Rate of Change and other Metrics:
a Live Study of the World Wide Web

Fred Douglis, Anja Feldmann, Balachander Krishnamurthy
AT&T Labs - Research
Jeffrey Mogul
Digital Equipment Corporation - Western Research Laboratory

Rate of Change and other Metrics: a Live Study of the World Wide Web[*]

Fred Douglis[†]
Anja Feldmann[‡]
Balachander Krishnamurthy[§]
AT&T Labs – Research

Jeffrey Mogul[¶]
Digital Equipment Corporation – Western Research Laboratory

To appear, *USENIX Symposium on Internetworking Technologies and Systems*
December 1997

Abstract

Caching in the World Wide Web is based on two critical assumptions: that a significant fraction of requests reaccess resources that have already been retrieved; and that those resources do not change between accesses.

We tested the validity of these assumptions, and their dependence on characteristics of Web resources, including access rate, age at time of reference, content type, resource size, and Internet top-level domain. We also measured the rate at which resources change, and the prevalence of duplicate copies in the Web.

We quantified the potential benefit of a shared proxy-caching server in a large environment by using traces that were collected at the Internet connection points for two large corporations, representing significant numbers of references. Only 22% of the resources referenced in the traces we analyzed were accessed more than once, but about half of the references were to those multiply-referenced resources. Of this half, 13% were to a resource that had been modified since the previous traced reference to it.

We found that the content type and rate of access have a strong influence on these metrics, the domain has a moderate influence, and size has little effect. In addition, we studied other aspects of the rate of change, including semantic differences such as the insertion or deletion of anchors, phone numbers, and email addresses.

1 Introduction

The design and evaluation of Web server caches, and especially of caching proxy servers, depends on the dynamics both of client reference patterns and of the rate of change of Web resources. Some resources are explicitly indicated as uncacheable, often because they are dynamically generated. Other resources, though apparently cacheable, may change frequently. When a resource does change, the extent of the change can affect the performance of systems that use *delta-encodings* to propagate only the changes, rather than full copies of the updated resources [2, 12, 16]. The nature of the change is also relevant to systems that notify users when changes to a page have been detected (e.g., AIDE [7] or URL-minder [17]): one would like to have a metric of how “interesting” a change is. One example of an interesting change is the insertion of a new anchor (hyperlink) to another page.

A number of recent studies have attempted to character-

[*]

[†] Email: douglis@research.att.com.

[‡] Email: anja@research.att.com.

[§] Email: bala@research.att.com.

[¶] Email: mogul@pa.dec.com.

ize the World Wide Web in terms of content (e.g., [4, 22]), performance (e.g., [20]), or caching behavior (e.g., [11]). These studies generally use one of two approaches to collect data, either "crawling" (traversing a static Web topology), or analyzing proxy or server logs. Data collected using a crawler does not reflect the dynamic rates of accesses to Web resources. Data collected by analyzing logs can provide dynamic access information, such as access times and modification dates (although most existing servers and proxies provide meager log information, at best, and dynamically generated data will not typically include modification information).

To quantify the rate, nature, and extent of changes to Web resources, we collected traces at the Internet connections for two large corporate networks, including the full contents of request and response messages. One of these traces, obtained over 17 days at the gateway between AT&T Labs–Research and the external Internet, consists of 19 Gbytes of data. The other trace, obtained over 2 days at the primary Internet proxy for Digital Equipment Corporation, was collected by modifying the proxy software to record HTTP messages for selected URLs; it amounts to 9 Gbytes of data. The traces used in our study have been described elsewhere [16] and are discussed in greater detail in Section 2.

Our trace collection and analysis were motivated by several questions. A primary issue was the potential benefit of delta-encoding and/or compression to reduce bandwidth requirements, a study of which was presented separately [16]. Here we address other aspects of the rate of change. When possible, we consider how the metric is affected by variables such as frequency of access, content type, resource size, site, or top-level domain (TLD)[1]. We answer questions such as:

- How frequently are resources reaccessed? The frequency of reaccess is essential to the utility of caching and delta-encoding.

- What fraction of requests access a resource that has changed since the previous request to the same resource? If the fraction is high, simple caching may prove much less useful than a scheme that can take advantage of delta-encodings.

- How "old" are resources when accessed, i.e., what is the difference between the reference time and the last-modified time? The age of resources can be an important consideration in determining when to expire a possibly stale copy [11].

- For those references yielding explicit modification timestamps, how much time elapses between modifications to the same resource, and how do the modification rate and access rate of a resource interact? If a cache can detect modifications at regular intervals, it can use that information to improve data consistency.

- How much duplication is there in the Web? When one requests resource *X*, how often does one get something identical to resource *Y*, either on the same host or another one? Examples of such duplication include explicit mirrors and cases where a particular resource, such as an image, has been copied and made available under many URLs. The rate of duplication may be important to the success of protocols such as the proposed "HTTP Distribution and Replication Protocol" (DRP) [19], which would use content signatures, such as an MD5 checksum, to determine whether the content of a resource instance is cached under a different URL.

- Can we detect and exploit changes in semantically distinguishable elements of HTML documents, including syntactically marked elements such as anchors and other interresource references (i.e., HREF tags), and untagged elements such as telephone numbers and email addresses?

Our analyses show that over a period of over two weeks, many resources were never modified, others were modified often, and a significant fraction were modified at least once between each traced access. The rate of change depends on many factors, particularly content type but also TLD. On the other hand, the size of a resource does not appear to affect modification rates. A significant fraction of resources overlap within or between sites (i.e., different URLs refer to identical bodies). Our analysis of semantically distinguishable elements showed that some elements, such as telephone numbers, are relatively stable across versions; others, such as image references, are more likely to change from one version to another and in some cases are replaced completely on a regular basis.

[1] We use Bray's classification of TLDs [4], such as educational, commercial, government, regional, and so on.

The rest of this paper is organized as follows. Section 2 describes our traces. Section 3 elaborates on the metrics we have considered and reports our results. Section 4 discusses related work, and Section 5 concludes.

2 Traces

Both the Digital and AT&T traces stored full-content responses for a collection of outgoing HTTP requests from a corporate community. They did not log non-HTTP responses, and the AT&T trace also omitted HTTP transfers from a port other than the default HTTP port 80 (these constituted less than 1% of the data).

The AT&T trace was collected by monitoring packets through the AT&T Labs–Research firewall. All resources were logged, enabling us to consider the effects of content-type on the AT&T reference stream. The trace consists of 950,000 records from 465 distinct clients accessing 20,400 distinct servers and referencing 474,000 distinct URLs.

The Digital trace was gathered by a proxy server that passed requests through a corporate firewall. The proxy did not cache resources, though clients could. Due to storage constraints, the trace software filtered out resources with a set of extensions that suggested binary content, such as images and compressed data. Most, but not all, logged resources were textual data such as HTML.

Due to space constraints, we present results only for the AT&T trace. The restrictions on content-type in the Digital trace made it less useful for some of our analyses, but where we could obtain comparable results from both traces, we found them quite similar. The results from the Digital trace are available in an extended version of this paper [8].

3 Results

The following subsections correspond to the metrics discussed in the first section. Figure 1 provides a graphical representation of several of the attributes and their relationships to each other. Our metrics are derived from these attributes. For a particular resource, we consider a stream of accesses to it and the information available for each access. For status-200 responses (which return the body of the resource) and status-304 responses (which indicate that the resource has not changed since a previous time, provided in the request), we examine several attributes:

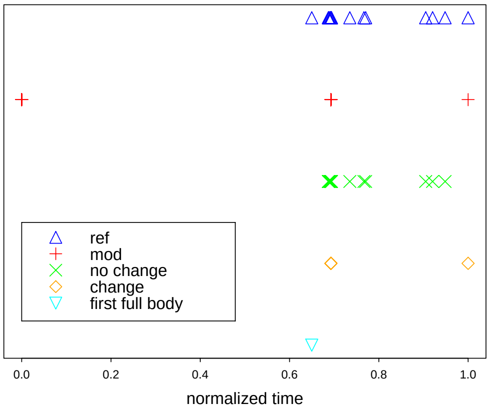


Figure 1: Visualization of the access stream for one resource. The x-axis represents a fraction of the period from the earliest last-modified timestamp for the resource until the latest reference to it. Each metric is spread across the y-axis (refer to the legend, and to the text for a detailed explanation.)

Request times The number of requests, and the time between each requests, is shown by the Δ marks in Figure 1.

Modification times The vast majority of status-200 responses (79%) contained last-modified timestamps (+ marks in Figure 1). When no last-modified information was available but the content changed, we assumed the resource was dynamically generated at the time of the request, and used the Date response header (or the timestamp of the local host if no Date header was provided).

Ages For those resources with a last-modified timestamp, the age[2] of a resource is the difference be-

[2] Note that this use of *age* differs from the HTTP/1.1 terminology, where a response Age header indicates how long a response has been

tween the request time and the last-modified time. Otherwise, it is 0. In Figure 1, the age of each reference (Δ marks in Figure 1) is the difference between the timestamp of the reference and the modification timestamp immediately below or to the left of the reference.

Modification intervals To determine the interval between modifications, we must first detect modifications. The last-modified timestamp is not always present, and when it is present, it sometimes changes even when the response body does not. Therefore, we detect changes by comparing the bodies of two successive responses for a resource. The first time a full-body response is received, we cannot tell whether it has changed (∇ marks in Figure 1). Subsequent references are indicated as "no change" (**x**) or "change" (⋄). For those modified responses with a last-modified timestamp, the time between two differing timestamps indicates a lower bound on the rate of change: the resource might have changed more than once between the observed accesses. If a modified response lacks a last-modified timestamp, then we assume that it changed at the time the response was generated. Again, the resource might have changed more than once between the observed accesses.

Statistics

In the following subsections, we present information about references and ages that span large time intervals—as much as 10^8 seconds (3.1 years) and higher. To focus on the trends across a wide time range, the graphs show the probability distributions with a logarithmic scale on the x-axis; the y-axis remains on a linear scale to emphasize the most common values. While a cumulative distribution function shows the probability that a value is less than x, it cannot clearly emphasize the most common values. Such values become more apparent when using a probability density of the *logarithm* of the data. Coupled with a logarithmic scale on the x-axis, plotting the density of the logarithm of the data facilitates direct comparisons between different parts of the graphs based on the area under the curve and is appropriate when using a log x-axis.

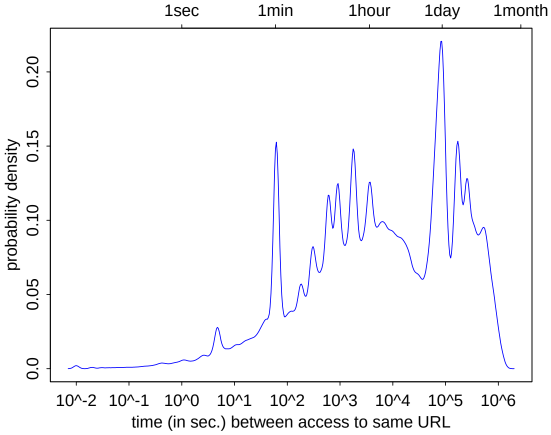


Figure 2: Density of time between accesses to the same resource, for all records in the AT&T trace. Time is shown on a log scale. Standard time units are shown across the top of the graph.

As we show later, content type bears on several of these statistics. Table 1 shows the distribution of the content types in the AT&T trace, as a fraction of unique resources and of all responses. In some cases a resource appeared with different content types over time, in which case the content type of the resource in our studies was determined by choosing the type that it appeared as most frequently. In terms of requests, images contributed to 69% of all accesses, and 64% of all resources. HTML accounted for just a fifth of accesses and about a quarter of resources. Application/octet-stream resources, which are arbitrary untyped data used by applications such as Pointcast, accounted for most of the rest of the accesses and resources. In terms of bytes transferred, GIFs contributed a relatively low amount of traffic for the number of accesses or resources, while all other content types contributed a greater share. (See [16] for additional statistics about content types.)

3.1 Access Rate

Figure 2 plots the density of the time between accesses to each resource for the AT&T trace. There are a number of peaks, with the most prominent ones corresponding to intervals of one minute and one day. The mean interarrival time was 25.4 hours with a median of 1.9 hours

cached [9].

Content	Accesses		Resources	
type	% by count	% by bytes	% by count	% by bytes
image/gif	57	36	48	18
text/html	20	21	24	33
image/jpeg	12	24	16	28
app'n/octet-stream	8	13	9	13
all others	2	6	3	8

Table 1: Content type distribution of the AT&T trace. Percentages do not sum to 100% due to rounding.

and a standard deviation of 49.6 hours. The huge difference between the median and the mean indicates that the mean is extremely sensitive to outliers. The mean of the data after applying a log-transform[3] gives a much better indication of where the weight of the probability distribution is. For this graph, the "transformed" mean is 1.6 hours.

Of the 474,000 distinct *resources* accessed in the AT&T trace, 105,000 (22%) were retrieved in a way that demonstrated repeated access: either multiple full-body status-200 responses, or at least one status-304 response that indicated that a client had cached the resource previously. A much higher portion of *references* (49%) demonstrated repeated access.

3.2 Change Ratio

We define the *change ratio* for a resource as the ratio of new instances to total references, as seen in the trace (i.e., the change ratio is the fraction of references to a resource that yield a changed instance). Overall we see that many resources are modified infrequently, but many more are modified often, and 16.5% of the resources that were accessed at least twice were modified every time they were accessed. Relative to all responses that were accesses more than once 13% had been changed since the previous traced reference to it. Yet considering all responses for which we could determine whether the resource was modified (either a status-304 response or a status-200 response that followed a previous status-200 response), 15.4% of responses reflected a modified resource.

Figure 3(a) graphs the cumulative fraction of resources that are at or below a given change ratio, organized by content type. Images almost never changed, while application/octet-stream resources almost always changed. For text/html, slightly over half the resources never changed, and most of the rest changed on each access after the first. However, this apparent high rate of change results largely from resources that were accessed just two or three times. Figure 3(b) shows just HTML resources, clustered by access count, and indicates that there is much more variation among the resources that were accessed six or more times, and that only about a fifth of those resources were modified on every access.

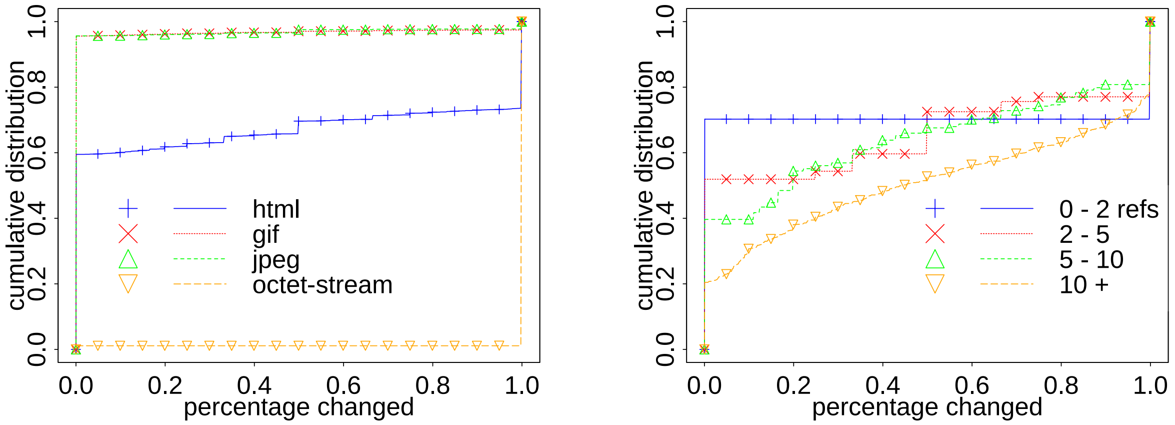
3.3 Age

Figure 4 presents density plots of the age of each resource when it is received, for those resources providing a last-modified timestamp. It omits resources for which the modification date is the access time, such as dynamically-generated data and particularly a large number of application/octet-stream resources. The results are clustered in several ways:

a. Resources are clustered by number of references. The most frequently referenced resources have the highest clustering of age, around the period of 10 days to 10 months. The curves are generally similar, indicating that the frequency of access does not have a large effect on age, although the most frequently accessed resources have a higher peak followed by a shorter tail (indicating somewhat younger responses overall).

b. Resources are clustered by content type, with application/octet-stream having a shape similar to the most frequent URLs in (a), since they contribute most of the references to the most frequently accessed resources. HTML responses are newer than the other types, with a mean of 1.8 months and a median of 12.8 days. This compares to a mean

[3] The log-transform of a set of data is exp(mean(log(data))).



(a) Grouped by content type.

(b) HTML only, by number of references.

Figure 3: Cumulative distribution of change ratio for the AT&T trace.

of 3.8 months and median of 63.9 days for gif resources, for example.

c. Resources are clustered by size. Here, there is not a large distinction between sizes, although the smallest resources tend to be somewhat older than others. This is unsurprising since there are many small images that are essentially static.

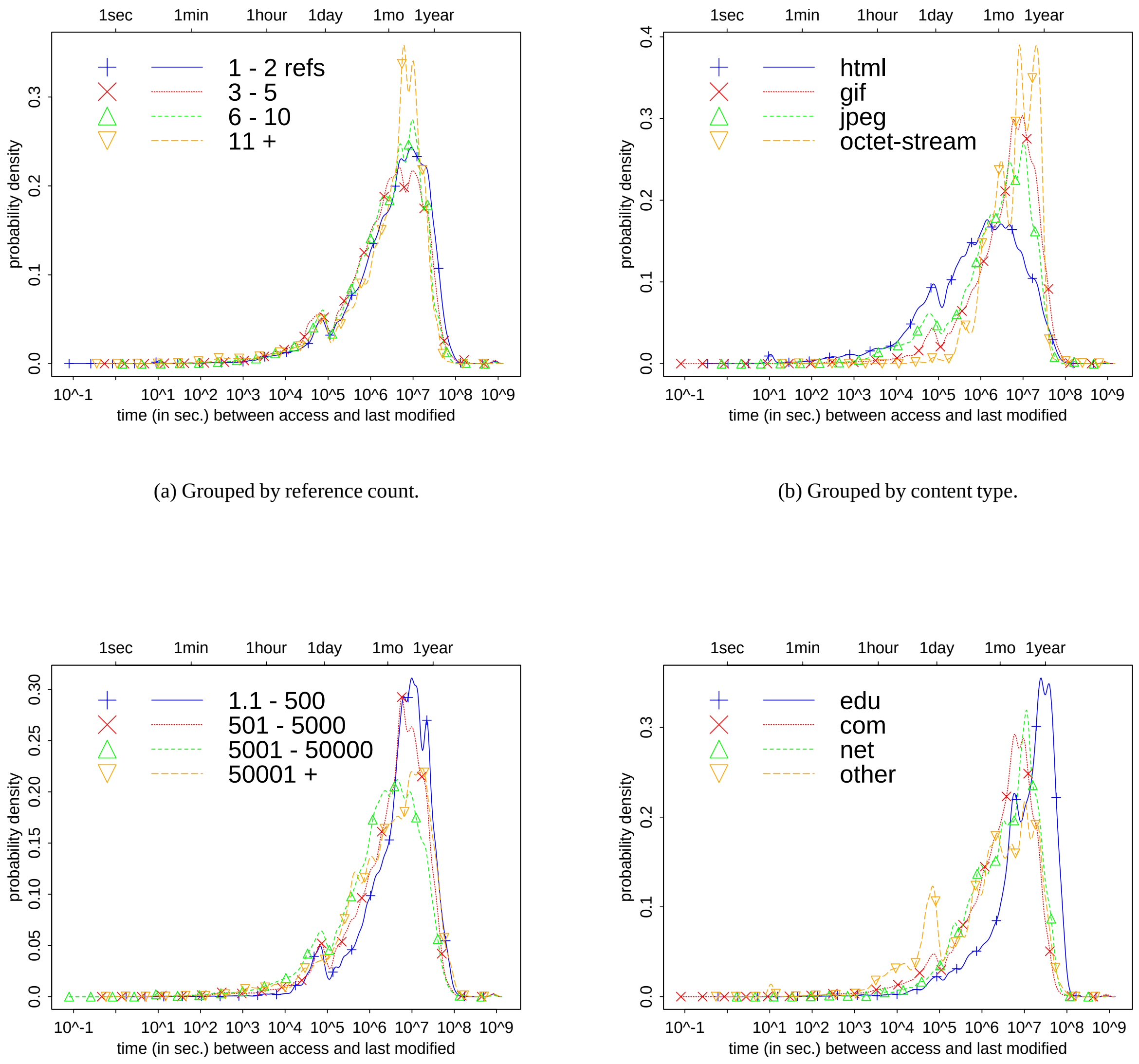
d. Resources are clustered by TLD, using Bray's categorization [4]. This clustering reduced the 20,400 host addresses to 13,300 distinct sites (such as a campus, or the high-order 16 bits of an IP address for numeric addresses). Educational sites serve resources that are noticeably older than other domains. Note that 17% of responses had no Host header; currently, these fall into the "other" category, and show some periodicity at an interval of 1 day.

Previously, Bestavros [3] and Gwertzmann and Seltzer [11] found that more popular resources changed less often than others. As described next in Section 3.4, we found that when a resource changed, its frequency of change was greater the more often it was accessed. This result suggested that in our trace, more popular resources might change more frequently rather than less. Figure 5 plots the mean age of resources, categorizing them into less frequently accessed resources (1–20 references) and more frequently accessed ones. Considering all content types (Figure 5(a)), more frequently accessed resources are clearly younger than less frequently accessed ones. Focussing on HTML (Figure 5(b)), the difference is even more pronounced.

The differences between our results and the earlier studies are striking, but they may be explained by considering the environments studied. The earlier studies reported on servers at Boston University and Harvard University (the educational TLD), while we looked at everything accessed by a community of users. A number of resources, such as "tickers" that update constantly changing information, were accessed frequently and changed on (nearly) each access.

3.4 Modification Rate

Figure 6 presents density plots of the time between last-modified timestamps for the AT&T trace, when a resource has changed. Figure 6(a) clusters the modification intervals by the number of accesses to each resource, demonstrating that, of the resources that change, the most frequently accessed resources have the shortest intervals between modification. (Naturally, since we are



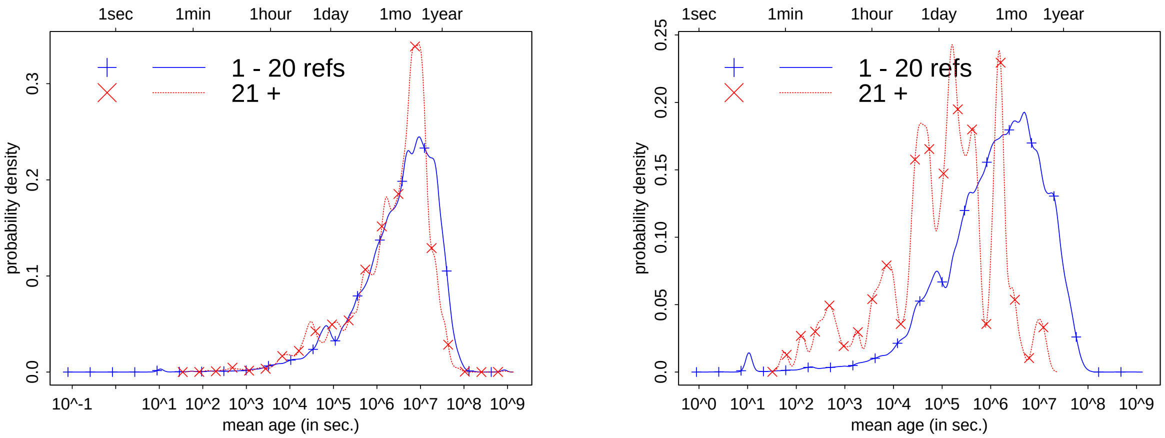
(a) Grouped by reference count.

(b) Grouped by content type.

(c) Grouped by size (in bytes).

(d) Grouped by top-level domain (TLD).

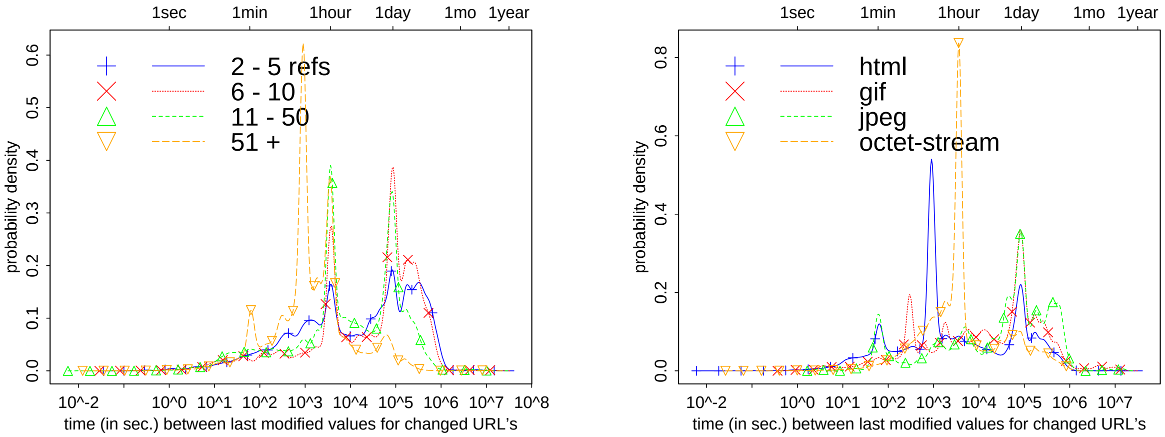
Figure 4: Density plot of age of resources, clustered by various metrics, for the AT&T trace. Times are shown on a log scale.



(a) All content types.

(b) HTML only.

Figure 5: Density plot of age of resources, focussing on frequently accessed resources, for the AT&T trace. Times are shown on a log scale.



(a) Grouped by reference count.

(b) Grouped by content type.

Figure 6: Density plot of the time between last-modified timestamps, clustered by reference count and content type, for the AT&T trace. Times are shown on a log scale.

limited to observing modification times when resources are accessed, we cannot tell whether there are resources that are changing frequently but being accessed no more than once during the trace interval.)

The peaks in Figure 6(a) appear at somewhat intuitive intervals: 1 minute, 15 minutes, 1 hour, and 1 day. As the number of references increases, one is more likely to observe updates with fixed periodicity, such as the peak at 15 minutes for resources that are accessed 51 or more times. Also, the probability density falls off after 10 days, which is largely due to the 17-day duration of our trace and the need to observe two different last-modified dates.

Figure 6(b) clusters the modification intervals by content type, and indicates that HTML resources that are modified at all will change more often than more static content types such as images. Some of these are the "tickers" mentioned above that are updated at 15-minute intervals. We also observe the effect of Pointcast (serving resources of type application/octet-stream), which has a one-hour update interval. Some images and HTML pages change daily, which may be partly due to the effect of advertisements and regular updates.

3.5 Duplication

Our content-based analysis required that we compute checksums of the content of each instance of each resource, in order to determine when changes occurred. In the process, we found some interesting phenomena: in particular, that 146,000 (18%) of the full-body responses in the AT&T trace that resulted in a new instance of a particular resource were identical to at least one other instance of a *different* resource. There are several common causes for this:

1. Multiple URLs may refer to a single server and return the same content. Most commonly this overlap is due to some form of unique session-identifier embedded within the URL. In one case alone, there were 443 distinct URLs that referred to the same content on the same host.

2. The same body may be replicated on multiple hosts, usually as an explicit "mirror," or an image that has been copied to reside with the HTML resources that embed it. The "Netscape Now" icon, the "blue ribbon" campaign, and various site-rating logos are examples of this.

3. Different resources may be used to convey information, for instance to inform a server of the origin of the link.

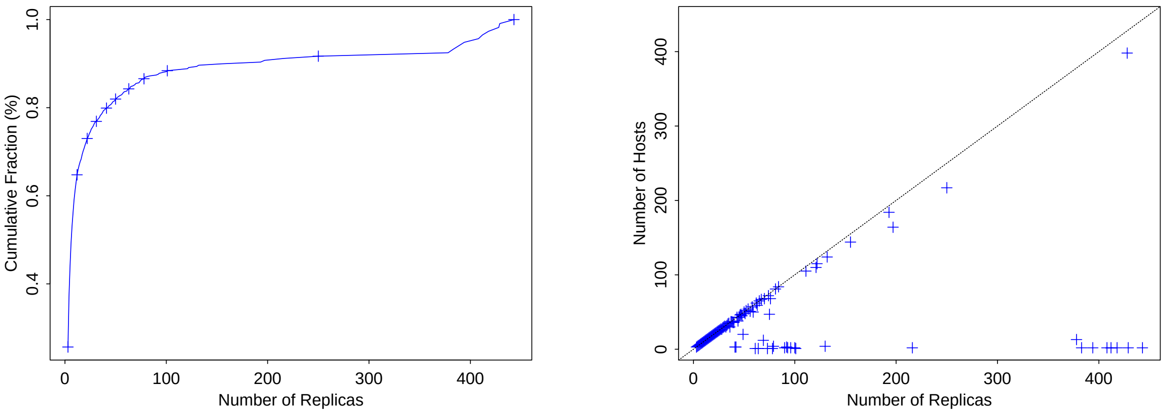
Figure 7(a) plots a cumulative distribution of the frequency of replication. Most bodies that are replicated appear just twice, but six appear over 400 times. Figure 7(b) plots the number of distinct hosts appearing in the set of resources for each replicated body, and shows that some appear just once (all replicas are served by the same host) while others follow the dashed line that indicates an equal number: every replica is served by a different host.

At first glance, the extent of replication suggests that a mechanism to identify replicas might serve to improve the effectiveness of caching. However, most of the resources are accessed multiple times and a traditional cache would eliminate many of the references to them. Of the rest, many are uncacheable and would need to be retrieved on each access regardless. Thus, the benefit of identifying duplicates would be to reduce the storage demands of the cache (not generally a large problem in today's environments) and to eliminate one access from each but the first host that serves the resource.

3.6 Semantic differences between instances

We define a semantically interesting item to be a recognizable pattern that occurs reasonably often in the context of Web pages. For example, telephone numbers (in various guises) are one class of pattern. Across instances of a Web page, changes in telephone numbers may be of interest. The manner in which we recognize semantically interesting patterns, referred to as *grinking* (for "grok and link"), is part of another project [14]; we concentrate here on the rate of change of semantically interesting items.

Using the AT&T trace, we looked for the following classes of patterns: HREFs (hyperlinks), IMGs (image references), email addresses, telephone numbers, and URL strings that occur in the body of a Web page. Because each of these forms can occur in many different ways, we probably did not recognize every occurrence. For example, a preliminary study [14] found over twenty



(a) Cumulative distribution of frequency of replication.

(b) Number of hosts by comparison to number of replicas.

Figure 7: Duplication of instances in the AT&T trace.

different variations of North American telephone number syntax.

More importantly, we cannot always assert a string matching one of these patterns is indeed a telephone number. For example, it is possible that the string "202-456-1111" is not actually a telephone number, although it is likely to be one, especially if the phrase "White House" appears in the same page. While we currently use a context-independent technique to recognize patterns, one could enhance the reliability of recognition by using the context surrounding the pattern. We would be more confident of a pattern suspected to be a telephone number if the string "phone", "telephone", or "tel" occurs in the surrounding context.

Grinking only makes sense for text files, which greatly reduced the number of responses we had to analyze. Also, we decided to look at only the first ten instances of a resource, since later changes are likely to follow the same behavior. We looked at 29846 instances of 8655 different resources. 55% of these resources were referenced twice; 71% were referenced three times or fewer; 90% were referenced 9 times or fewer. Table 2 presents the number of instances that had no recognizable forms of a particular type, such as HREFs.

For each instance of each resource we computed the semantic change by looking at the addition and deletions of forms. We define the *churn* for a given form as the fraction of occurrences of that form that change between successive instances of a resource. For example, if an instance of a resource has eight telephone numbers, and the next instance of that resource changes four of those telephone numbers, then the churn is 50%. We computed a churn value for each instance of a resource that contained the given form (except for the first one seen in the trace), then averaged the result over all instances, including the first. The results are in Table 3, which shows, for each class of form, the fraction of original occurrences of that form (as a percentage) that experienced a given amount of churn. For example, 1.5% of the recognized email addresses changed between instances in at least 75% of the cases.

As shown in Table 3, 5% of IMG references changed totally between instances, while fully qualified (10-digit) phone numbers changed the least. In 98% of the cases, when 10-digit telephone numbers were present, they did not change at all between instances.

These results are not too surprising. The stability of forms like telephone numbers may be useful in other contexts. In the future, we would like to compare the semantic difference between instances against a bitwise delta-encoding. Such a measure would tell us if the instances differ *only* in recognizably meaningful ways.

Form	Instances	Percent
HREF	7720	25.9
IMG	8331	27.9
Email	23795	79.7
10-digit phone	27531	92.2
7-digit phone	23788	79.7

Table 2: Number of instances which had no forms recognized

Churn	HREF	IMG	Email	10-digit Phone	7-digit Phone
100%	3.3	4.7	1.4	0.9	3.2
$\geq$ 75%	5.6	6.2	1.5	1.0	4.9
$\geq$ 50%	9.7	12.6	2.1	1.4	6.3
$\geq$ 25%	17.8	24.6	2.6	1.6	7.1
0%	41.2	48.6	96.5	98.0	90.2

Table 3: Percentage of instances having a given value of “churn” in semantically recognized forms.

3.7 Additional Statistics

We analyzed the packet traces to compute statistics on a number of issues beyond the issue of the rate of change. In particular, we were interested in the presence of information in the HTTP request or response headers that would affect the cachability of resources: modification timestamps, authentication, cookies, pragmas, or expiration timestamps. Of the 820,000 status-200 responses, 650,000 (79.4%) contained last-modified times, without which browsers and proxy-caching servers will generally not cache a resource. Surprisingly, 136,000 responses (16.5%) involved cookies, while 48,500 (5.9%) had some form of explicit disabling of the cache, nearly all of which are from a Pragma: no-cache directive.

4 Related work

One can gather data from a number of sources, both static (based on crawling) and dynamic (based on user accesses). Viles and French [20] studied the availability and latency of a set of servers that were found through web-crawling, primarily to ascertain when machines were accessible and how long it took to contact them. Woodruff, et al [22] used the Web crawler for the Inktomi [13] search engine to categorize resources based on such attributes as size, tags, and file extensions. For HTML documents, they found a mean document size of 4.4 Kbytes. Bray [4] similarly used the Open Text Index [18] to analyze a large set of resources. He found an mean resource size of 6.5 Kbytes and a median of 2 Kbytes. Bray's study primarily focussed on the relationships *between* resources, e.g. the number of inbound and outbound links.

Our traces represent dynamic accesses, so the sizes of resources that are actually retrieved by a set of hosts is expected to be different from the set of all resources found by a web-crawler. In our AT&T trace, the mean was nearly 8 Kbytes, with a median of 3.3 Kbytes. Our Digital proxy trace showed a mean of less than 7 Kbytes, and a median of 4.0 Kbytes. The response-body size differences between our two traces is due to the omission of certain content types from the Digital trace; these content-types show a larger mean, and a larger variance, than the included types [16].

Several studies have considered dynamic accesses, though they have not considered the frequency or extent of modifications. Cunha et al. [6] instrumented NCSA Mosaic to gather client-based traces. Those traces were then used to consider document type distributions, resource popularity, and caching policies. Williams et al. [21] studied logs from several environments to evaluate policies governing the removal of documents from a cache. Like us, they used logs from proxy-caching servers as well as *tcpdump*, but they examined headers only. They noted that dynamic documents that are presently uncacheable could be used to transmit the differences between versions. This idea was developed in more detail in WebExpress [12] and "optimistic deltas" [2]. A later study by Mogul et al. [16] quantified the potential benefits of delta-encoding and compression, using the same traces as we used for this paper. Arlitt and Williamson used server logs from several sites to analyze document types and sizes, frequency of reference, inter-reference times, aborted connections, and other metrics [1]. Here we considered many of the same issues, from the perspective of a collection of clients rather than a relatively small number of servers.

Kroeger et al. [15] recently studied the potential for caching to reduce latency, using simulations based on traces of request and response headers. They found that even an infinite-size proxy cache could eliminate at most 26% of the latency in their traces, largely because of the same factors we observed: many URLs are accessed only once, and many are modified too often for caching to be effective.

Gribble and Brewer [10] studied traces from a large collection of clients at U.C. Berkeley, gathered via a packet-sniffer like the one used for our AT&T trace. They examined a largely different set of metrics, such as access rates, locality of reference, and service response times.

Broder, et al. [5] analyze the *syntactic* similarity of files, using a web-crawler to create "sketches" of all accessible resources on the Web. These sketches can be used to find resources that are substantially similar. Such an approach might be an efficient way to find near-duplicates to which our work on semantic differences (and our previous work on delta-encoding [16]) is best applied.

5 Conclusions and Future Work

We have used live traces of two large corporate communities to evaluate the rate and nature of change of Web resources. We found that many resources change frequently, and that the frequency of access, age since last modified, and frequency of modification depend on several factors, especially content type and top-level domain, but not size.

Our observations suggest limits on the utility of simple Web caches. The assumptions upon which most current Web caching is based, locality of reference and stability of value, are only valid for a subset of the resources in the Web. Designers of advanced Web caches must confront the high rate-of-change in the Web, if they are going to provide significant latency or bandwidth improvements over existing caches.

In addition to the rate-based analysis, we performed semantic comparisons to multiple versions of textual documents and found that some entities such as telephone numbers are remarkably stable across versions. Semantic comparisons may prove useful in conjunction with notification tools [7, 17] as well as search engines, directories, and other Web services.

We are collecting a significantly larger trace dataset, to verify our conclusions here. We also intend to perform an extended semantic-difference study to locate minor changes in otherwise-identical web pages. We plan to investigate whether rate-of-change metrics can identify cases where it might be useful to pre-compute and cache delta-encodings at the server, and where prefetching of resources or delta-encodings might be beneficial.

Acknowledgments

Gideon Glass provided helpful comments on an earlier draft. We also thank the anonymous referees for their comments.

References

[1] Martin F. Arlitt and Carey L. Williamson. Web server workload characterization: The search for invariants (extended version). Technical Report DISCUS Working Paper 96-3, Dept. of Computer Science, University of Saskatchewan, March 1996. Available as ftp://ftp.cs.usask.ca/pub/discus/paper.96.3.ps.Z.

[2] Gaurav Banga, Fred Douglis, and Michael Rabinovich. Optimistic deltas for WWW latency reduction. In *Proceedings of 1997 USENIX Technical Conference*, pages 289–303, Anaheim, CA, January 1997. Also available as http://www.research.att.com/˜douglis/papers/optdel.ps.gz.

[3] A. Bestavros. Speculative data dissemination and service to reduce server load, network traffic and service time in distributed information systems. In *Proceedings of the 12th International Conference on Data Engineering*, pages 180–189, New Orleans, February 1996.

[4] Tim Bray. Measuring the Web. In *Proceedings of the Fifth International World Wide Web Conference*, pages 993–1005, Paris, France, May 1996. Also available as http://www5conf.inria.fr/fich_html/papers/P9/Overview.html.

[5] Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig. Syntactic clustering of the web. In *Proceedings of the Sixth International World Wide Web Conference*, Santa Clara, CA, April 1997. Available as http://www6.nttlabs.com/HyperNews/get/PAPER205.html.

[6] Carlos R. Cunha, Azer Bestavros, and Mark E. Crovella. Characteristics of WWW client-based traces. Technical Report BU-CS-95-010, Computer Science Department, Boston University, July 1996. Also available as http://www.cs.bu.edu/techreports/95-010-www-client-traces.ps.Z.

[7] Fred Douglis, Thomas Ball, Yih-Farn Chen, and Eleftherios Koutsofios. The AT&T Internet Difference Engine: Tracking and viewing changes on the web. *World Wide Web*, January 1998. To appear. Also published as AT&T Labs–Research TR 97.23.1, April, 1997, available as http://www.research.att.com/~douglis/papers/aide.ps.

[8] Fred Douglis, Anja Feldmann, Balachander Krishnamurthy, and Jeffrey Mogul. Rate of change and other metrics: a live study of the World Wide Web. Technical Report #97.24.2, AT&T Labs–Research, Florham Park, NJ, December 1997. Available as http://www.research.att.com/library/trs/TRs/97/97.24/97.24.2.body.ps.

[9] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, et al. RFC 2068: Hypertext transfer protocol — HTTP/1.1, January 1997.

[10] Steven D. Gribble and Eric A. Brewer. System design issues for internet middleware services: Deductions from a large client trace. In *Proceedings of the Symposium on Internetworking Systems and Technologies*. USENIX, December 1997. To appear.

[11] James Gwertzman and Margo Seltzer. World-Wide Web cache consistency. In *Proceedings of 1996 USENIX Technical Conference*, pages 141–151, San Diego, CA, January 1996. Also available as http://www.eecs.harvard.edu/~vino/web/usenix.196/.

[12] Barron C. Housel and David B. Lindquist. WebExpress: A system for optimizing Web browsing in a wireless environment. In *Proceedings of the Second Annual International Conference on Mobile Computing and Networking*, pages 108–116, Rye, New York, November 1996. ACM. Also available as http://www.networking.ibm.com/artour/artwewp.htm.

[13] Inktomi. http://inktomi.berkeley.edu, January 1997.

[14] Guy Jacobson, Balachander Krishnamurthy, and Divesh Srivastava. Grink: To grok and link. Technical Memorandum, AT&T Labs–Research, July 1996.

[15] Thomas M. Kroeger, Darrell D. E. Long, and Jeffrey C. Mogul. Exploring the bounds of web latency reduction from caching and prefetching. In *Proceedings of the Symposium on Internetworking Systems and Technologies*. USENIX, December 1997. To appear. Available as http://WWW.cse.ucsc.edu/~tmk/ideal.ps.

[16] Jeffrey Mogul, Fred Douglis, Anja Feldmann, and Balachander Krishnamurthy. Potential benefits of delta-encoding and data compression for HTTP. In *Proceedings of SIGCOMM'97*, pages 181–194, Cannes, France, September 1997. ACM. An extended version appears as Digital Equipment Corporation Western Research Lab TR 97/4, July, 1997, available as http://www.research.digital.com/wrl/techreports/abstracts/97.4.html.

[17] Url-minder. http://www.netmind.com/URL-minder/URL-minder.html, December 1996.

[18] Opentext. http://www.opentext.com, 1997.

[19] Arthur van Hoff, John Giannandrea, Mark Hapner, Steve Carter, and Milo Medin. The http distribution and replication protocol. W3C Note, available as http://www.w3.org/TR/NOTE-drp-19970825.html, August 1997.

[20] Charles L. Viles and James C. French. Availability and latency of World Wide Web information servers. *Computing Systems*, 8(1):61–91, Winter 1995.

[21] S. Williams, M. Abrams, C. R. Standridge, G. Abdulla, and E. A. Fox. Removal policies in network caches for World-Wide Web documents. In *Proceedings of SIGCOMM'96*, volume 26,4, pages 293–305, New York, August 1996. ACM. Also available as http://ei.cs.vt.edu/~succeed/96WAASF1/.

[22] Allison Woodruff, Paul M. Aoki, Eric Brewer, Paul Gauthier, and Lawrence A. Rowe. An investigation of documents from the WWW. In *Proceedings of the Fifth International WWW Conference*, pages 963–979, Paris, France, May 1996. Also available as http://www5conf.inria.fr/fich_html/papers/P7/Overview.html.