WPI-CS-TR-00-12 July 2000

The Contribution of DNS Lookup Costs to Web Object Retrieval

by

Craig E. Wills
Hao Shang

Computer Science
Technical Report
Series

WORCESTER POLYTECHNIC INSTITUTE

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

Abstract

There are a number of factors that contribute to the performance between clients and servers in the World Wide Web. In this work we have concentrated on the contribution of DNS lookup to the overall Web object retrieval time. We found that the DNS mechanism performed better for popular Web servers than for random Web servers. Performance was better both in terms of local cache hit rates, which would be expected, but also for authoritative and non-local, non-authoritative response times.

We also found that the DNS lookup time contributed more than one second to approximately 20% of retrievals for the Web objects on the home page of a larger list of popular servers. While these numbers are significant, they do not reflect the expected access patterns for URLs. Results from real user request log show that only about 20% of DNS requests are not cached locally. If 20-30% of these non-cached lookups take more than one second then only 4-6% of all DNS lookups should take more than one second. These numbers appear to be less of a problem for overall Web retrieval, particularly if one considers that many DNS lookups resulting from small DNS TTL values do not yield different IP addresses for successive requests.

Keywords: Web Performance, Domain Name System

1 Introduction

There are a number of factors that contribute to the performance between clients and servers in the World Wide Web. In previous work we have studied the relative influence of some of these factors and their interaction with each other [5]. In this work we focus on a factor not considered in our prior work—the influence of Domain Name System (DNS) lookup costs.

The Domain Name System performs a number of tasks, most notably it is used to map host names to Internet (IP) addresses [6, 7]. As part of a URL, users specify a server from which to retrieve an object. The software agent retrieving the object, either a browser, or a proxy cache working on behalf of a browser, first uses DNS to resolve the server name to its Internet address.

Little published work has been done in considering the cost of this DNS resolution in the end-to-end cost to retrieve Web objects from a server. The most notable piece of work is a study by Cohen and Kaplan on the potential performance improvement due to browsers or proxies pre-resolving a server name [1]. This technique is studied in conjunction with other work that can be done prior to a request to reduce the overall delay of an HTTP request.

Cohen and Kaplan have a related paper focusing on the impact of pro-active renewal policies for DNS cache entries [2]. They also investigate the effect of simultaneously retrieving Web content from servers with cached, but stale DNS entries, in expectation that validation of the DNS entry will show that the server to Internet address mapping has not changed.

Researchers at Telcordia Technologies have developed a tool called *webtest*, to study the delay for four components of Web retrieval: DNS delay, connection delay, server delay and transmission delay [3]. The test measures these delays for a set of random URLs with results from another Web site also showing results for popular URLs [4]. Of interest to our work is the relatively high number of servers that yielded a DNS lookup time of over one second. In our own tests with webtest here at WPI we found that 29% of random servers tested yielded a DNS lookup time of over two seconds.

These prior pieces of work and our own interest in the interaction of factors contributing to end-to-end Web costs motivate this study to better understand the following issues:

- the effect of the time-to-live (TTL) value for cached DNS entries. Caching is a key component of the DNS resolution mechanism and to our knowledge only [2] has studied its effect on DNS lookup times. There are a number of issues including the expected cache hit rate for a set of requests, the number of stale hits due to expired DNS entry TTLs and the frequency at which the DNS mappings change for mappings that result in a single as well as multiple IP addresses.

- if the performance of the DNS mechanism differs for random versus popular Web servers. We are interested in not only DNS cache hit rates and associated TTL values, but also the cost to lookup non-cached mappings.

- the impact of DNS performance on end-to-end performance. In the big picture the importance of DNS costs is relative to the many other factors that contribute to Web retrieval costs.

The organization of this report includes a brief description on the DNS look-up mechanism followed by the list of questions studied in this work and the methodology used to

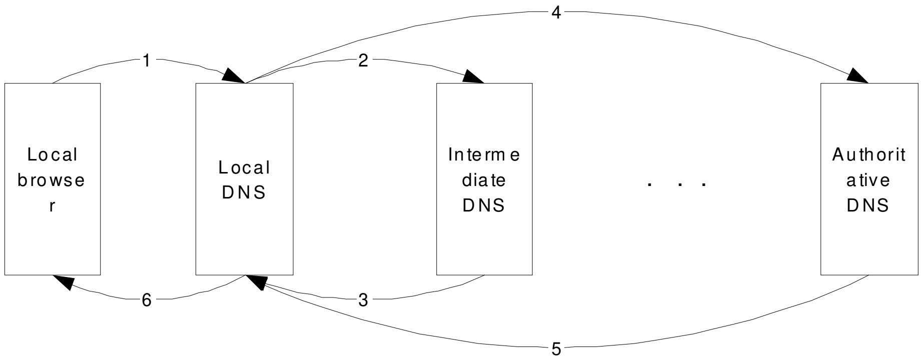

Figure 1: Basic DNS Lookup Mechanism

study them. It continues with presentation and discussion of the results and concludes with a summary of the findings along with directions for future work.

2 The Domain Name System Look-up Mechanism

The primary goal of DNS is a consistent name space for referring to resources. For Web applications, DNS is used look up the IP addresses of specified Web servers.

Figure 1 shows the basic DNS lookup mechanism where the local DNS service is invoked in a recursive manner. First, the local browser uses domain name resolving APIs (provided by the operating system platform) to send a DNS query to the local DNS. The local DNS either gives an answer from its own cache immediately or does iterative queries to get an answer from intermediate DNSs or eventually from the authoritative DNS before returning it to the browser.

The local DNS server normally saves the RRs (resource records) in its cache each time it gets answers from other DNSs. For each RR, there is a TTL field which is set by the authoritative DNS for this RR. The value of TTL is the duration that the RR can be kept in the cache.

3 Study

Motivated by the issues outlined in the introduction, our study focus on the following questions related to DNS performance.

1. What is the percentage of DNS requests than can be satisfied by locally cached DNS entries? How many of these requests are for entries, which have been previously seen, but are not used because of an expired TTL value? These questions directly examine the cost savings of caching DNS entries and the effect that stale entries have on costs.

2. How much does the use of a minimum TTL value by user agents such as Web browsers affect DNS performance? Some Web user agents use a minimum TTL, such as 15 minutes in the Mozilla browser [9]. This technique ignores small TTLs assigned by

authoritative servers thus reducing the number of DNS lookups at the risk of mapping to a less than optimal server.

3. How much does the mapping between a server name and its IP address change? Does it make a difference whether the server name maps to one or more than one IP addresses? If there are many cases where the DNS mapping does not change and the TTL is small then the DNS mechanism is causing unneeded DNS requests to be generated.

4. What is the range of TTL values assigned to DNS mappings by authoritative servers? Long TTLs allow DNS entries to be cached and improve the effective performance of the DNS mechanism.

5. What is the percentage of requests to popular servers amongst a log of user requests? We expect popular servers to have a large influence on this set of requests.

6. What is the range of TTL values assigned to DNS mappings for popular Web servers? How does this range compare to the range for random servers?

7. What is the performance of the DNS lookup mechanism for popular Web servers compared with random Web servers. Performance is measured both in local cache hit rates and costs to lookup non-local mappings.

8. What is the performance of the DNS mechanism for single IP mappings compared to multiple IP mappings? What is the performance of the mechanism for mappings with short TTLs compared with long TTLs?

9. What is the impact of DNS costs on end-to-end Web retrieval costs for a Web server? The most important aspect of the cost to map a server name to its IP address is the relative impact of this cost on the overall time to retrieve the set of Web objects from the server.

4 Methodology

We used a three part study to examine the previous set of questions.

4.1 Caching Policy Simulation

The first study involved analysis of a various proxy logs from NLANR [10]. These logs are for requests made by caches in the NLANR cache hierarchy. Basic information about these logs is shown in Table 1.

We used these logs to study the effects of DNS caching for a set of user-generated requests. We call this study the Caching Policy Simulation (CPS) as it simulates the DNS requests associated with a set of object retrievals. The simulation maintains its own DNS cache and retains the relative timing of NLANR log entries in replaying DNS lookup costs. Thus the replay of a one day log took one day. We did not try to match the time-of-day for replay of requests.

Table 1: NLANR Log Basic Information

Log	Entries	Server	Date
ac99	523414	Access	09/20/1999
bo00	770598	Boulder, Co (bo2)	02/17/2000
uc00	827962	Urbana-Champaign, Il	03/07/2000

Whenever a simulated cache miss is found then our simulator uses the Unix DNS resolver routines to access the real DNS mechanism and retrieve the DNS mappings and authoritative TTL value. The resolver routines were configured to work through the local site (WPI) DNS cache, but if a non-authoritative (i.e. cached) entry was returned then an explicit request was made to the authoritative server for this entry. Note that we did *not* measure the time to make these retrievals in our study because our approach does not mimic behavior of the *gethostbyname()* call typically used for DNS lookups. Rather we wanted to force retrieval of an authoritative response so our simulation had the authoritative TTL. Unknown servers were dropped from the study when replaying the log.

4.2 Response Time Study

The second study was designed to study the performance of the DNS mechanism from the WPI site. For this study we used the DNS resolver routines to mimic behavior of *gethostbyname()* so that all requests were sent to the campus DNS server. This approach not only allowed us to time the DNS mechanism, but also determine whether the DNS response was authoritative (A) or non-authoritative (NA) along with the number of IP addresses mapped to a given server name.

We used this approach to study the performance of popular versus random servers. A list of popular (or "hot") servers was obtained from `100hot.com` [11]. A list of random servers was obtained by randomly sampling the servers referenced in the `ac99` NLANR log.

We sampled each of the popular and random server sets over one day. Both on a weekday and a weekend. Each hour we issued DNS queries for the popular sites and 100 random sites. The set of random sites varied each hour. We separated the responses into three categories: locally cached non-authoritative responses, non-locally cached non-authoritative responses and authoritative responses. We used a cut-off value of 10 milliseconds to distinguish locally cached DNS entries (typical lookup times were 2-4ms) from intermediate DNS cache entries.

4.3 End-to-End Performance Study

Our third study was to insert the DNS lookup code from the previous study into the *httperf* Web server performance tool [8]. Httperf was designed to retrieve one or more objects from a Web server. It normally uses *gethostbyname()* to map a server name to an IP address. We followed the methodology used in [5] to retrieve the set of objects from a Web server after using our DNS resolver code to map the server name to an IP address.

5 Results

5.1 Impact of DNS Caching

We first used results from replaying the logs to determine DNS cache hit rates under different scenarios. In the first scenario we assumed that each object is retrieved via a separate network connection and a DNS lookup would be needed. Table 2 shows the results for this scenario when the authoritative DNS TTL is strictly followed. The results show that 87-94% of all lookups are resolved by a local DNS cache with 3-8% of the requests for servers previously seen, but whose DNS mapping is no longer valid. The latter portion of Table 2 shows the same results if we assume that a minimum DNS TTL of 15 minutes is always used. This assumption increases the number of DNS cache hits by 1-3%.

Table 2: DNS Cache Hit Rates (per Object) for Authoritative and Minimum TTL

		Authoritative TTL			Minimum TTL		
Log	Entries	Miss	Hit-Fresh	Hit-Stale	Miss	Hit-Fresh	Hit-Stale
ac99	523414	0.027	0.946	0.027	0.027	0.957	0.016
bo00	770598	0.048	0.870	0.082	0.048	0.902	0.050
uc00	827962	0.036	0.906	0.058	0.036	0.929	0.035

As an alternate approach for measuring DNS cache hit rates, we also replayed the log under the assumption that all object requests to the same server within a 15 second window were likely for objects on the same Web page. With the availability of persistent connections in HTTP/1.1, these objects could be retrieved on the same network connection and therefore require only a DNS lookup for the first request within the time window. Table 3 shows cache hit results under this assumption when the authoritative TTL and a minimum TTL are used. This assumption results in many fewer DNS lookups with a 71-78% hit rate for the authoritative TTL and a 77-82% hit rate for the minimum TTL. The number of stale hits (requests for previously seen, but stale DNS entries) increases to 10-18% and 7-12% for the respective TTL approach.

Table 3: DNS Cache Hit Rates (per Connection/Page) for Authoritative and Minimum TTL

		Authoritative TTL			Minimum TTL		
Log	Entries	Miss	Hit-Fresh	Hit-Stale	Miss	Hit-Fresh	Hit-Stale
ac99	126146	0.112	0.784	0.104	0.112	0.822	0.066
bo00	332239	0.111	0.708	0.181	0.111	0.773	0.116
uc00	279651	0.107	0.731	0.163	0.107	0.790	0.104

5.2 Variability of DNS Mappings

Because a non-trivial number of per connection DNS lookups resulted in stale hits, our next step was to investigate the variability of the DNS mappings between successive lookups. We wanted to determine if DNS mappings with a relatively small TTL really were changing between successive lookups or if the DNS TTL was perhaps smaller than needed. Table 4 shows the results for the authoritative and minimum TTL.

Table 4: Repeat DNS Mapping Change Percentage

Log	Auth. TTL	Min. TTL
ac99	20.0%	12.5%
bo00	19.5%	10.2%
uc00	22.2%	11.5%

The results show that a minority of repeat DNS lookups actually return different values between successive retrievals. The larger percentages for the authoritative TTL compared to minimum TTL also indicates that the mappings for servers with DNS TTLs of under 15 minutes do change and that the minimum TTL approach is reusing some DNS mappings that do change when the authoritative TTL is used.

As a further test to see if the DNS mapping for a server ever changed during the replay of the logs, we analyzed the results on a per-server basis. If the DNS mapping for a server name ever differed during the log replay then that server was marked as one for which the mapping did change. Table 5 shows the results of this analysis.

Table 5: Repeat DNS Mapping Change Percentage on a Per-Server Basis

Log	Servers	Auth. TTL	Min. TTL
ac99	14298	10.1%	10.0%
bo00	37373	6.1%	5.9%
uc00	29640	7.0%	7.0%

Table 5 shows similar results as Table 4 in that the DNS mappings for most servers do not change even though the DNS TTL has expired for a mapping. These results indicate that DNS TTLs for some servers are set too low resulting in unnecessary authoritative DNS lookups.

5.3 Multiple Versus Single IP Address Mapping

We also examined whether a single IP address was returned for a DNS lookup or whether multiple IP addresses were returned. The latter is one approach for doing simple load balancing amongst a set of server machines. In doing the analysis we looked at the TTLs returned for each set of servers and the percentage of time that the DNS mappings changed.

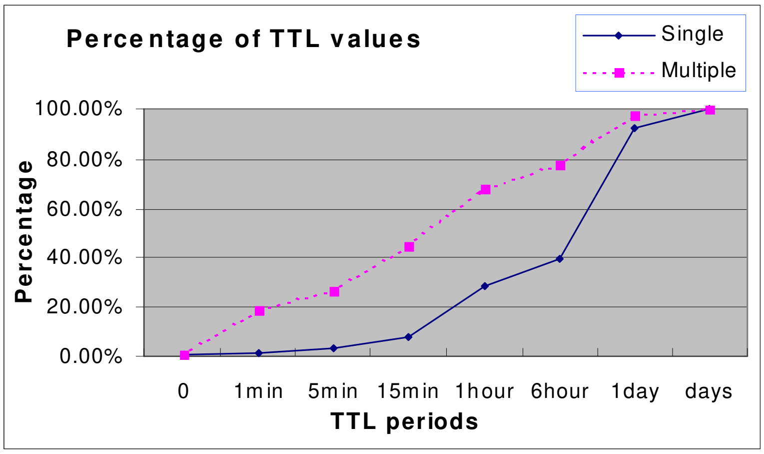

Figure 2: Range of DNS TTL values for Single and Multiple IP Addresses

Figure 2 shows a cumulative distribution graph plotting the authoritative TTL for servers in each of the two sets. In the graph, TTL values are grouped into eight groups—a group for TTLs of zero seconds, a group for TTLs from one second to a minute, for TTLs from over a minute to five minutes, from over five minutes to 15 minutes and continuing until the last group includes all TTLs over one day. As the results show, the multiple IP address sites have smaller TTLs. Our conjecture is that these sites are normally more popular sites and hence are more likely to change their DNS mappings even though they are already using multiple IP names for load balancing.

We also looked at the percentage of these DNS mapping that change. In doing this analysis we always sorted the list of multiple IP addresses before doing the comparison so that if two sets of addresses were the same, but in different orders then we would mark them as equivalent. Table 6 shows these results with the multiple IP mappings changing much more frequently.

Table 6: DNS Mapping Change Percentage on a Per-Server Basis for Multiple vs. Single IP

Log	Multiple IP	Single IP
ac99	25.8%	4.0%
bo00	21.0%	3.1%
uc00	23.5%	3.6%

5.4 Influence of Popular Servers

The next part of our study began to look at popular (or "hot") servers. We first used the set of 126 servers returned from `100hot.com` to determine the relative contribution of these servers to the requests in our logs. Because many of these popular Web sites use additional servers to serve images, ads and distributed content we also retrieved the home page of each

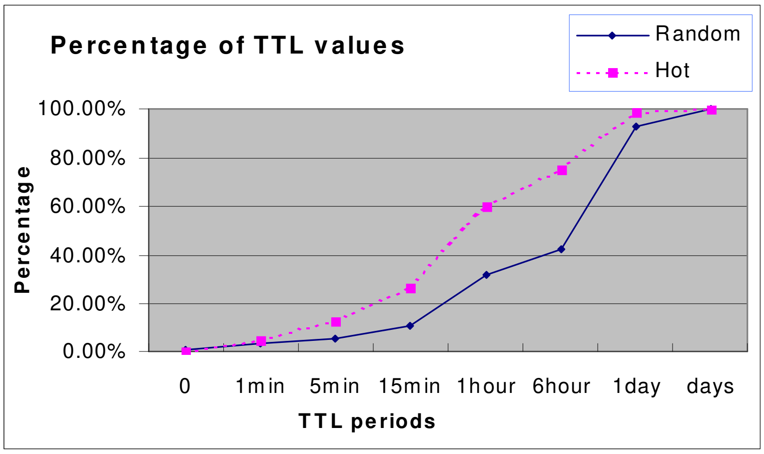

Figure 3: The Cumulative Distribution of TTL values for Random and Hot Servers

of these server sites to obtain a list of auxiliary servers being used by these sites to serve content. Table 7 shows the results if we consider all requests in a log or if we group requests within a 15-second window as "pages."

Table 7: Ratio of Object and Page Requests from Hot Servers and Their Auxiliary Servers

	Requests		Pages	
Log	Hot	Aux	Hot	Aux
ac99	0.237	0.047	0.067	0.073
bo00	0.048	0.081	0.050	0.083
uc00	0.062	0.092	0.058	0.107

The results show a relatively small percentage of requests going to these hot servers. The auxiliary servers generally received more requests, although not all of these requests may be the result of a request to a hot server. Further work in this direction with more varied user logs and traces is needed to determine the contribution of popular servers to a set of Web requests.

5.5 Range of DNS TTLs

Using this set of hot server sites we returned to our DNS study and examined the range of TTL values for the hot servers versus the set of random servers in the ac99 log. These results are shown in Figure 3. Not surprisingly, the results show that the hot servers generally have a smaller DNS TTL than do the random servers.

5.6 DNS Response Time Performance

The next piece of work was the response time study described in Section 4.2. All data collected in the previous portion of our work was independent of the site collecting the data

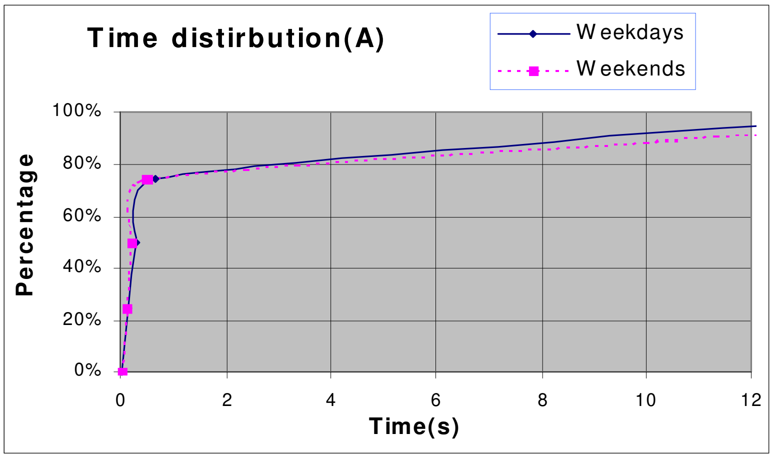

Figure 4: Cumulative Time Distribution for Authoritative Responses

as we focused on DNS cache hit rates and DNS TTL values. In this study we focused on the time to do a DNS lookup from a client at WPI. All DNS requests were made using the local WPI DNS cache so that effects from other local users influence the results. DNS requests were made on an hourly basis during a weekday and a weekend day to the set of hot servers and a set of random servers drawn from the NLANR logs.

We used the DNS resolver routines available with the Unix operating system to augment the function of *gethostbyname()* with additional information of whether the DNS mapping being returned was authoritative (A) or non-authoritative (NA), the number of IP addresses available and the DNS TTL. Through simple tests we determined that non-authoritative responses returned from the local WPI DNS cache took 2-4 msec. Using this information we separated out DNS responses into three categories:

1. Local Cache—non-authoritative responses taking less than 10ms.

2. Non-Authoritative—non-authoritative responses taking more than 10ms.

3. Authoritative—responses marked as authoritative.

Using these categories, we also did some analysis on time-of-day variations for cache hit rates and the average DNS response time over the course of a day. We did not see large variations during the day. Rather these results are summarized in Table 8 showing the median and average response times for the set of all DNS requests during a 24 hour period. The table shows that hot servers have a higher hit rate in the local cache and that hit rates are higher during the week than on the weekend. These results are not surprising given that the same set of hot servers are tested each hour and the results are influenced by other users. It is interesting to note that the non-authoritative, non-local responses often take more time that the authoritative responses.

Also important to note is that the DNS performance for hot servers is consistently better than for random servers for both non-local, non-authoritative responses and authoritative responses. These results could indicate better performance by the authoritative DNS servers for hot servers or that the random servers were generally located further away on the network.

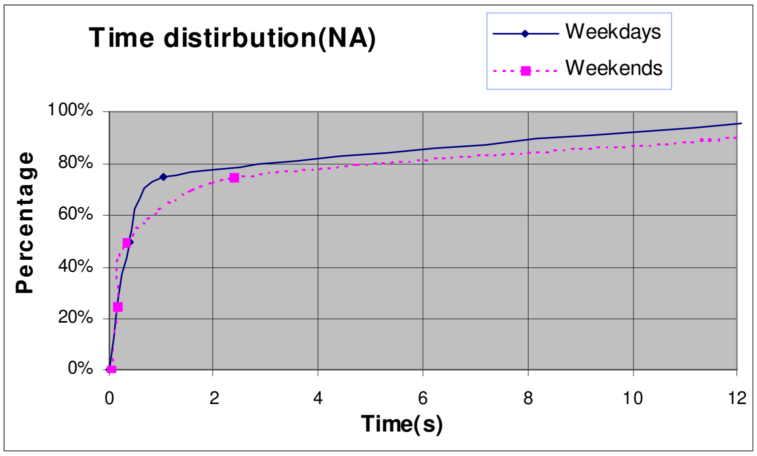

Figure 5: Cumulative Time Distribution for Non-Authoritative Responses

Table 8: Retrieval times (msec.) for Hot and Random Sites

	Local Cache			Non-Authoritative			Authoritative			All
Test Set (Day)	Pct	Med	Ave	Pct	Med	Ave	Pct	Med	Ave	Ave
Hot (w'day)	61%	3	3	8%	140	850	31%	208	546	239
R'dom(w'day)	32%	2	3	10%	407	1571	59%	299	787	615
Hot (w'end)	42%	3	3	7%	232	1721	50%	139	438	343
R'dom(w'end)	13%	3	3	7%	332	2280	81%	206	1012	972

We further analyzed this distribution by plotting the cumulative distribution for the authoritative (A) and non-authoritative responses. These are shown in Figures 4 and 5. Each figure shows that generally 20-25% of DNS lookups for servers not in the local cache take more than one second.

Table 9 attempts to eliminate the effects of repeated retrievals for the same hot servers by focusing on results for just the first retrieval. As expected, the local cache hit rate for hot servers is still higher than for the random servers due to the influence of other users, but not quite as high.

5.7 DNS Performance Relative to DNS TTL

As another direction in our study we looked to see if there was any correlation between the size of the DNS TTL and the time to do an authoritative lookup of the DNS mapping. The hypothesis is the servers with shorter TTLs would yield faster DNS response. The results for authoritative responses from random servers are shown in Figure 6. The results show that the servers with a TTL value of zero give the worst results with those servers with a DNS TTL less than or equal to one minute giving the best results with a bit of an increase in response times as the TTL grows.

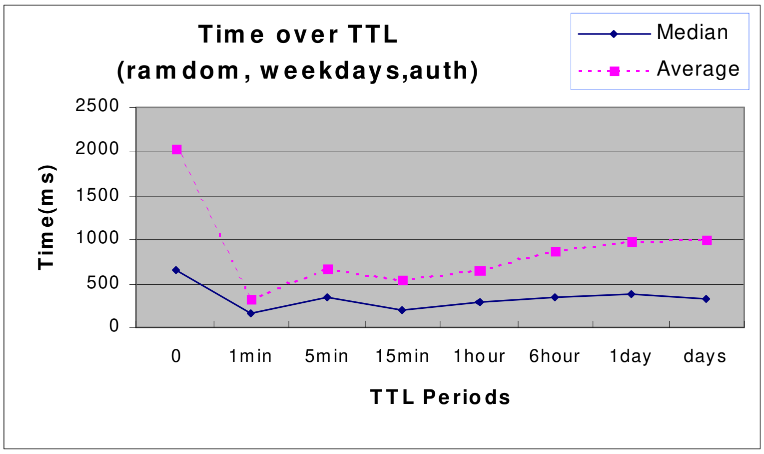

Figure 6: DNS Lookup Performance Relative to the DNS TTL

Table 9: Times (msec.) for First Retrievals from Hot and Random Sites

	Local Cache			Non-Authoritative			Authoritative			All
Test Set (Day)	Pct	Med	Ave	Pct	Med	Ave	Pct	Med	Ave	Ave
Hot (w'day)	43%	3	3	6%	493	648	51%	395	621	357
R'dom(w'day)	34%	3	3	9%	493	544	57%	499	1132	695
Hot (w'end)	32%	3	3	10%	61	1085	57%	1210	303	285
R'dom(w'end)	10%	2	3	3%	3512	2399	87%	143	1090	1021

5.8 DNS as a Contribution to End-to-End Costs

The last portion of our study used the methodology from a previous study [5] to measure the response time of the DNS lookup for a server relative to the time to retrieve the set of Web objects from that server. The set of servers was a list of approximately 700 popular servers from which valid responses for the home page of 670 servers were obtained. In serving these home pages, some images were located on other servers so that objects were retrieved from a total of 859 servers. The time to lookup up the IP address for each server was measured as well as the time to retrieve all objects from the server. These objects were retrieved using one network connection for each object with up to four objects being retrieved in parallel. This retrieval approach was labeled as “burst-1.0” in [5].

The first result derived from this test is the cumulative distribution graph of Figure 7 showing the DNS lookup times. It shows a a similar shape as Figures 4 and 5, but has sharper delineations at three and five seconds. These are known to be timeouts for different DNS clients.

The second result is to compare the DNS lookup time with the time to retrieve the set of objects for the server. A scatter plot for each of the 859 servers is shown in Figure 8. The results show a large gap between one and three seconds for the DNS lookup time. The DNS lookup cost was relatively significant contributor to the object retrieval time for servers

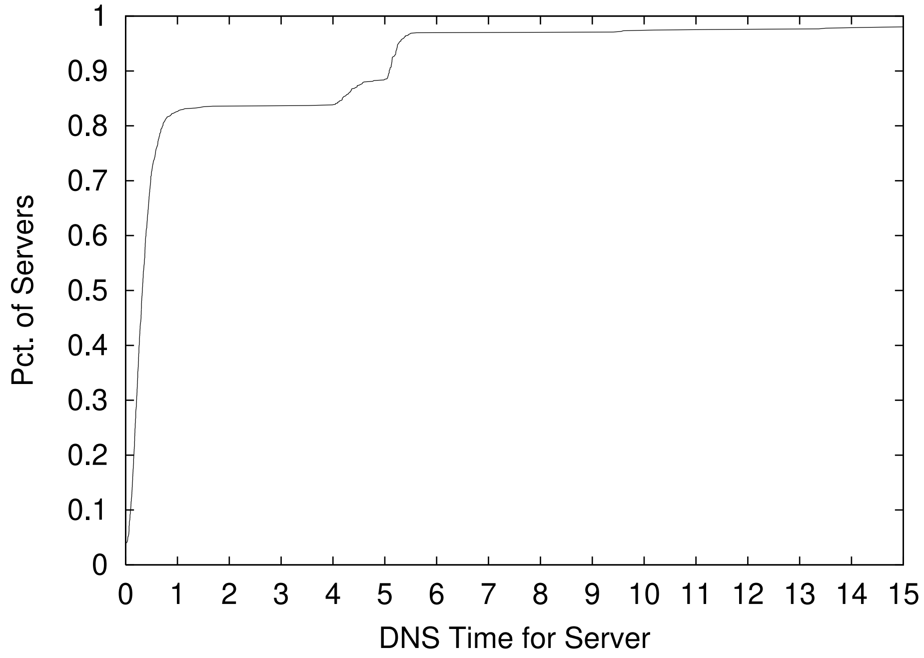

Figure 7: Cumulative Distribution of DNS Lookup Times for Popular Server Set

above this gap. This constitutes close to 20% of the servers.

6 Summary and Future Work

In this work we have concentrated on the contribution of DNS lookup to the overall Web object retrieval time. In replaying real logs of user requests we found that cached DNS entries can be used for 87-94% if each object retrieval generates a DNS request. In the more likely scenario that object requests to the same server in a short period of time generate only one DNS request then 71-78% of DNS requests were local cache hits. These values improve by about 5% if a minimum TTL of 15 minutes is imposed. These results indicate that most DNS lookups are handled by the local cache.

We also found that 7-18% of DNS lookups were cache misses for DNS entries that had been previously seen, but had become stale. Further analysis showed that only 10-20% of these DNS lookups returned a different value than the previous DNS lookup. These results indicate many authoritative DNS servers set the TTL to a lower value than the DNS lookup characteristics would warrant. The results also show that DNS lookups returning multiple IP addresses change much more frequently than lookups returning a single IP address.

In a response time study done from a client at WPI we found that authoritative DNS servers for popular, or hot, Web servers performed better than those DNS servers for random Web servers. Performance was better both in terms of local cache hit rates, which would be expected, but also for authoritative and non-local, non-authoritative responses. We also found that authoritative DNS lookups for servers returning a DNS TTL of zero showed the worst performance with lookups returning a DNS TTL up to one minute showing the best

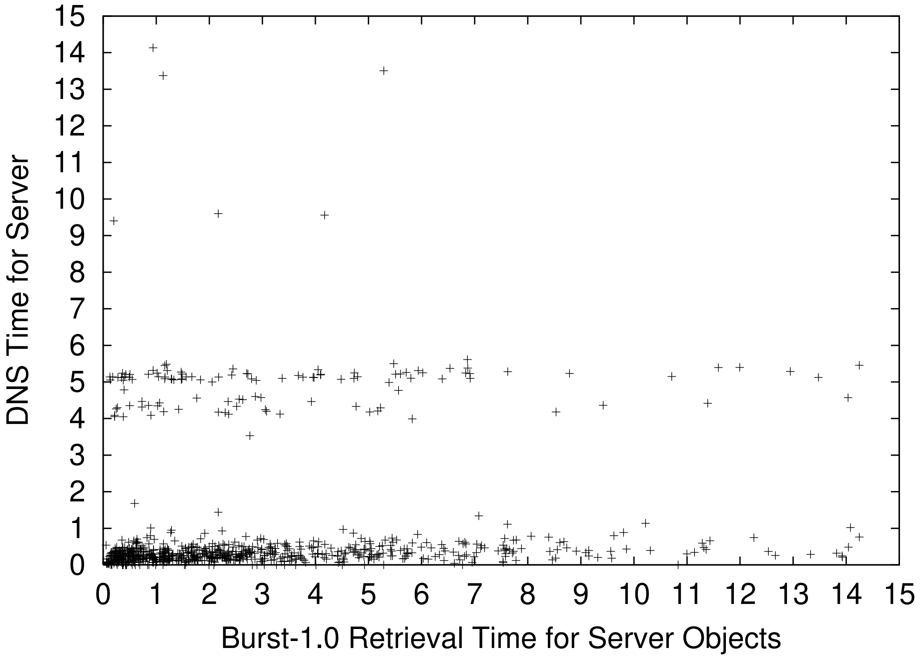

Figure 8: Relative Performance of DNS Lookup Time Compared with Object Download Time

response time. Authoritative responses for other DNS TTLs showed slightly worse DNS response times.

Finally we found that the DNS lookup time contributed more than one second to approximately 20% of retrievals for the Web objects on the home page of a larger list of popular servers. While these numbers are significant, they do not reflect the expected access patterns for URLs. If we go back to the log replay results that show only about 20% of DNS requests are not cached locally and if 20-30% of these non-cached lookups take more than one second then only 4-6% of all DNS lookups should take more than one second. These numbers appear to be less of a problem for overall Web retrieval, particularly if more larger DNS TTL values are used by servers returning infrequently changing DNS entries.

Future work should be done to explore DNS servers returning small DNS TTL values and to better determine how frequently they do change. This study should be done from more than one client. Another interesting result to follow up on is the percentage of popular servers that are referenced by real user requests. In only one of our logs was this percentage larger than 10%. This number is smaller than we would have expected.

References

[1] Edith Cohen and Haim Kaplan. Prefetching the means for document transfer: A new approach for reducing web latency. In *Proceedings of the IEEE Infocom 2000 Conference*, Tel-Aviv, Israel, March 2000. IEEE.

[2] Edith Cohen and Haim Kaplan. Proactive caching of DNS records: Addressing a performance bottleneck. In *Proceedings of the Symposium on Applications and the Internet (SAINT)*, January 2001. Accepted for publication.

[3] Christian Huitema. Webtest - testing the internet quality of service. `http://www.netsizer.com/webtest.html`.

[4] Christian Huitema. Internet quality of service assessment, February 2000. `ftp://ftp.telcordia.com/pub/huitema/stats/quality_today.html`.

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

[6] P. Mockapetris. Domain names – concepts and facilities, November 1987. `http://www.dns.net/dnsrd/rfc/rfc1034/rfc1034.html`.

[7] P. Mockapetris. Domain names – implementation and specification, November 1987. `http://www.dns.net/dnsrd/rfc/rfc1035/rfc1035.html`.

[8] David Mosberger and Tai Jin. httperf – a tool for measuring web server performance. In *Workshop on Internet Server Performance*, Madison, Wisconsin USA, June 1998.

[9] The Mozilla organization. `http://www.mozilla.org/`.

[10] NLANR. Proxy cache log traces, 2000. `ftp://ircache.nlanr.net/Traces/`.

[11] 100hot.com.