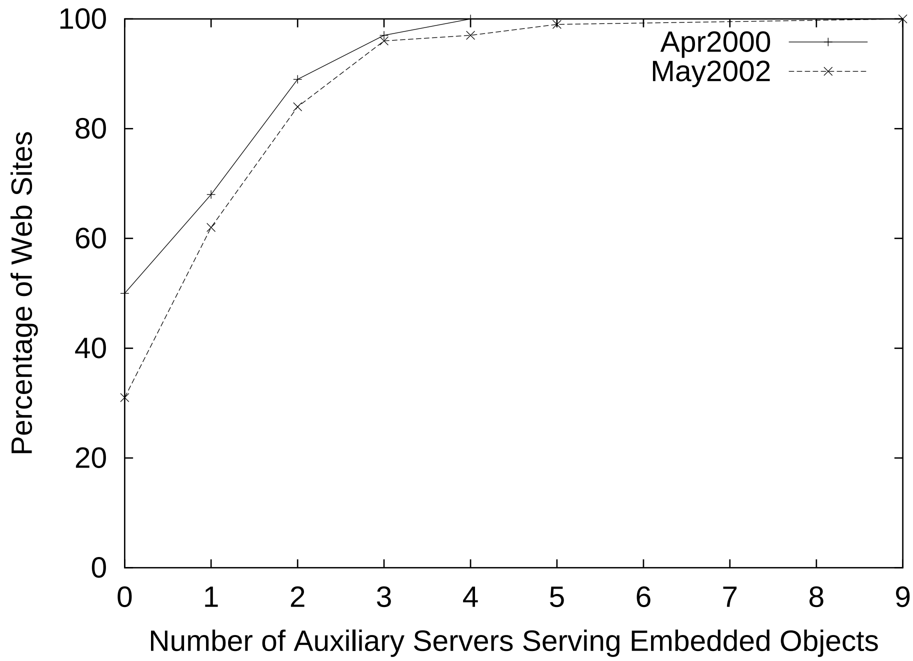


Figure 2: Cumulative Distribution for Number of Auxiliary Servers Serving Embedded Objects for a Site Home Page

Not only were we interested in the amount of auxiliary server use, but also the role of these auxiliary servers. We know that some sites use other servers local to the site to serve particular types of objects such as images. Other servers are used for embedded ad objects. We are also interested in the number of sites using identifiable CDN servers to serve a portion of the embedded content. In analyzing our results for this type of classification we used the opportunity to combine our collected data with data obtained in November 1999 as part of [14] and similar data obtained in May 2001. 51 sites overlap between these two data sets and our study. These 51 sites are identified in Appendix A.

Using results from these 51 sites over four data collection periods we first determined the percentage of these sites using auxiliary servers to serve embedded objects on the site home page. These results are shown for the first data line in Figure 3. The results show a steady increase from 51% of sites in November 1999 to 73% in May 2002. Note the 73% figure for the 51 sites compares to a 69% figure for all 120 sites in May 2002. Similarly, a 57% figure for the 51 sites compares to a 50% figure for all 120 sites in April 2000.

The remaining data lines in Figure 3 show results for different types of auxiliary servers. The number of sites employing local auxiliary servers has also shown a steady increase from

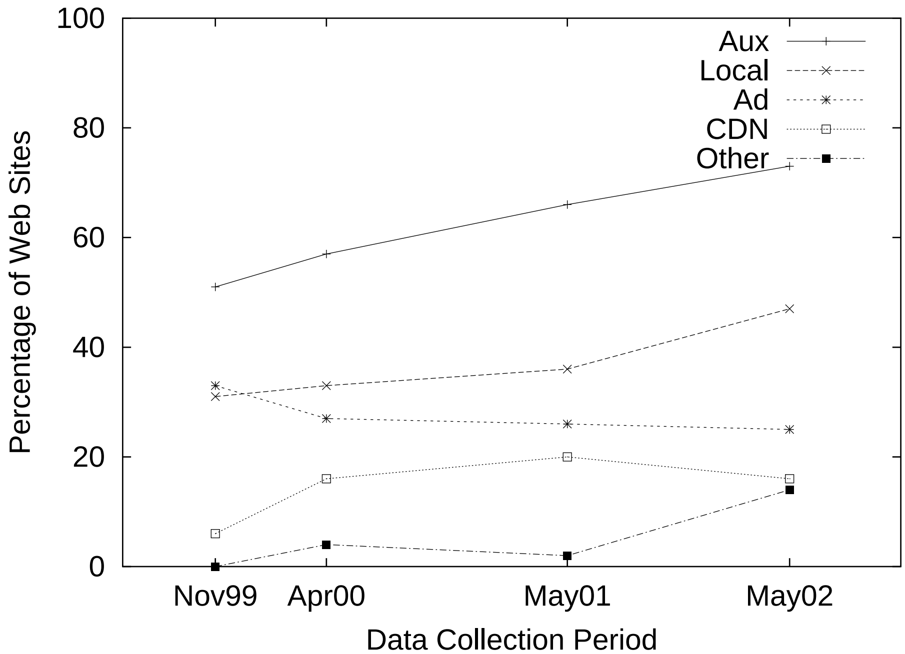


Figure 3: Percentage of Sites Using Auxiliary Servers to Serve Embedded Objects

31% to 47% of sites. The use of identifiable ad servers has dropped slightly to 25% of sites in May 2002. Also of interest is the percentage of servers we could identify as coming from known CDN servers. This percentage rose from 8% in November 1999 to 16% in April 2000 and then peaked at 20% in May 2001. However, in May 2002 the percentage dropped back to 16% with some new sites using CDN servers for the first time, but other sites no longer employing them.

6 Summary and Future Work

Much research work on load balancing and content distribution techniques has been published with many offerings available in the marketplace that provide these services. However, little work has been published on what techniques are being used. A key contribution of our work is a longitudinal study on the use of load balancing techniques at a set of popular Web sites. This study was done over a two-year period using a client-based methodology that while limited in characterizing transparent load balancing approaches, can be applied to any site.

Our results show that over 60% of sites continue to map client requests to a single IP address indicating that either transparent load balancing is being used or that the sites use only a single server to handle requests. We believe the latter is less likely given that these are among the most popular Web sites. Despite the apparent use of transparent techniques we found that 5-6% of these sites show some amount of HTTP header inconsistency when comparing the results of multiple client retrievals. While this percentage is relatively small, it represents sites where the load balancing is not transparent. In a few cases, these inconsistencies can lead to poor interaction with client caches where cached content that is still fresh will be discarded when the client validates it with the server.

In both our past and most recent study, we found 18% of sites use DNS redirection to map a client request to different machines. We again found some inconsistencies among headers returned from different machines, but the number of inconsistencies was almost zero in the May 2002 study. We found 15-20% of sites using HTTP redirection in the two studies with only three or four of these sites actually using HTTP redirection to redirect a client to a different server. We found no evidence of URL rewriting being used amongst the set of sites.

Another contribution of our work is a study on the extent of content distribution from the origin server of a site to auxiliary servers local and outside of the site. We found a clear increase in the number of embedded objects on these pages over a two-year period and an even bigger increase in the number of these objects being served by auxiliary servers. 69% of the 120 test sites used auxiliary servers in May 2002 compared to 50% in April 2000. In a more detailed study of 51 sites, using additional data, we found 51% of these sites used auxiliary servers in November 1999 compared to 73% in May 2002. Looking at specific types of auxiliary servers we see the number of sites using other servers local to the site increase from 31% to 47% over the same time period. The use of known CDN servers increased from 8% in November 1999 to 20% in May 2001, but dropped back to 16% of the 51 sites in May 2002. While this use of content distribution potentially moves content closer to clients and increases opportunities for parallel retrieval of embedded objects, it also reduces the potential benefit of persistent connections because the client must establish connections to more server machines.

There are a number of directions for future work on this project. One obvious direction of such a longitudinal study is to continue it to understand if the use of load balancing techniques or extent of content distribution changes. It would also be useful to conduct the study from more than one client site so that sites that use client-specific redirection or URL rewriting could be detected.

As a means to confirm our methodology it would be beneficial to follow up with site administrators to confirm the load balancing techniques used by these sites—particularly for cases of transparent load balancing. Another direction is to explore combinations of load balancing techniques. For example, in our study we did detect that some sites using HTTP redirection are also using multiple IP addresses for the servers doing the redirection. Sites using DNS redirection may be redirecting clients to a cluster rather than a single machine.

References

[1] Akamai. `http://www.akamai.com/`.

[2] Mohit Aron, Peter Druschel, and Willy Zwaenepoel. Efficient support for P-HTTP in cluster-based web servers. In *Proceedings of the 1999 USENIX Annual Technical Conference*, Monterrey, CA USA, June 1999. USENIX Association.

[3] Mohit Aron, Peter Druschel, and Willy Zwaenepoel. Cluster reserves: A mechanism for resource management in cluster-based network servers. In *Proceedings of the ACM SIGMETRICS 2000 Conference*, Santa Clara, California, June 2000. ACM.

[4] T. Brisco. DNS support for load balancing, 1995. RFC 1794.

[5] Valeria Cardellini, Emiliano Casalicchio, Michele Colajanni, and Philip S. Yu. The state of the art in locally distributed web-server systems. *ACM Computing Surveys*, 34(2):263–311, June 2002.

[6] Emiliano Casalicchio, Valeria Cardellini, and Michele Colajanni. Content-aware dispatching algorithms for cluster-based web servers. *Cluster Computing*, 5(1), October 2002.

[7] Michele Colajanni, Philip S. Yu, and Daniel M. Dias. Analysis of task assignment policies in scalable distributed web-server systems. *IEEE Transactions on Parallel and Distributed Systems*, 9(6):585–600, June 1998.

[8] Om P. Damani, P. Emerald Chung, Yennun Huang, Chandra Kintala, and Yi-Min Wang. ONE-IP: techniques for hosting a service on a cluster of machines. *Computer Networks and ISDN Systems*, 29(8–13), 1997.

[9] R. Fielding, J. Gettys, J.C. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol – HTTP/1.1, June 1999. RFC 2616, IETF, Draft Standard.

[10] Guerney D.H. Hunt, G. S. Goldszmidt, Richard P. King, and Rajat Mukherjee. Network dispatcher: a connection router for scalable internet services. *Computer Networks and ISDN Systems*, 30(1–7):347–357, April 1998.

[11] Kirk L. Johnson, John F. Carr, Mark S. Day, and M. Frans Kaashoek. The measured performance of content distribution networks. In *Proceedings of the 5th International Web Caching Workshop and Content Delivery Workshop*, Lisbon, Portugal, May 2000.

[12] Balachander Krishnamurthy, Craig Wills, and Yin Zhang. On the use and performance of content distribution networks. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, November 2001.

[13] Balachander Krishnamurthy, Craig Wills, and Yin Zhang. Preliminary measurements on the effect of server adaptation for web content delivery. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, November 2002.

[14] Balachander Krishnamurthy and Craig E. Wills. Analyzing factors that influence end-to-end web performance. In *Proceedings of the 9th International World Wide Web Conference*, Amsterdam, Netherlands, May 2000.

[15] James Moberg. How to block ads (&web bugs) without extra software, 2002. SunStar Media. `http://ssmedia.com/Utilities/hosts/`.

[16] Henrik Frystyk Nielsen, Jim Gettys, Anselm Baird-Smith, Eric Prud'hommeaux, Hikon Lie, and Chris Lilley. Network performance effects of HTTP/1.1, CSS1, and PNG. In *Proceedings of the ACM SIGCOMM '97 Conference*. ACM, September 1997.

[17] 100hot.com.

[18] Speedera. `http://www.speedera.com/`.

[19] Sujit Vaidya and Kenneth J. Christensen. A single system image server cluster using duplicated MAC and IP addresses. In *Proceedings of the IEEE Conference on Local Computer Networks*, Tampa, FL USA, November 2001.

[20] Craig E. Wills and Mikhail Mikhailov. Towards a better understanding of web resources and server responses for improved caching. In *Proceedings of the Eighth International World Wide Web Conference*, May 1999.

A Web Site List

The following is the list of 120 Web sites used in this study. The list of 51 sites used for the study of content distribution described in Section 4.5 are marked with a †.

home.netscape.com†	java.sun.com†	www.100free.com	www.about.com
www.altavista.com†	www.amazon.com†	www.aol.com†	www.apple.com†
www.bbc.co.uk	www.berkeley.edu	www.beseen.com†	www.bloomberg.com†
www.bluemountain.com†	www.burstmedia.com	www.cdworld.com	www.chek.com
www.cisco.com†	www.city.net	www.cjb.net	www.classifieds2000.com
www.cnet.com†	www.cnn.com†	www.collegeclub.com	www.compaq.com†
www.dai.net	www.dash.com	www.datais.com	www.dell.com†
www.developer.com†	www.download.com	www.ebay.com†	www.ecircles.com
www.entrypoint.com	www.etrade.com	www.eudoramail.com	www.excite.com†
www.fbcgi.com	www.fidelity.com	www.fool.com†	www.fortunecity.com†
www.four11.com	www.freethemes.com†	www.freevote.com	www.game-revolution.com
www.gamedemo.com	www.gamepen.com†	www.gamespot.com†	www.go.com†
www.go2net.com	www.headbone.com	www.homestead.com	www.hotfiles.com
www.hotmail.com	www.hp.com†	www.ibm.com	www.infosel.com
www.infospace.com†	www.internet.com†	www.ircache.net	www.linkexchange.com†
www.looksmart.com	www.lycos.com	www.macromedia.com	www.mail.com
www.mapquest.com	www.mckinley.com	www.microsoft.com†	www.monster.com
www.mp3.com†	www.mplayer.com	www.msn.com	www.msnbc.com†
www.mtv.com†	www.nai.com†	www.nasa.gov†	www.nasdaq.com
www.nettaxi.com†	www.news.com	www.nih.gov	www.noaa.gov
www.onelist.com	www.passport.com	www.pathfinder.com†	www.quote.com
www.real.com	www.realtor.com	www.recycler.com	www.scea.sony.com
www.schwab.com†	www.search.com	www.shareplay.com	www.simplenet.com
www.snap.com†	www.snowball.com	www.sony.com†	www.spedia.net
www.sportsline.com†	www.station.sony.com	www.sun.com†	www.talkcity.com
www.tandem.com†	www.theglobe.com	www.tminterzines.com†	www.tucows.com
www.ugo.net†	www.uolmail.com	www.uproar.com	www.usa.net
www.usatoday.com†	www.weather.com†	www.web1000.com	www.webcrawler.com†
www.webjump.com	www.whowhere.com	www.winamp.com†	www.women.com
www.wwf.com†	www.xoom.com†	www.yahoo.com†	www.zdnet.com†